



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΚΑΙ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΤΟΠΟΓΡΑΦΙΑΣ – ΕΡΓΑΣΤΗΡΙΟ ΤΗΛΕΠΙΣΚΟΠΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη και Αξιολόγηση Αλγορίθμων Τηλεπισκόπησης για τη Διαχείριση και Ανάλυση Μεγάλων Γεωχωρικών Δεδομένων



Δημήτριος Μπληζιώτης

Αθήνα, Ιούλιος 2014

Αφιερώνεται

*Στους γονείς μου
Λεωνίδα και Ελευθερία*

*Στον αδερφό μου
Γιώργο*

Στην Κλεοπάτρα

ΠΡΟΛΟΓΟΣ

Η παρούσα Διπλωματική Εργασία εκπονήθηκε στα πλαίσια ολοκλήρωσης των προπτυχιακών μου σπουδών στη Σχολή Αγρονόμων και Τοπογράφων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου. Το θέμα της διπλωματικής μου ανατέθηκε από το εργαστήριο Τηλεπισκόπησης του Τομέα Τοπογραφίας της σχολής.

Ο Αγρονόμος και Τοπογράφος Μηχανικός ασχολείται με ένα μεγάλο εύρος επιστημονικών πεδίων όπως είναι για παράδειγμα η τηλεπισκόπηση, η φωτογραμμετρία, η γεωγραφία, η γεωδαισία, το κτηματολόγιο, τα συγκοινωνιακά, η διαχείριση των υδατικών πόρων και άλλα. Τα τελευταία χρόνια με την είσοδο των σύγχρονων τεχνολογιών, τα αντικείμενα ενασχόλησης του Τοπογράφου Μηχανικού έχουν διευρυνθεί και έχουν αποκτήσει νέες διαστάσεις. Πιο συγκεκριμένα, με την ενσωμάτωση της δορυφορικής τεχνολογίας καθώς και των Γεωγραφικών Συστημάτων Πληροφοριών (GIS) έχουν αναπτυχθεί ειδικές μέθοδοι για την επεξεργασία ψηφιακών εικόνων και διανυσματικών στοιχείων παρέχοντας ιδιαίτερα χρήσιμες πληροφορίες, διευκολύνοντας και βελτιώνοντας τον εντοπισμό, τη χαρτογράφηση και την πλοήγηση σε πολλές εφαρμογές με πολύ μεγάλη χωρική ακρίβεια.

Στο πλαίσιο ενσωμάτωσης των νέων τεχνολογιών στο επιστημονικό φάσμα του Αγρονόμου και Τοπογράφου Μηχανικού ανέλαβα τη συγκεκριμένη Διπλωματική Εργασία με σκοπό την αξιοποίηση μεθόδων παρατήρησης γης και των GIS για την ανάπτυξη αλγορίθμων τηλεπισκόπησης αξιοποιώντας μεγάλα γεωχωρικά δεδομένα. Το κίνητρο μου για να αναλάβω το συγκεκριμένο θέμα ήταν το ενδιαφέρον μου αφενός για τον τομέα της τηλεπισκόπησης και αφετέρου για την επιστήμη της πληροφορικής, των ανοιχτών λογισμικών και του προγραμματισμού.

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω όσους βοήθησαν στην υλοποίηση της παρούσας εργασίας. Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Λέκτορα κύριο Καραντζαλο, για την απεριόριστη βοήθεια και συμπαράστασή του σε όλη τη διάρκεια εκπόνησης της εργασίας. Επιπλέον, θα ήθελα να ευχαριστήσω τον κύριο Ιωσηφίδη Χρήστο, Ε.ΔΙ.Π., καθώς και τον Δρ. Κολοκούση Πολυχρόνη, Ε.ΔΙ.Π., οι οποίοι με «μύησαν» στον κόσμο του ελεύθερου και ανοιχτού λογισμικού προσφέροντάς μου παράλληλα βοήθεια σε τεχνικά θέματα.

ΠΕΡΙΛΗΨΗ

Στη συγκεκριμένη διπλωματική εργασία σχεδιάστηκαν, αναπτύχθηκαν και αξιολογήθηκαν αλγόριθμοι τηλεπισκόπησης, με απώτερο σκοπό τη διαχείριση και ανάλυση μεγάλων γεωχωρικών δεδομένων. Για το σκοπό αυτό, χρησιμοποιήθηκαν δορυφορικά τηλεπισκοπικά δεδομένα από το δορυφόρο Landsat 8, καθώς και ελεύθερα διανυσματικά δεδομένα από αντίστοιχους ανοιχτούς ιστοχώρους.

Πιο συγκεκριμένα, ο σχεδιασμός επικεντρώθηκε στην αποτελεσματική αντιμετώπιση εφαρμογών ανίχνευσης μεταβολών, αξιοποιώντας διαδικασίες μηχανικής μάθησης σε μεγάλα δεδομένα παρατήρησης της γης. Προς την κατεύθυνση αυτή, αναπτύχθηκαν και αξιολογήθηκαν αλγόριθμοι σε όλα τα στάδια της ανάλυσης των τηλεπισκοπικών δεδομένων, δίνοντας ιδιαίτερη έμφαση στην προεπεξεργασία τους. Αρχικά, στο στάδιο της προεπεξεργασίας των δεδομένων αναπτύχθηκαν λογισμικά, σε γλώσσα προγραμματισμού C/C++, που πραγματοποιούν τις απαραίτητες ραδιομετρικές διορθώσεις σύμφωνα με τη βιβλιογραφία της USGS. Παράλληλα, αναπτύχθηκε λογισμικό που ανιχνεύει και αφαιρεί σύννεφα και σκιές από μια απεικόνιση λαμβάνοντας υπόψη διαχρονικά δεδομένα και χρονοσειρές του ίδιου δορυφόρου. Επίσης, στον πυρήνα της μεθοδολογίας που αναπτύχθηκε εφαρμόζονται διαδικασίες μηχανικής μάθησης καθώς και εξόρυξης δεδομένων (data mining) σε αλγορίθμους ταξινόμησης, με στόχο τη δημιουργία και εφαρμογή ενός στατιστικού μοντέλου εκπαίδευσης για την ταξινόμηση περιοχών που δεν συμμετείχαν στη διαδικασία εκπαίδευσης. Τέλος, τα αποτελέσματα που προέκυψαν από τις επεξεργασίες αποθηκεύονται σε γεωχωρική βάση και ενσωματώνονται σε δοκιμαστική έκδοση εφαρμογής διαδικτυακού συστήματος GIS για τη διαχείριση, απεικόνιση και διάθεσή τους.

Όλα τα εργαλεία που χρησιμοποιούνται στο πλαίσιο της διπλωματικής εργασίας αξιοποιούν λογισμικά και δεδομένα που διέπονται από τις αρχές του ανοιχτού λογισμικού και των ανοιχτών προτύπων αντίστοιχα. Πιο συγκεκριμένα, χρησιμοποιούνται λογισμικά όπως είναι το orfeoToolbox, το qgis, η βιβλιοθήκη OpenCV κ.α. Επιπλέον, τα προγράμματα που δημιουργήθηκαν διέπονται και αυτά από τις αρχές του ελεύθερου λογισμικού και χαρακτηρίζονται από την άδεια GNU General Public License με σκοπό την περαιτέρω έρευνα και βελτίωσή τους.

Λέξεις Κλειδιά: Μεγάλα Δεδομένα, Μηχανική μάθηση, Εξόρυξη Δεδομένων, Προεπεξεργασίες, Ταξινομήσεις, Μέθοδοι Παρατήρησης Γης, Ανοιχτά Δεδομένα, Ελεύθερα Λογισμικά, Χρήσεις/Κάλυψη Γης

ABSTRACT

In this Diploma Thesis are designed, developed and evaluated remote sensing algorithms, which are designed with the underlying aim to manage and analyze big geospatial data. For this purpose, remote sensing data from the satellite Landsat 8 and freely vector data from the respective websites were used.

As part of this work, it was attempted to devise a methodological approach that can effectively address change detection applications, utilizing machine learning processes in big data, for earth observation purposes. For this purpose, algorithms and programs were created and evaluated at all stages of the analysis process, placing great emphasis on the preprocessing of the data. At the stage of preprocessing programs, in C/C++ computer language, were developed which accomplish the necessary radiometric processing for data of Landsat 8, in accordance to the official bibliography of USGS. Moreover, a program that detects and removes clouds and shadows of a scene, taking into account multitemporal data and time series of the same satellite, was developed and tested in various areas of interest. The methodological approach that is presented, takes advantage of machine learning procedures and data mining techniques in order to implement a statistical learning model, which can classify areas that weren't involved in the training process. Finally, the results that were produced from the processings were stored in geospatial database and incorporated in demo Web Gis application for their management, display and disposal.

The aim of this thesis was also to use software and data governed by the principles of open source and open standards, respectively. More specifically, software such as orfeo toolbox, Qgis and the library OpenCV were used in all the stages of the analysis. Additionally, the programs that were created are also governed by the principles of free software and are characterized by the licence GNU General Public Licence for further research and improvement.

Keywords: Big Data, Machine Learning, Data Mining, Preprocessing, Classification, Earth Observation Methods, Open Data, Free Software, Land Cover

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	i
ΠΕΡΙΛΗΨΗ.....	ii
ABSTRACT.....	iii
1. ΕΙΣΑΓΩΓΗ.....	1
1.1 Τηλεπισκόπηση και Μέθοδοι Παρατήρησης Γης.....	2
1.2 Σύγχρονες Εφαρμογές	2
1.3 Ποιοτικά Χαρακτηριστικά Δεδομένων	3
1.4 Επεξεργασία και Ανάλυση Δεδομένων	4
1.5 Δορυφορικοί Τηλεπισκοπικοί Δέκτες	5
1.6 Λογισμικά Επεξεργασίας	5
1.6.1 Ελεύθερα λογισμικά	6
1.6.2 Εμπορικά λογισμικά.....	7
2. ΜΕΓΑΛΑ ΓΕΩΧΩΡΙΚΑ ΔΕΔΟΜΕΝΑ ΠΑΡΑΤΗΡΗΣΗΣ ΓΗΣ	11
2.1 Μεγάλα Δεδομένα – Γενικά στοιχεία	12
2.2 Εφαρμογές σε Μεγάλα Δεδομένα.....	14
2.3 Εφαρμογές Μεγάλων Δεδομένων στα GIS.....	15
2.4 Εφαρμογές Μεγάλων Δεδομένων στην Τηλεπισκόπηση.....	17
2.4.1 Distributed Oceanographic Data Systems (DODS)	18
2.4.2 MODIS Online Visualization and Analysis System (MOVAS).....	19
2.4.3 Near-line Archive Data Mining (NADM)	19
2.4.4 MODISAzure.....	20
2.4.5 Mass Processing.....	21
2.4.6 Συγχώνευση Πολυμορφικών Δεδομένων.....	23
2.5 Τεχνολογικά Προβλήματα – Τεχνικές Δυσκολίες	24
2.6 Πλατφόρμες Διαχείρισης και Επεξεργασίας	25
2.6.1 Hadoop (Apache Software Foundation)	26
2.6.2 Rasdaman, Raster Data Manager	27
2.7 Υπολογιστικά Συστήματα και Απαιτήσεις	29
2.7.1 Συστήματα	29
2.7.2 Απαιτήσεις	31
3. ΠΡΟΓΡΑΜΜΑ ΔΟΡΥΦΟΡΩΝ LANDSAT (USGS).....	33
3.1 Γενικά Στοιχεία.....	34
3.2 Δορυφορικό Πρόγραμμα Landsat 8	35
3.2.1 Βασικά στοιχεία	35
3.2.2 Φάσμα που αναλύει.....	35
3.2.3 Γεωμετρική και χρονική διακριτική ικανότητα	37
3.2.4 Τεχνικά χαρακτηριστικά προϊόντων	37
3.2.5 Παραδείγματα εφαρμογής έγχρωμων σύνθετων Landsat 8	38
3.3 Βασικές Εφαρμογές	41

3.4	WELD: Web – ENABLED LANDSAT DATA	41
4.	ΤΕΧΝΟΛΟΓΙΕΣ ΑΙΧΜΗΣ ΣΤΗΝ ΤΑΞΙΝΟΜΗΣΗ ΚΑΙ ΑΝΙΧΝΕΥΣΗ ΜΕΤΑΒΟΛΩΝ ..	47
4.1	Εισαγωγή.....	49
4.2	Μέθοδοι Ανίχνευσης Μεταβολών.....	50
4.2.1	Μέθοδοι άγλεβρας εικόνας.....	50
4.2.2	Μέθοδοι μετασχηματισμού δεδομένων	54
4.2.3	Μέθοδοι στηριζόμενοι σε ταξινόμηση.....	55
4.3	Ανίχνευση Μεταβολών σε Μεγάλα Γεωχωρικά Δεδομένα.....	57
4.4	Data Mining.....	58
4.5	Μέθοδοι Ταξινόμησης και Μηχανική Μάθηση	60
4.5.1	SVM	60
4.5.2	Decision tree	66
4.5.3	K-nearest neighbors.....	69
4.5.4	Στατιστικά μοντέλα.....	73
4.6	Παραδείγματα Εφαρμογών	75
4.6.1	Παρουσίαση χρησιμοποιούμενης τεχνολογίας	76
4.6.2	SVM	77
4.6.3	Decision tree	81
4.6.4	K-nearest neighbors	85
4.7	Σύγκριση – Αξιολόγηση Αποτελεσμάτων	88
5.	ΑΝΑΠΤΥΞΗ ΜΕΘΟΔΟΛΟΓΙΑΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΜΕΤΑΒΟΛΩΝ	90
5.1	Συνοπτική Παρουσίαση Μεθοδολογίας.....	91
5.2	Προεργασία	94
5.2.1	Επιλογή Δεδομένων.....	94
5.2.2	Έλεγχος Γεωμετρίας.....	94
5.2.3	Ραδιομετρικές Διορθώσεις.....	94
5.2.4	Pansharpening	96
5.2.5	Κατασκευή Μωσαικού.....	97
5.3	Ανίχνευση και Αφαίρεση Συννέφων και Σκιών	97
5.3.1	Μέθοδοι ανίχνευσης συννέφων και σκιών	97
5.3.2	Ανάπτυξη Αλγορίθμου για την Ανίχνευση και Αφαίρεση Συννέφων και Σκιών με Βάση Διαχρονικά Γεωχωρικά Δεδομένα του Landsat 8	100
5.4	Ταξινόμηση	104
5.5	Post Classification	105
5.6	Ανίχνευση Μεταβολών	105
6.	ΠΟΣΟΤΙΚΗ ΚΑΙ ΠΟΙΟΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ	106
6.1	Επιλογή Περιοχής Μελέτης	107
6.2	Επιλογή Δεδομένων.....	107
6.3	Προεργασία Δεδομένων	110
6.3.1	Έλεγχος γεωμετρίας.....	110

6.3.2	Εφαρμογή ραδιομετρικών διορθώσεων	112
6.3.3	Εφαρμογή pansharpening	115
6.4	Δημιουργία Μωσαικού	117
6.5	Δημιουργία Μοντέλου Εκπαίδευσης	118
6.6	Επιλογή Περιοχής προς Ταξινόμηση	123
6.7	Εφαρμογή του Αλγορίθμου Ανίχνευσης και Αφαίρεσης Συννέφων και Σκιών.....	125
6.8	Εφαρμογή του Μοντέλου Εκπαίδευσης στην Περιοχή Μελέτης.....	130
6.9	Αξιολόγηση Ταξινόμησης	132
6.10	Εφαρμογή Ανίχνευσης Μεταβολών	135
6.11	Πειραματικά Αποτελέσματα στην Ευρύτερη Περιοχή της Θεσσαλίας.....	139
6.12	Ενσωμάτωση Αποτελεσμάτων στη Γεωχωρική Βάση και σε Web Gis.....	143
7.	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	145
7.1	Συμπεράσματα.....	150
7.2	Προοπτικές Εξέλιξης	151
	ΒΙΒΛΙΟΓΡΑΦΙΑ	153
	ΠΑΡΑΡΤΗΜΑ Ι	157
	ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	157
	ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ`	159
	ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ.....	162
	Orfeo Toolbox – Train Images Classifier	163
	Orfeo Toolbox – Train Lib SVM	176
	Orfeo Toolbox – Train Decision Tree	178
	Orfeo Toolbox – Train KNN	181
	Orfeo Toolbox – Bundle To Perfect Sensor.....	182
	Orfeo Toolbox – Radiometric Indices	186
	Στατιστικό μοντέλο εκμάθησης svm από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif.....	200
	Στατιστικό μοντέλο εκμάθησης decision tree από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif.....	201
	Στατιστικό μοντέλο εκμάθησης k-nearest neighbors από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif.....	204
	Στατιστικά Μοντέλου Εκπαίδευσης για την Περιοχή της Θεσσαλίας.....	212
	ΠΑΡΑΡΤΗΜΑ ΙΙ.....	230
	Μετατροπή Ψηφιακών Τιμών σε Τιμές Φαινόμενης Ανακλαστικότητας για τον Δέκτη OLI του Δορυφόρου Landsat 8 σε γλώσσα C	230
	Μετατροπή Ψηφιακών Τιμών σε Τιμές Θερμοκρασίας για το Κανάλι 10 (TIRS) του Δορυφόρου Landsat 8 σε γλώσσα C.....	233
	Μετατροπή Ψηφιακών Τιμών σε Τιμές Θερμοκρασίας για το Κανάλι 11 (TIRS) του Δορυφόρου Landsat 8 σε γλώσσα C.....	236

Αλγόριθμος Ανίχνευσης και Αφαίρεσης των Συννέφων και των Σκιών τους σε Δορυφορικές Εικόνες Landsat 8 σε γλώσσα C	239
Εξαγωγή Αποτελεσμάτων σε Web Gis με Χρήση Ανοιχτών Προτύπων	239

Κεφάλαιο 1: ΕΙΣΑΓΩΓΗ

Στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη και αξιολόγηση αλγορίθμων τηλεπισκόπησης για τη διαχείριση και ανάλυση μεγάλων δεδομένων παρατήρησης γης. Για την υλοποίηση των αλγορίθμων και την περάτωση των απαιτούμενων διαδικασιών με σκοπό την επίτευξη του παραπάνω στόχου χρησιμοποιήθηκαν λογισμικά και δεδομένα που διέπονται από τις αρχές του ανοιχτού λογισμικού και των ανοιχτών προτύπων αντίστοιχα.

Στη συνέχεια του κεφαλαίου παρουσιάζονται συνοπτικά τα βασικά χαρακτηριστικά της επιστήμης της Τηλεπισκόπησης, παρουσιάζοντας βασικές έννοιες και εφαρμογές που καλύπτει η συγκεκριμένη επιστήμη καθώς και τα δεδομένα που αξιοποιεί για όλες τις φάσεις των διαδικασιών ανάλυσης.

1.1 Τηλεπισκόπηση και Μέθοδοι Παρατήρησης Γης

Τηλεπισκόπηση είναι η επιστήμη που έχει ως στόχο την εξαγωγή ποιοτικής και ποσοτικής πληροφορίας για την επιφάνεια της γης από απόσταση, προκειμένου να μελετηθεί, ερμηνευθεί και αντιμετωπιστεί ένα φαινόμενο, αντικείμενο ή πρόβλημα. Εξάγονται, δηλαδή, συμπεράσματα για το συγκεκριμένο φαινόμενο/αντικείμενο χωρίς να είναι αναγκαία η άμεση επαφή με αυτό. Στην παραπάνω διαδικασία συμβάλλει η χρήση και η επεξεργασία δεδομένων από αισθητήρες (πολυφασματικοί, υπερφασματικοί, ραντάρ κτλ) καθώς και από πλατφόρμες λήψης (δορυφόροι, UAV, επίγειοι κτλ). Πρακτικά πρόκειται για μια διεπιστημονική διαδικασία που απαιτεί γνώσεις φωτοερμηνείας, γεωδαισίας, φωτογραμμετρίας καθώς και χρήση ηλεκτρονικών υπολογιστών κάνοντας παράλληλη χρήση οπτικών και ψηφιακών τεχνικών. [Αργιαλάς, 1998]

Για να επιτευχθεί η παρατήρηση της επιφάνειας της γης γίνεται χρήση ψηφιακών σαρωτών, οι οποίοι είναι εγκατεστημένοι σε τεχνητούς δορυφόρους, αεροσκάφη, UAV, που ανιχνεύουν την αντανάκλαση της ηλεκτρομαγνητικής ακτινοβολίας της γήινης επιφάνειας αποδίδοντας την ως ψηφιακή εικόνα. Κάθε αντικείμενο της γήινης επιφάνειας έχει μοναδικό τρόπο να αντανακλά την ηλεκτρομαγνητική ακτινοβολία σε διαφορετικά μήκη κύματος και με βάση τη φασματική συμπεριφορά του πραγματοποιείται ο διαχωρισμός και η ερμηνεία του.

(el.wikipedia.org/wiki/Τηλεπισκόπηση)

1.2 Σύγχρονες Εφαρμογές

Η τηλεπισκόπησης λόγω της φύσης της καθώς και των διεπιστημονικών αντικειμένων που συνδυάζει είναι προφανές πως η χρήση και η εφαρμογή της συναντάται στην παρατήρηση και αντιμετώπιση πολλών και διαφορετικών φαινομένων που λαμβάνουν χώρα στη γη και όχι μόνο. Τα κυριότερα από τα φαινόμενα και αντικείμενα που ερευνά είναι τα εξής: [Αργιαλάς et al, 2010]

- Καιρικές και κλιματικές αλλαγές
- Υδάτινοι πόροι
- Ανάγλυφο και γεωμορφολογία
- Είδη βλάστησης
- Πυγές Ρύπανσης
- Ανιχνεύσεις μεταβολών στις χρήσεις γης
- Ανίχνευση φυσικών καταστροφών (πυρκαγιές, πλημμύρες, κτλ)
- Γεωργία και αμπελουργία ακριβείας
- Χαρτογράφηση βιοφυσικών παραμέτρων
- Δημιουργία ψηφιακών μοντέλων εδάφους
- Ανθρωπογενείς παρεμβάσεις στο περιβάλλον

Επιπλέον, οι εφαρμογές της τηλεπισκόπησης δεν περιορίζονται αποκλειστικά στην ερμηνεία και παρατήρηση του φυσικού γήινου αναγλύφου, καθώς λόγω των τεχνολογιών που συνδυάζει, καλύπτει και επιστήμες όπως είναι η αστρονομία, η ιατρική, η βιολογία, η φυσική και άλλες. Ανάλογα, λοιπόν, με τον επιστημονικό τομέα τα δεδομένα αξιοποιούνται με διαφορετικό και μοναδικό τρόπο. Για παράδειγμα, οι μετεωρολόγοι παρακολουθούν τις κλιματικές αλλαγές και προβλέπουν τον καιρό, οι βιολόγοι μελετούν τις αλλαγές στο οικοσύστημα ενώ οι δασολόγοι ασχολούνται με την πρόγνωση και παρακολούθηση των δασικών πυρκαγιών. [el.wikipedia.org/wiki/Τηλεπισκόπηση]

1.3 Ποιοτικά Χαρακτηριστικά Δεδομένων

Όπως αναφέρθηκε η τηλεπισκόπηση αξιοποιεί σε σημαντικό βάθος ποιοτική και ποσοτική πληροφορία. Την πληροφορία αυτή την λαμβάνει από δορυφορικούς δέκτες, από αεροφωτογραφίες, από ραντάρ, από επίγειες παρατηρήσεις, από σόναρ και άλλους αισθητήρες.

Επιπλέον, τα αντίστοιχα δεδομένα μπορούν να είναι αναλογικά ή ψηφιακά ανάλογα με τον τρόπο απόκτησής τους και με το σκοπό του προβλήματος που πρέπει να επιλυθεί. Χαρακτηριστικό, επίσης, των τηλεπισκοπικών δεκτών είναι η διακριτική ικανότητα που χαρακτηρίζει κάθε μέσο απόκτησης δεδομένων. Πιο συγκεκριμένα, υπάρχουν διαφοροποιήσεις σχετικά με τη γεωμετρική διακριτική ικανότητα βάσει της οποίας καθορίζεται και η ελάχιστη επιφάνεια η οποία μπορεί να επεξεργαστεί. Στις ψηφιακές φωτογραφίες υπάρχει η αντιστοιχία του εικονοστοιχείου (pixel) με την επιφάνεια εδάφους. Διαφοροποιήσεις υπάρχουν, ακόμα, στη φασματική διακριτική ικανότητα, δηλαδή στον αριθμό των καναλιών που παράγονται οι εικόνες και δίνουν τη δυνατότητα παρακολούθησής τους σε διαφορετικά μήκη κύματος. Επίσης, η πληροφορία διαφοροποιείται ως προς την ραδιομετρική διακριτική ικανότητα (χαρακτηρίζει την ισχύ του σήματος) που σχετίζεται με τον αριθμό των δυαδικών ψηφίων (bits) που έχει η εικόνα και καθορίζει έμμεσα τις

αποχρώσεις των τόνων του γκρι που μπορούν να εμφανιστούν με την χρήση ηλεκτρονικού υπολογιστή. Τέλος, οι δέκτες διαφοροποιούνται ως προς το χρόνο που παρέχουν πληροφορία/εικόνα και πολλές φορές αυτός καθορίζει και την ανάλυση της εικόνας. Για παράδειγμα, δορυφόροι που λαμβάνουν εικόνες σε μικρά χρονικά διαστήματα παρουσιάζουν χαμηλότερη ανάλυση εικόνας σε σχέση με αντίστοιχους που λαμβάνουν εικόνες σε μεγαλύτερα διαστήματα. [Αργιαλάς *et al*, 2010]

1.4 Επεξεργασία και Ανάλυση Δεδομένων

Τα δεδομένα που χρησιμοποιούνται για εφαρμογές τηλεπισκόπησης πολλές φορές δεν είναι άμεσα αξιοποιήσιμα και απαιτείται προεπεξεργασία των δεδομένων προκειμένου να απαλειφθούν τα σφάλματα που θα οδηγούσαν σε λανθασμένα αποτελέσματα και συμπεράσματα. Πιο συγκεκριμένα, οι απαραίτητες, στο μεγαλύτερο πλήθος των εφαρμογών, διορθώσεις είναι οι γεωμετρικές, οι ραδιομετρικές και οι ατμοσφαιρικές διορθώσεις.

Όσον αφορά τις γεωμετρικές διορθώσεις, πραγματοποιείται η αναγωγή της καμπύλης της γεωμετρίας της εικόνας σε επίπεδη γεωμετρία απαλείφοντας τα σφάλματα και δίνοντας τη δυνατότητα για πραγματοποίηση μεγάλης ακρίβειας μετρήσεων. Επιπλέον, διαδικασίες γεωμετρικών διορθώσεων πραγματοποιούνται και σε περιοχές με ιδιαίτερα έντονο ανάγλυφο ή με έλλειψη πληροφορίας καθώς και σε εφαρμογές που είναι απαραίτητη η ένταξη της εικόνας σε ένα σύστημα γεωγραφικών ή προβολικών συντεταγμένων. (el.wikipedia.org/wiki/Τηλεπισκόπηση)

Στόχος των ραδιομετρικών διορθώσεων είναι η αποκατάσταση των εικόνων μέσω της διόρθωσης των ραδιομετρικών αλλοιώσεων αυτών κατά τη διάρκεια της σάρωσης, της καταγραφής και της αναπαραγωγής τους. Με τον τρόπο αυτό απαλοίφονται σφάλματα που σχετίζονται τόσο με τα τεχνικά στοιχεία του αισθητήρα όσο και με ραδιομετρικά σφάλματα που προέρχονται από την επίδραση του αναγλύφου, των συννέφων, της σκιάς συννέφων και άλλα. [Αργιαλάς, 1998]

Με τις ατμοσφαιρικές διορθώσεις απαλοίφονται σφάλματα που προέρχονται από την επίδραση της ατμόσφαιρας κατά την αντανάκλαση της ηλεκτρομαγνητικής ακτινοβολίας από τη γήινη επιφάνεια στον αντίστοιχο δέκτη. Σημαντική επίδραση στα συγκεκριμένα σφάλματα έχουν τα αέρια και υγρά συστατικά της ατμόσφαιρας που αλλοιώνουν την ποσότητα της εισερχόμενης στον αισθητήρα ακτινοβολίας. [Αργιαλάς, 1998]

Η ανάλυση των τηλεπισκοπικών δεδομένων διακρίνεται σε τρεις κατηγορίες ανάλογα με τον τρόπο που αυτά αξιοποιούνται. Πιο συγκεκριμένα, στην ποσοτική ανάλυση ζητούμενο είναι η μέτρηση μιας ιδιότητας-χαρακτηριστικού ενώ στην

περίπτωση της ποιοτικής ανάλυσης αποτυπώνεται-εντοπίζεται κάποιο χαρακτηριστικό. Τέλος, στην κατηγορία της οπτικής ανάλυσης ή φωτοερμηνείας ο αναλυτής προσπαθεί να ερμηνεύσει και εξάγει πληροφορίες για ένα αντικείμενο σύμφωνα με το χρώμα του, την υφή του, τη σχέση του αντικειμένου με τα τριγύρω και άλλα. [Αργιαλάς, 1998]

1.5 Δορυφορικοί Τηλεπισκοπικοί Δέκτες

Οι τηλεπισκοπικοί δέκτες, ανάλογα με το είδος της ηλεκτρομαγνητικής ακτινοβολίας που ανακλάται και ανιχνεύεται από αυτούς, μπορούν να αναλυθούν σε δύο είδη: τους ενεργητικούς και τους παθητικούς. Οι παθητικοί ανιχνεύουν ηλεκτρομαγνητική ακτινοβολία προερχόμενη από μία φυσική πηγή (π.χ. ήλιος) ενώ οι ενεργητικοί χρησιμοποιούν τη δική τους πηγή ακτινοβολίας για να φωτίσουν το αντικείμενο. Οι δορυφορικοί τηλεπισκοπικοί δέκτες ανήκουν στην κατηγορία των παθητικών δεκτών, οι οποίοι αποτελούνται από δορυφόρους που χαρακτηρίζονται από προκαθορισμένη τροχιά και διάρκεια πτήσης ικανή για την παρακολούθηση της γης καθώς και τον αντίστοιχο τηλεπισκοπικό δέκτη. [Καρτάλης & Φείδας, 2007]

Η επιλογή του κατάλληλου δορυφορικού δέκτη πραγματοποιείται ανάλογα με την κλίμακα και τα χαρακτηριστικά της εφαρμογής. Για παράδειγμα, υπάρχουν εφαρμογές όπου η ακρίβεια των μετρήσεων παίζει σημαντικό ρόλο οπότε επιλέγονται δορυφορικοί δέκτες με έμφαση στη χωρική διακριτική ικανότητα ενώ άλλες φορές δίνεται περισσότερη έμφαση στην ποιοτική ή θεματική πληροφορία. [Καρτάλης & Φείδας, 2007]

Επιλεκτικά, μερικοί από τους πιο γνωστούς δορυφορικούς δέκτες που χρησιμοποιούνται στην Τηλεπισκόπηση είναι οι παρακάτω:

- Advanced Very High Resolution Radiometer (AVHRR) των δορυφόρων NOAA
- MODerate Resolution Imaging Spectroradiometer (MODIS) των δορυφόρων Terra και Aqua
- Landsat 8 της USGS
- High Resolution Visible (HRV) δορυφόρων SPOT 7
- Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) του δορυφόρου Terra
- Οι δορυφορικοί δέκτες πολύ υψηλής χωρικής διακριτικής ικανότητας Ikonos και Quickbird
- Sentinel 1, ESA

1.6 Λογισμικά Επεξεργασίας

Τα ψηφιακά δεδομένα τηλεπισκόπησης μπορούν να ερμηνευθούν και να επεξεργαστούν από προγράμματα ηλεκτρονικών υπολογιστών, τα οποία διαχειρίζονται δεδομένα που έχουν καταγραφεί σε μορφή εικονοστοιχείων. Σε

γενικές γραμμές, τα λογισμικά τηλεπισκόπησης διαχωρίζονται ως προς δύο διαφορετικά επίπεδα ανάλογα με το αν είναι εμπορικά ή ελεύθερα καθώς και με το αν πραγματοποιούν αλγόριθμους με βάση το εικονοστοιχείο ή το αντικείμενο ή και τα δύο. Ακολουθεί διάκριση των πιο γνωστών λογισμικών ανάλογα με το αν είναι εμπορικά ή ελεύθερα και περιγράφονται τα κυριότερα χαρακτηριστικά τους.

1.6.1 Ελεύθερα λογισμικά

Ένας σύντομος ορισμός του ελεύθερου λογισμικού είναι ο εξής: Ελεύθερο λογισμικό είναι το λογισμικό που μπορεί να χρησιμοποιηθεί, αντιγραφεί, μελετηθεί, τροποποιηθεί και αναδιανεμηθεί χωρίς περιορισμό και χωρίς κόστος. Οι αρχές, λοιπόν, που διέπουν το ελεύθερο λογισμικό είναι οι εξής:

- Χρησιμοποιείται ελεύθερα για κάθε χρήση
- Διατίθεται ο πηγαίος του κώδικας
- Διανέμεται ελεύθερα και χωρίς περιορισμούς για την αντιγραφή του
- Μπορεί να τροποποιηθεί από τον οποιονδήποτε, έτσι ώστε να επιτευχθεί βελτίωσή του

Η χρήση των ελεύθερων λογισμικών χαρακτηρίζεται από πληθώρα πλεονεκτημάτων. Μερικά από τα πιο σημαντικά είναι τα εξής:

1. Ελεύθερη και δωρεάν διανομή
2. Βελτίωση του λογισμικού και εξάπλωση της τεχνογνωσίας
3. Τροποποίηση του πηγαίου κώδικα σύμφωνα με τις ατομικές ανάγκες και προτιμήσεις του κάθε χρήστη
4. Παρέχεται πολλαπλός έλεγχος του κώδικα καθώς ο κώδικας είναι ανοιχτός σε όλους τους χρήστες καθιστώντας πιο εύκολη τη διαδικασία εκσφαλμάτωσης (debugging)

[www.gnu.org/philosophy/free-sw.html]

Ακολουθεί η παράθεση των πιο γνωστών και αποτελεσματικών ελεύθερων λογισμικών που χρησιμοποιούνται στις τηλεπισκοπικές εφαρμογές και ελέγχονται αν και ποια από τα παρακάτω χαρακτηριστικά καλύπτουν.

Σύγχρονα χαρακτηριστικά τηλεπισκοπικών λογισμικών

- [1]: Επεξεργασία raster/vector
- [2]: Διαχείριση και ανάλυση γεωχωρικών δεδομένων
- [3]: Γραφικά και παραγωγή χαρτών
- [4]: Χωρική μοντελοποίηση
- [5]: Απεικόνιση
- [6]: Ανάλυση τρισδιάστατων γεωχωρικών δεδομένων
- [7]: Αλγόριθμοι μηχανικής μάθησης
- [8]: Αντικειμενοστραφή ανάλυση

[9]: Εξειδικευμένοι αλγόριθμοι για συγκεκριμένους δορυφόρους

[10]: Δυνατότητα επεξεργασίας ποικίλων φορμάτ εικόνων (Gtiff, ERS, HDF κτλ)

Λογισμικό	Χαρακτηριστικά/ Δυνατότητες	Σύντομη Περιγραφή	Ιστότοπος
GRASS	[1], [2], [3], [4], [5], [6], [9], [10]	- Γλώσσες πηγαίου κώδικα: C, C++, Python, Tcl - Διαθέσιμο σε Linux, Mac OSX, Windows - Gui, command line - Δυνατότητα ενσωμάτωσης εξωτερικών λογισμικών γεωστατιστικών π.χ. R	grass.osgeo.org
SAGA GIS	[1], [2], [3], [4], [5], [6],[10]	- Γλώσσα πηγαίου κώδικα: C++ - Διαθέσιμο σε Linux, Mac OSX, Windows - Gui, command line	www.saga-gis.org/en/index.html
ORFEO TOOLBOX	[1], [2], [3], [4], [5], [6], [7], [10]	- Γλώσσα πηγαίου κώδικα: C++ - Διαθέσιμο και σε python, java, IDL - Διαθέσιμο σε Linux, Mac OSX, Windows - Gui, command line	www.orfeo-toolbox.org/otb/
OPTICKS	[1], [2], [3], [4], [5], [6],[10]	- Γλώσσες πηγαίου κώδικα: C++, python - Διαθέσιμο σε Linux, Solaris, Windows - Gui - Δυνατότητα ενσωμάτωσης προσθέτων	www.opticks.org/confluence/display/opticks/Welcome+To+Opticks
BEAM	[1], [2], [3], [4], [5], [6],[9],[10]	- Γλώσσα πηγαίου κώδικα: java - Διαθέσιμο σε Linux, Mac OSX, Windows - Gui	www.brockmann-consult.de/cms/web/beam/
QGIS	[1], [2], [3], [4], [5], [6], [7], [9], [10]	- Γλώσσες πηγαίου κώδικα: C++, python, QT - Διαθέσιμο σε Linux, Mac OSX, Windows - Gui, command line - Δυνατότητα ενσωμάτωσης μεγάλου πλήθους προσθέτων - Ενσωμάτωση αλγορίθμων από grass, saga gis, orfeo toolbox	www.qgis.org/en/site/

Πίνακας 1.1: Χαρακτηριστικά ελεύθερων λογισμικών Τηλεπισκόπησης

1.6.2 Εμπορικά λογισμικά

Αντίστοιχα με τα ελεύθερα λογισμικά έχουν αναπτυχθεί πολλά εμπορικά λογισμικά, τα οποία χρησιμοποιούν μεγάλο πλήθος εξειδικευμένων τηλεπισκοπικών αλγορίθμων. Σε αντίθεση με τα ελεύθερα λογισμικά η απόκτησή τους δεν είναι δωρεάν και υπάρχουν συγκεκριμένοι περιορισμοί σχετικά με τη χρήση και τη διανομή τους.

Στον επόμενο πίνακα παρουσιάζονται τα σημαντικότερα από αυτά με τα αντίστοιχα χαρακτηριστικά τους κατά περίπτωση.

Λογισμικό	Χαρακτηριστικά/ Δυνατότητες	Σύντομη Περιγραφή	Ιστότοπος
ERDAS IMAGINE	[1], [2], [3], [4], [5], [6], [9], [10]	▪ Γλώσσες πηγαίου κώδικα: C, C++ ▪ Διαθέσιμο σε Windows ▪ Gui	www.hexagongeospatial.com/products/ERDAS-IMAGINE/Details.aspx
ENVI	[1], [2], [3], [4], [5], [6], [9], [10]	▪ Γλώσσα πηγαίου κώδικα: IDL ▪ Διαθέσιμο σε Windows ▪ Gui ▪ Ενσωμάτωσή του στο ArcGIS	http://www.exelisvis.com/ProductsServices/ENVI/ENVI.aspx
ERMapper	[1], [2], [3], [4], [5], [6], [10]	▪ Διαθέσιμο σε Windows ▪ Gui ▪ Υποστηρίζει παραγωγή scripts στη C γλώσσα προγραμματισμού	http://www.hexagongeospatial.com/products/ERDAS-ERMapper/Details.aspx
eCognition	[1], [2], [3], [4], [5], [6], [8], [10]	▪ Εξειδικευμένο πρόγραμμα αντικειμενοστραφούς ανάλυσης ▪ Διαθέσιμο σε Windows ▪ Gui ▪ Δυνατότητα αυτοματοποίησης διαδικασιών μέσω scripting	http://www.ecognition.com
PCI	[1], [2], [3], [4], [5], [6], [9], [10]	▪ Διαθέσιμο σε Linux, Windows ▪ Gui ▪ Δυνατότητα αυτοματοποίησης διαδικασιών μέσω scripting	http://www.pcigeomatics.com
ArcGIS	[1], [2], [3], [4], [5], [6],[10]	▪ Διαθέσιμο σε Windows ▪ Gui ▪ Δυνατότητα αυτοματοποίησης διαδικασιών μέσω scripting σε python	http://www.esri.com/software/arcgis

Πίνακας 1.2: Χαρακτηριστικά εμπορικών λογισμικών Τηλεπισκόπησης

Στις εφαρμογές της Τηλεπισκόπησης αξιοποιούνται τις περισσότερες φορές μεγάλου όγκου δεδομένα, ακόμα και από διαφορετικούς δέκτες, καθιστώντας δύσκολη την επεξεργασία και ανάλυσή τους. Λαμβάνοντας το παραπάνω υπόψη, έγινε προσπάθεια, στη συγκεκριμένη εργασία, να αναπτυχθεί μεθοδολογική προσέγγιση για τη διαχείριση και ανάλυση μεγάλων γεωχωρικών δεδομένων, αξιοποιώντας δορυφορικά τηλεπισκοπικά δεδομένα του δορυφόρου Landsat 8.

Ακολουθεί η διάρθρωση της εργασίας:

- **Κεφάλαιο 2:** Στο συγκεκριμένο κεφάλαιο πραγματοποιείται εισαγωγή στην έννοια των μεγάλων γεωχωρικών δεδομένων (Big Geospatial Data) και αναφέρονται τα βασικά χαρακτηριστικά τους. Στη συνέχεια, παρουσιάζεται συνοπτικά το εύρος των αντικειμένων και εφαρμογών, στις οποίες αυτά εμφανίζονται και δίνονται αντίστοιχα παραδείγματα σε τομείς όπως το διαδίκτυο, τα GIS και η τηλεπισκόπηση. Παράλληλα, περιγράφονται τα λογισμικά που χρησιμοποιούνται με στόχο την αποθήκευση και διαχείριση τέτοιων δεδομένων, εντοπίζονται οι σημαντικότερες δυσκολίες που συνεπάγεται η χρήση τους, ενώ τέλος αναφέρονται οι τελευταίες τεχνολογικές εξελίξεις και προοπτικές ανάπτυξης στον τομέα της διαχείρισης και επεξεργασίας τους.
- **Κεφάλαιο 3:** Σε αυτό το κεφάλαιο παρουσιάζονται τα βασικά χαρακτηριστικά του προγράμματος Landsat της USGS, καθώς και εφαρμογές για τις οποίες ενδείκνυται η χρήση των προϊόντων του συγκεκριμένου προγράμματος. Παράλληλα, περιγράφεται μία εφαρμογή που αξιοποιεί μεγάλα δεδομένα που έχουν προέλθει από το πρόγραμμα δορυφόρων της Landsat.
- **Κεφάλαιο 4:** Στο συγκεκριμένο κεφάλαιο παρουσιάζονται τεχνολογίες αιχμής που αξιοποιούνται στις διαδικασίες ταξινόμησης και ανίχνευσης μεταβολών. Πιο συγκεκριμένα, παρουσιάζονται αλγόριθμοι ταξινόμησης που αξιοποιούν διαδικασίες μηχανικής μάθησης για την αντιμετώπιση προβλημάτων που σχετίζονται με την εξόρυξη δεδομένων (Data Mining). Στη συνέχεια, πραγματοποιούνται εφαρμογές τέτοιων αλγορίθμων και ακολουθεί αξιολόγηση των αποτελεσμάτων για κάθε μέθοδο που χρησιμοποιήθηκε.
- **Κεφάλαιο 5:** Σε αυτό το κεφάλαιο παρουσιάζεται και αναλύεται μεθοδολογική προσέγγιση που ως τελικό στόχο έχει την ανίχνευση μεταβολών με αλγόριθμο επιβλεπόμενης ταξινόμησης αξιοποιώντας διαδικασίες μηχανικής μάθησης σε μεγάλα γεωχωρικά δεδομένα. Στα πλαίσια της συγκεκριμένης μεθοδολογίας αναπτύχθηκε και αξιολογήθηκε αλγόριθμος ανίχνευσης και αφαίρεσης συννέφων και σκιών, αξιοποιώντας διαχρονικά δεδομένα του δορυφόρου Landsat 8.
- **Κεφάλαιο 6:** Στο κεφάλαιο αυτό παρουσιάζονται και αξιολογούνται τα αποτελέσματα από την εφαρμογή της μεθοδολογίας που ακολουθήθηκε. Οι εφαρμογές που παρουσιάζονται αφορούν στην ανίχνευση μεταβολών τυχαίας περιοχής της ευρύτερης περιοχής της Θεσσαλίας καθώς και στην ταξινόμηση της περιφέρειας της Θεσσαλίας, αξιοποιώντας και στις δύο περιπτώσεις τη

μεθοδολογική προσέγγιση που αναπτύχθηκε. Τέλος, τα αποτελέσματα που προέκυψαν από την μεθοδολογική προσέγγιση που αναπτύχθηκε παρουσιάστηκαν μέσω δοκιμαστικής έκδοσης εφαρμογής σε διαδικτυακό γεωπληροφοριακό σύστημα.

- **Κεφάλαιο 7:** Στο συγκεκριμένο κεφάλαιο παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εφαρμογή της μεθοδολογίας που αναπτύχθηκε, επισημαίνονται τα σημαντικότερα προβλήματα που παρουσιάστηκαν στα διάφορα στάδια της διαδικασίας ανάλυσης, ενώ τέλος δίνονται κατευθύνσεις για περαιτέρω έρευνα σε θέματα που αφορούν τη βελτίωση της μεθοδολογικής προσέγγισης που ακολουθήθηκε.
- **Παραρτήματα:** Στο Παράρτημα I βρίσκονται οι πηγαίοι κώδικες των αλγορίθμων που αξιοποιήθηκαν καθ' όλη τη διάρκεια της παρούσας εργασίας, καθώς και τα στατιστικά μοντέλα εκμάθησης που προέκυψαν από τους διάφορους αλγορίθμους ταξινόμησης. Στο Παράρτημα II βρίσκονται οι πηγαίοι κώδικες των αλγορίθμων που αναπτύχθηκαν στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας.

Κεφάλαιο 2: Μεγάλα Γεωχωρικά Δεδομένα Παρατήρησης Γης

2.1 Μεγάλα Δεδομένα – Γενικά στοιχεία

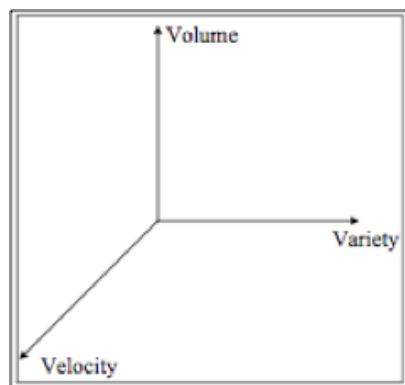
Η έννοια των Big Data ορίζεται ευρέως με τη χρήση των τριών “V”s. Σύμφωνα με τον ορισμό της εταιρείας Gartner ισχύουν τα εξής: [Gartner Inc., 2011]

Volume - Όγκος: Η τεράστια αύξηση του όγκου των δεδομένων στα πλαίσια των υπολογιστικών συστημάτων προκαλείται τόσο από τους ήδη υπάρχοντες παραδοσιακούς τύπους δεδομένων όσο και από νέους τύπους. Ο τόσο μεγάλος όγκος δεδομένων καθώς και πληροφοριών δεν δημιουργούν απλά προβλήματα αποθήκευσης όσο και σοβαρά θέματα στις διαδικασίες ανάλυσης.

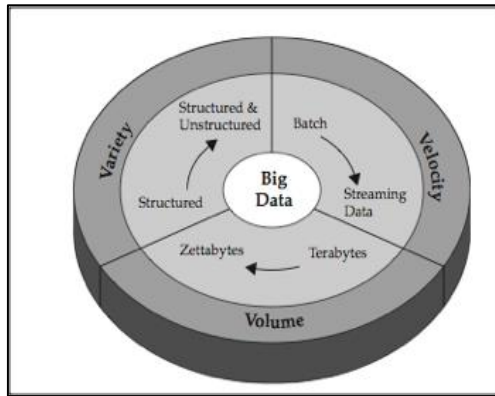
Velocity – Ταχύτητα: Πρόκειται για ρεύματα πολλαπλών δεδομένων, τα οποία είναι αυστηρά δομημένα και άμεσα διαθέσιμα για πρόσβαση και παράδοση. Επιπλέον, η έννοια της ταχύτητας σχετίζεται τόσο με τον ρυθμό παραγωγής δεδομένων όσο και με τον ρυθμό που πρέπει τα δεδομένα να υποβάλλονται σε επεξεργασία για να καλύψουν τη ζήτηση.

Variety – Ποικιλία: Μεγάλο ζήτημα του τόσο μεγάλου όγκου δεδομένων είναι η «μετάφραση» των πληροφοριών σε αποφάσεις, λόγω του γεγονότος ότι υπάρχουν πολλά διαφορετικά είδη πληροφορίας προς ανάλυση. Η ποικιλία συναντάται τόσο σε μορφή πινάκων (βάσεων δεδομένων), εγγράφων, βίντεο, εικόνων, ήχου ακόμα και σε απλά καθημερινά πράγματα όπως είναι η πληροφορία στο διαδίκτυο μέσω ιστοτόπων, διαδικτυακών ταχυδρομείων και άλλα.

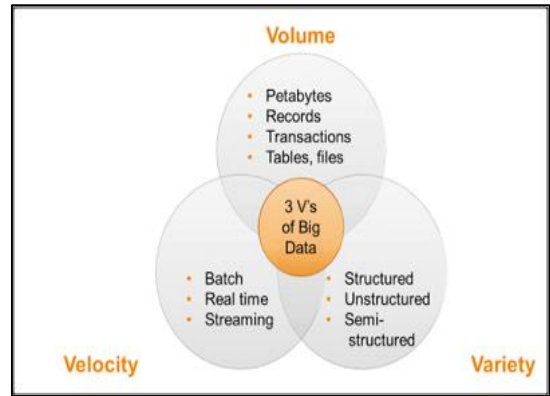
Ακολουθούν σχηματικές παρουσιάσεις του ορισμού των Big Data ως προς τις τρεις μεταβλητές που αναφέρθηκαν:



Εικόνα 2.1: Ορισμός 1 (V^3) των χαρακτηριστικών των Big Data
[Suthaharan, 2013]



Εικόνα 2.2: Ορισμός 2 (V^3) των χαρακτηριστικών των Big Data
[IBM, 2011]



Εικόνα 2.3: Ορισμός 3 (V^3) των χαρακτηριστικών των Big Data
[The 3 V's of Big Data and their Technologies, www.geektime.com, 2013]

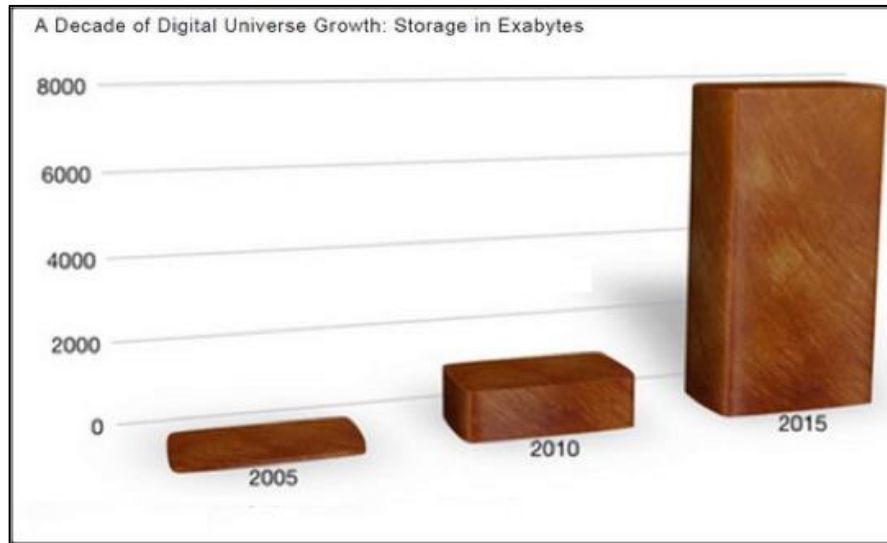
Σύμφωνα με τον ορισμό των Big Data γίνεται αντιληπτό ότι κάποια χρονική στιγμή όταν ο όγκος, η ποικιλία και η ταχύτητα των στοιχείων αυξηθούν σημαντικά, οι σημερινές τεχνικές και τεχνολογίες μπορεί να μην είναι σε θέση να διαχειριστούν την αποθήκευση και την επεξεργασία των δεδομένων. Στο σημείο αυτό ορίζεται η έννοια των Big Data. Προκειμένου, λοιπόν, να υπάρχει επιτυχής ανάλυση και κατανόηση των χαρακτηριστικών των δεδομένων και να εξαχθούν χρήσιμα γεωμετρικά και στατιστικά στοιχεία έχει αναπτυχθεί η έννοια των Big Data Analytics. [Suthaharan, 2013]

Οι προκλήσεις αυτές περιλαμβάνουν τη σύλληψη, επιμέλεια, αποθήκευση, αναζήτηση, επεργασία, κοινή χρήση, μεταφορά, ανάλυση και οπτικοποίηση των δεδομένων. Η ανάλυση μεγάλων δεδομένων απαιτεί το διαχωρισμό του συνόλου των δεδομένων σε μικρότερα υποσύνολα δημιουργώντας συσχετίσεις μεταξύ τους. Με τον τρόπο αυτό εξάγεται και αξιοποιείται ποιοτική πληροφορία που προέρχεται μέσα από τους συσχετισμούς των υποσυνόλων των δεδομένων.

[http://en.wikipedia.org/wiki/Big_data]

Σύμφωνα με την International Data Corporation (IDC) παρατηρείται τρομερή αύξηση του ψηφιακού περιεχομένου που συγκεντρώνουν οι περισσότερες επιχειρήσεις και ο συγκεκριμένος όγκος δημιουργεί ανησυχία στα θέματα διαχείρισης και επεξεργασίας των δεδομένων. Σύμφωνα με έρευνα της IDC παρατηρήθηκε αύξηση του ψηφιακού περιεχομένου στον κόσμο από το 2011 στο 2012 σε ποσοστό 48% που μεταφράζεται σε αύξηση 2,7 δισεκατομμυρίων terabytes. Επίπλέον, σύμφωνα με τη συγκεκριμένη έρευνα εκτιμάται πως το 2015 το ψηφιακό περιεχόμενο θα φτάσει τα 8 δισεκατομμύρια terabytes. Παρατηρήθηκε επίσης πως το 90% του συνολικού ψηφιακού περιεχομένου στον κόσμο δημιουργήθηκε τα τελευταία δύο χρόνια κάτι που συντέλεσε και η αυξανόμενη αποθήκευση δεδομένων λόγω μείωσης του κόστους αποθήκευσης. [IDC Research, 2011]

Ακολουθεί το διάγραμμα σύμφωνα με την έρευνα που πραγματοποιήθηκε από την IDC:



Διάγραμμα 2.1: Αύξηση ψηφιακού περιεχομένου
IDC's Digital Universe Study, sponsored by EMC, June 2011

2.2 Εφαρμογές σε Μεγάλα Δεδομένα

Λαμβάνοντας υπόψιν τον ορισμό και τα χαρακτηριστικά των Big Data καθώς και την ραγδαία ανάπτυξη της τεχνολογίας γίνεται αντιληπτό πως οι εφαρμογές τους συναντώνται σε πολλούς, αν όχι όλους, τους τομείς της ζωής. Στο συγκεκριμένο κεφάλαιο παρουσιάζονται συνοπτικά κάποιοι χαρακτηριστικοί τομείς που διαχειρίζονται μεγάλου όγκου δεδομένα καθώς και ο τρόπος που αξιοποιούν την πληροφορία.

Big Data και διαδίκτυο

Ο χώρος του διαδικτύου ίσως αποτελεί ένα από τα πιο αντιπροσωπευτικά παραδείγματα, στον οποίο γίνεται αντιληπτή η ύπαρξη των Big Data. Μεγάλες εταιρείες αναγκάζονται να συλλέγουν και να επεξεργάζονται μεγάλου όγκου δεδομένα σε καθημερινή βάση. Πιο συγκεκριμένα, ιστοχώροι όπως το facebook και το twitter συλλέγουν καθημερινά 25+ terabytes και 12+ terabytes δεδομένων αντίστοιχα, σύμφωνα με έρευνα της IBM το 2012. Επιπλέον, σύμφωνα με την ίδια έρευνα, η εταιρεία Google μέσω των διάφορων εφαρμογών της (mail, google drive, google earth και άλλων) συλλέγει δεδομένα όγκου μεγαλύτερου των 80 terabytes ημερησίως. *[University of California, Berkeley, 2012]*

Big Data και εμπόριο

Χαρακτηριστικά παραδείγματα ύπαρξης και αντιμετώπισης προβλημάτων στα Big Data εμφανίζονται και σε εταιρείες που ασχολούνται με το εμπόριο καθώς και σε διαδικτυακούς ιστοτόπους που ασχολούνται με εμπορικές λειτουργίες. Πιο συγκεκριμένα, η εταιρεία ebay που ασχολείται με το ηλεκτρονικό εμπόριο,

συναλλάσσεται με περισσότερους από 108 εκατομμύρια πελάτες και δέχεται καθημερινά περισσότερα από 250 εκατομμύρια αιτήματα στον ιστοχώρο της. Παρόμοιες καταστάσεις αντιμετωπίζουν και εταιρείες που ασχολούνται με το εμπόριο ηλεκτρονικών συσκευών, όπου σύμφωνα με το πανεπιστήμιο του Berkeley στην Καλιφόρνια υπολογίζεται πως κάθε 5 δευτερόλεπτα πωλείται ένα κινητό τηλέφωνο. Από το παραπάνω γίνεται αντιληπτός ο όγκος πληροφορίας που πρέπει να αποθηκεύει κάθε εταιρεία τόσο για τους πελάτες της όσο και για τα προϊόντα της. [University of California, Berkeley, 2012]

Big Data και φορείς υγείας

Ακολουθεί πίνακας που απεικονίζει τον ετήσιο όγκο δεδομένων σε όλες τις μορφές διαγνωστικής απεικόνισης στα πλαίσια των πρακτικών ακτινολογίας στις ΗΠΑ.

Modality	Part B non HMO	All Medicare	All Population	Per 1000 people	Ave study size (GB)	Total annual data generated in GB
CT	22 million	29 million	87 million	287	0.25	21,750,000
MR	7 million	9 million	26 million	86	0.2	5,200,000
Ultrasound	40 million	53 million	159 million	522	0.1	15,900,000
Interventional	10 million	13 million	40 million	131	0.2	8,000,000
Nuclear Medicine	10 million	14 million	41 million	8	0.1	4,100,000
PET	1 million	1 million	2 million	8	0.1	200,000
Xray, total incl. Mammography	84 million	111 million	332 million	1,091	0.04	13,280,000
All Diagnostic Radiology	174 million	229 million	687 million	2,259	0.1	68,700,000
Σύνολο						68.7 PETabytes

Πίνακας 2.1: Δεδομένα διαγνωστικής απεικόνισης στις ΗΠΑ
Indiana University Pervasive Technology Institute, 2011

2.3 Εφαρμογές Μεγάλων Δεδομένων στα GIS

Μεγάλου πλήθους εφαρμογών έχουν, επίσης, αναπτυχθεί με την ταυτόχρονη αξιοποίηση των μεγάλων γεωχωρικών δεδομένων και των γεωγραφικών πληροφοριακών συστημάτων. Στο συγκεκριμένο κεφάλαιο παρουσιάζονται χαρακτηριστικά παραδείγματα εφαρμογών που επεξεργάζονται μεγάλου όγκου γεωχωρικά δεδομένα προκειμένου να αξιοποιήσουν την πληροφορία με τον αποδοτικότερο δυνατό τρόπο.

Big Data και αντιμετώπιση καταστροφών

Δημιουργία χαρτών μέσω πληροφορίας που προέρχεται από τα μέσα μαζικής ενημέρωσης και το κοινό (εικόνες, βίντεο και μηνύματα) προκειμένου να χαρτογραφηθεί ο τόπος που συνέβη μία καταστροφή, η σοβαρότητα αυτής καθώς και η εύρεση της βέλτιστης διαδρομής για την άμεση παρέμβαση των αρμόδιων υπηρεσιών αν αυτό κριθεί αναγκαίο. Χαρακτηριστικό παράδειγμα της συγκεκριμένης εφαρμογής είναι η αξιοποίηση μεγάλων γεωχωρικών δεδομένων στη διαδικασία ανάλυσης προκειμένου να γίνει έγκαιρη πρόγνωση του τυφώνα Irene στην Φλόριντα το 2011 και ελαχιστοποίηση των συνεπειών. [ESRI, *Geospatial Visualization and Big Data*, 2011]

Big Data και οικονομικά

Ορισμένες από τις εταιρείες που προσφέρουν στους καταναλωτές χρηματοοικονομικές υπηρεσίες χαρτογραφούν δεδομένα τοποθεσίας μέσω των συστημάτων συναλλαγών (RDBMS), των πληροφοριών των πελατών (CRM) καθώς και των μέσων μαζικής ενημέρωσης, προκειμένου να αποκτήσουν πλήρη στοιχεία για τους χρήστες των υπηρεσιών τους. Με τον τρόπο αυτό, οι εν λόγω εταιρείες μπορούν να διαχειριστούν περιπτώσεις απάτης ή να προσφέρουν αποτελεσματικότερες υπηρεσίες μάρκετινγκ. [ESRI, *Geospatial Visualization and Big Data*, 2011]

Big Data και κυβέρνηση

Υπηρεσίες ομοσπονδιακού, πολιτειακού και τοπικού επιπέδου χρησιμοποιούν μεγάλα δεδομένα σε ποικίλες χωρικές αναλύσεις. Οι ομοσπονδιακές κυβερνητικές υπηρεσίες χρησιμοποιούν συστήματα γεωγραφικών πληροφοριών (GIS) και μεγάλα δεδομένα για πολλά είδη δραστηριοτήτων, όπως για παράδειγμα την ανάλυση της θέσης σε πραγματικό χρόνο, την ανάλυση δεδομένων από αισθητήρες, την ανίχνευση πιθανής απάτης καθώς και την παρακολούθηση εξάπλωσης των ασθενειών. Σε οργανισμούς τοπικής αυτοδιοίκησης, η τεχνολογία των μεγάλων δεδομένων χρησιμοποιείται, για παράδειγμα, στη βελτίωση της αποτελεσματικότητας των υπηρεσιών που προσφέρουν τα τηλεφωνικά κέντρα, καθώς επίσης και στη διαδικασία υποβολής αιτήσεων στις υπηρεσίες της πόλης, οι οποίες σχετίζονται με τη διαχείριση περιουσιακών στοιχείων. [ESRI, *Geospatial Visualization and Big Data*, 2011]

Big Data και ασφάλεια

Η χαρτογράφηση μιας τοποθεσίας μέσω των μέσων μαζικής ενημέρωσης κατά τη διάρκεια φυσικών καταστροφών βοηθά τις ασφαλιστικές εταιρείες να παρακολουθούν τις οικονομικές επιπτώσεις και να βοηθούν τους πελάτες τους εφόσον αυτό κριθεί αναγκαίο. Επιπλέον, μέσω της ανάλυσης μεγάλων γεωχωρικών

δεδομένων είναι σε θέση να ανιχνεύουν απάτες και να ελέγχουν την ακρίβεια των ισχυρισμών. *[ESRI, Geospatial Visualization and Big Data, 2011]*

Big Data και φυσικοί πόροι

Η βιομηχανία πετρελαίου είχε εμπλακεί βαθύτατα στα μεγάλα δεδομένα πριν ακόμη δοθεί ο ορισμός για αυτά. Πιο συγκεκριμένα, υπάρχουν terabytes δεδομένων από θέσεις που σχετίζονται με σεισμικά φαινόμενα συλλεγμένα από όλον τον κόσμο και είναι διαθέσιμα στη βιομηχανία πετρελαίου για δραστηριότητες εξερεύνησης και εξόρυξης πετρελαίου. *[ESRI, Geospatial Visualization and Big Data, 2011]*

Big Data και λιανεμπόριο

Ένα νέο είδος εφαρμογής που αξιοποιεί τα γεωγραφικά συστήματα πληροφοριών και την τεχνολογία των μεγάλων δεδομένων σε σχέδια μαζικού λιανοεμπορίου παρακολουθεί και «φιλτράρει» τις πληροφορίες από τα μέσα ενημέρωσης προκειμένου να εντοπίσει την ιδανική θέση εγκατάστασης καταστημάτων. *[ESRI, Geospatial Visualization and Big Data, 2011]*

Big Data και τηλεπικοινωνίες

Τα τηλεφωνικά κέντρα συγκεντρώνουν τεράστιες ποσότητες δεδομένων, εκ των οποίων ένα μεγάλο μέρος είναι αδόμητο ή σε μικρό βαθμό δομημένο. Με την ταξινόμηση και την χαρτογράφηση των κλήσεων συχνά ανιχνεύονται πρότυπα που υποδεικνύουν αδυναμίες στις αντίστοιχες υποδομές. *[ESRI, Geospatial Visualization and Big Data, 2011]*

Big Data και υποδομές

Έξυπνες εφαρμογές πλέγματος παράγουν τεράστιες ποσότητες δεδομένων που ανιχνεύουν τη χρησιμότητα των υποδομών που είναι σε λειτουργία. Τα συγκεκριμένα δεδομένα παράγονται σε πραγματικό χρόνο και προβάλλονται σε χάρτες που δείχνουν το επίπεδο χρήσης των υποδομών και βοηθούν τους διαχειριστές να αντιμετωπίσουν έγκαιρα πιθανές βλάβες. Για παράδειγμα, η συγκεκριμένη χωρική ανάλυση του δικτύου πλέγματος μπορεί να εντοπίσει τα περιουσιακά στοιχεία που θα μπορούσαν να επηρεαστούν αρνητικά από την υψηλή χρήση σε βάθος χρόνου. *[ESRI, Geospatial Visualization and Big Data, 2011]*

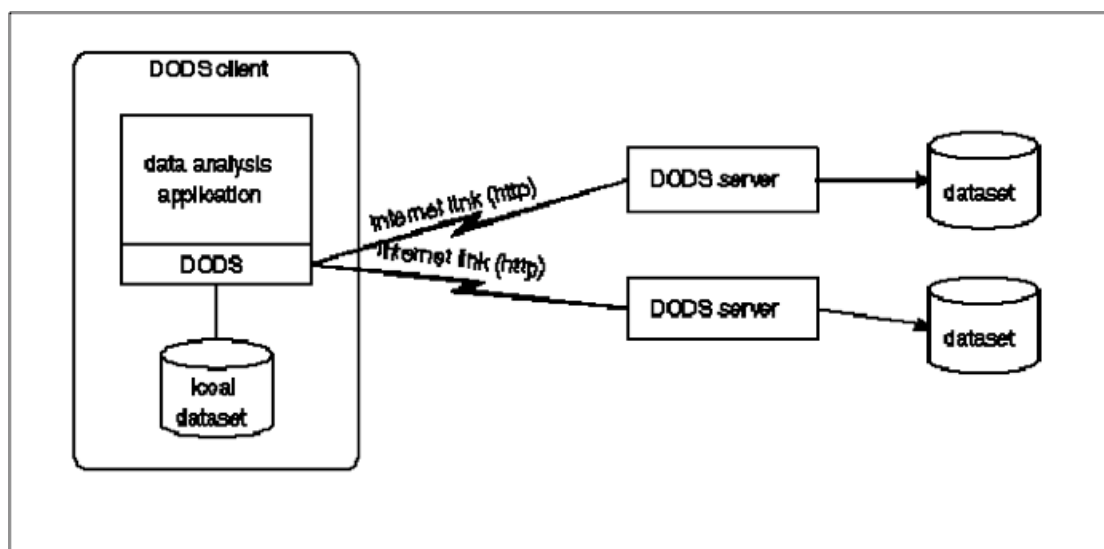
2.4 Εφαρμογές Μεγάλων Δεδομένων στην Τηλεπισκόπηση

Όπως αναφέρθηκε στο κεφάλαιο 1 της παρούσας εργασίας η Τηλεπισκόπηση είναι η επιστήμη που εκμεταλλεύεται σε μεγάλο βαθμό δεδομένα που παραγονται από δέκτες που είναι τοποθετημένοι σε δορυφόρους, αεροσκάφη καθώς και επίγειους σταθμούς. Επιπλέον, τα δεδομένα ανάλογα από το δέκτη που έχουν παραχθεί διαφέρουν στη γεωμετρική, φασματική και ραδιομετρική διακριτική ικανότητά τους

καθώς και στο χρονικό βήμα που είναι διαθέσιμα προς χρήση. Για το λόγο αυτό έχουν αναπτυχθεί και χρησιμοποιηθεί πολλαπλοί δέκτες από διάφορες εταιρείες προκειμένου να καλυφθούν όλοι οι δυνατοί συνδυασμοί και να παράγονται συνεχή δεδομένα που καλύπτουν την επιφάνεια της γης σε όλες τις δυνατές χρονικές περιόδους παράγοντας έτσι ένα τεράστιο όγκο γεωχωρικών δεδομένων. Για παράδειγμα, το πρόγραμμα της NASA: Earth Science Program αποτελείται από μια σειρά δορυφόρων (15 στο σύνολο) με ονομασία Earth Observing System Data and Information System (EOSDIS) που με τη λειτουργία του μέχρι το 2012 έχει δημιουργηθεί αρχείο δεδομένων που ξεπερνά τα 7,5 petabytes δεδομένων. Επιπλέον, χαρακτηριστικό παράδειγμα διαχείρισης μεγάλων γεωχωρικών δεδομένων είναι το πρόγραμμα Sentinel της ESA, το οποίο παράγει περίπου 7 Terabytes ημερησίως. *[Earth Imaging Journal, 2013]*

2.4.1 Distributed Oceanographic Data Systems (DODS)

Τα κατανεμημένα συστήματα ωκεανογραφικών δεδομένων (dods) αποτελούν ένα ολοκληρωμένο λογισμικό που επιτρέπει την παροχή δεδομένων μεταξύ των φορέων παροχής δεδομένων καθώς και των χρηστών. Χρησιμοποιεί το πρωτόκολλο Open Source Project for a Network Data Access Protocol (OPeNDAP), το οποίο έχει υιοθετηθεί από το Εθνικό γραφείο για την ολοκληρωμένη και διαρκή παρακολούθηση των ωκεανών. Επιτρέπει στους χρήστες να έχουν πρόσβαση σε απομακρυσμένα δεδομένα μέσω του διαδικτύου χρησιμοποιώντας διαδικασίες ανάλυσης και πακέτα οπτικοποίησης όπως είναι το matlab, ferret, IDL. Τα δεδομένα παρέχονται σε διάφορα format συμπεριλαμβανομένων των NetCDF και HDF. *[http://dods.ipsl.jussieu.fr/prism/wp4a/WP4a_prism_knmi.pdf, 2014]*



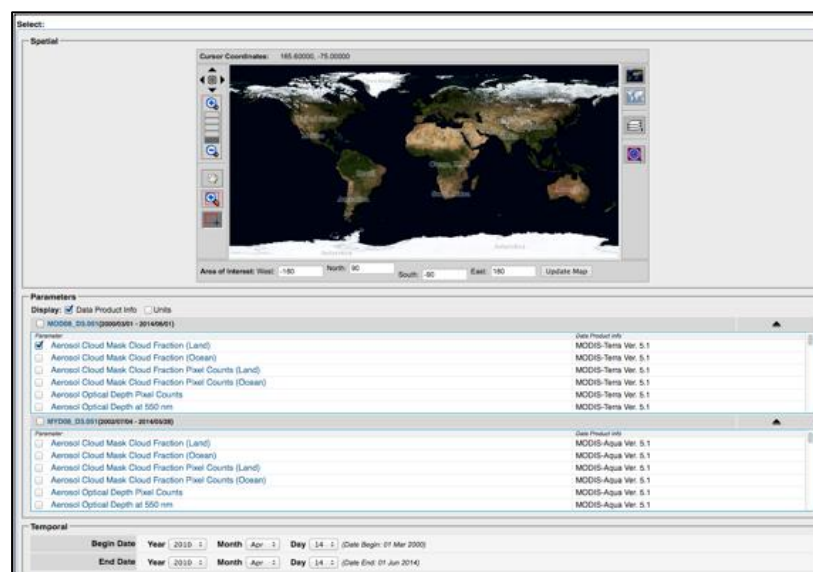
Εικόνα 2.4: Αρχιτεκτονική DODS

http://dods.ipsl.jussieu.fr/prism/wp4a/WP4a_prism_knmi.pdf, 2014

2.4.2 MODIS Online Visualization and Analysis System (MOVAS)

Το πρόγραμμα MOVAS αναπτύχθηκε από την GES DAAC και AreoCenter (GSFC's Laboratory for Atmospheres) και παρέχει στους χρήστες μία εύκολη στη χρήση διαδικτυακή πλατφόρμα για την απεικόνιση και ανάλυση των προϊόντων Modis. Μέχρι στιγμής τα μηνιαία παγκόσμια MODIS Atmosphere προϊόντα είναι διαθέσιμα μέσω της πλατφόρμας αυτής. Επίσης, μέσω του προγράμματος προσφέρονται δεδομένα MODIS –Terra και MODIS – Aqua από το Φεβρουάριο του 2000 και τον Ιούλη του 2002 αντίστοιχα. Ανάλογα με τη χρονολογία των δεδομένων έχουν δημιουργηθεί διαφορετικές συλλογές δεδομένων (Collections) ώστε να είναι εύκολη και γρήγορη η πρόσβαση από τους χρήστες.

[http://gcmd.nasa.gov/records/GES_DAAC_MOVAS.html, 2014]

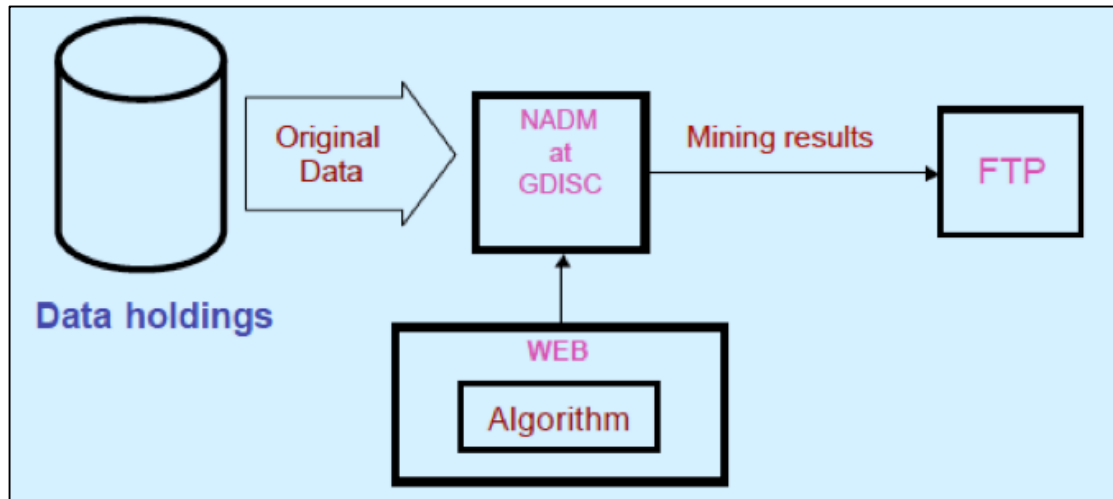


Εικόνα 2.5: MOVAS platform

http://gdata1.sci.gsfc.nasa.gov/daac-bin/G3/gui.cgi?instance_id=MODIS_DAILY_L3, 2014

2.4.3 Near-line Archive Data Mining (NADM)

Το πρόγραμμα Earth Observing System (EOS) της NASA παράγει τεράστιες ποσότητες δεδομένων με αποτέλεσμα κάποιες φορές να καθίσταται δύσκολη η παροχή τους στους χρηστές. Για το λόγο αυτό παρέχεται η δυνατότητα μεταφόρτωσης (upload) των αλγορίθμων των χρηστών στο χώρο αποθήκευσης των πρωτογενών δεδομένων χωρίς να καθίσταται, έτσι, απαραίτητο το κατέβασμα των δεδομένων αυτών. Η παραπάνω διαδικασία πραγματοποιείται μέσω του συστήματος NADM που λειτουργεί σαν μια πύλη εξόρυξης δεδομένων από το πρόγραμμα EOS, μεγέθους τουλάχιστον 50 terabytes. Στο πρόγραμμα υποστηρίζονται οι C, FORTRAN, IDL γλώσσες προγραμματισμού και ο χρήστης έχει τη δυνατότητα της απευθείας ανάπτυξης και εκτέλεσης αλγορίθμων εφόσον πραγματοποιηθούν οι απαραίτητοι έλεγχοι ασφαλείας. [American Geophysical Union, 2002]



Εικόνα 2.6: Αρχιτεκτονική NADM

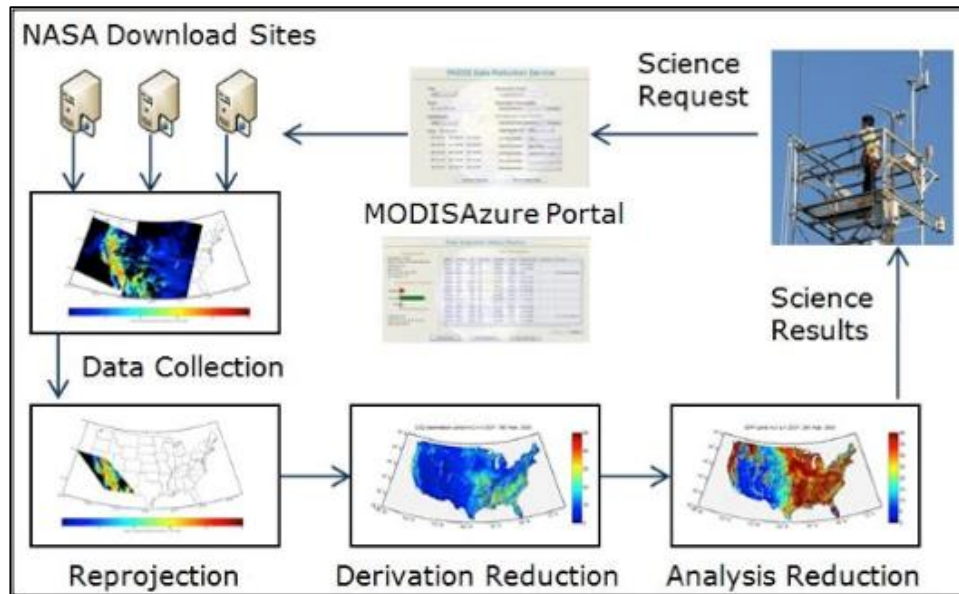
Leptoukh, Savtchenko, Kempner, Gopalan, & Berrick, 2011

2.4.4 MODISAzure

Το πρόγραμμα MODISAzure αποτελεί έναν «αγωγό» για τη λήψη, επεξεργασία και μείωση των διάφορων δορυφορικών δεδομένων χρησιμοποιώντας την τεχνολογία Windows Azure, έτσι ώστε να επιτευχθούν αποτελέσματα μαζικής υπολογιστικής ισχύος, μέσω εφαρμογών σε επίπεδο cloud. Το έργο MODISAzure αποσκοπεί στην επίλυση των προβλημάτων που αφορούν στην πρόσβαση σε μεγάλα τηλεπισκοπικά δεδομένα από τους δορυφόρους MODIS και άλλους. Πιο συγκεκριμένα, μέσω του προγράμματος αυτού αυτοματοποιούνται τα παρακάτω τέσσερα στάδια:

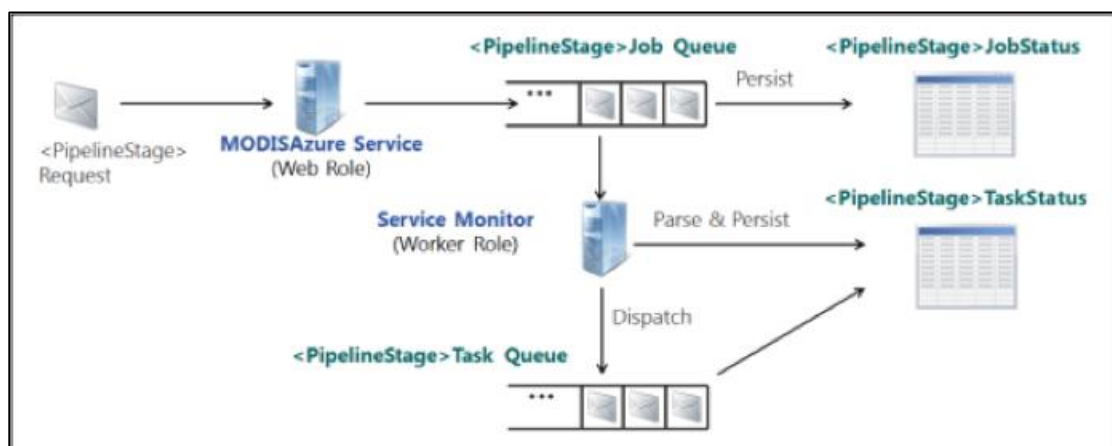
[<http://research.microsoft.com/en-us/projects/modisazure/default.aspx>, 2014]

- Μεταφόρτωση δεδομένων από αποθετήρια όπως το LPDAAC (Land Products Distributed Active Archive Center) και το Goddard Space center
- Επαναπροβολή και ενάρμοση της εικόνας σε κοινές συντεταγμένες και αναλύσεις
- Εκτέλεση κατ' αίτηση (on demand) δύο αλγορίθμων που προέρχονται από το χρήστη και σχετίζονται με την παραγωγή και επεξεργασία διάφορων μεταβλητών καθώς και με την παραγωγή γραφημάτων, πινάκων, στατιστικών και άλλων
- Παράδοση των τελικών αποτελεσμάτων στην επιφάνεια εργασίας του χρήστη



Εικόνα 2.7: Στάδια MODIS Azure

<http://research.microsoft.com/en-us/projects/modisazure/default.aspx>, 2014



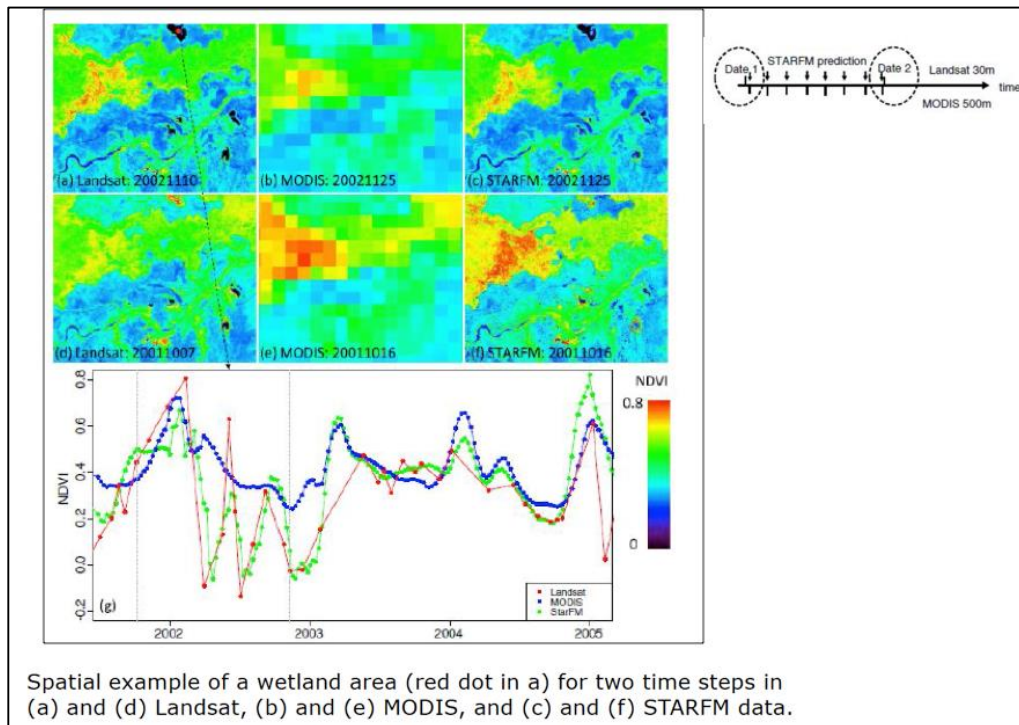
Εικόνα 2.8: Αρχιτεκτονική MODIS Azure

<http://research.microsoft.com/en-us/projects/modisazure/default.aspx>, 2014

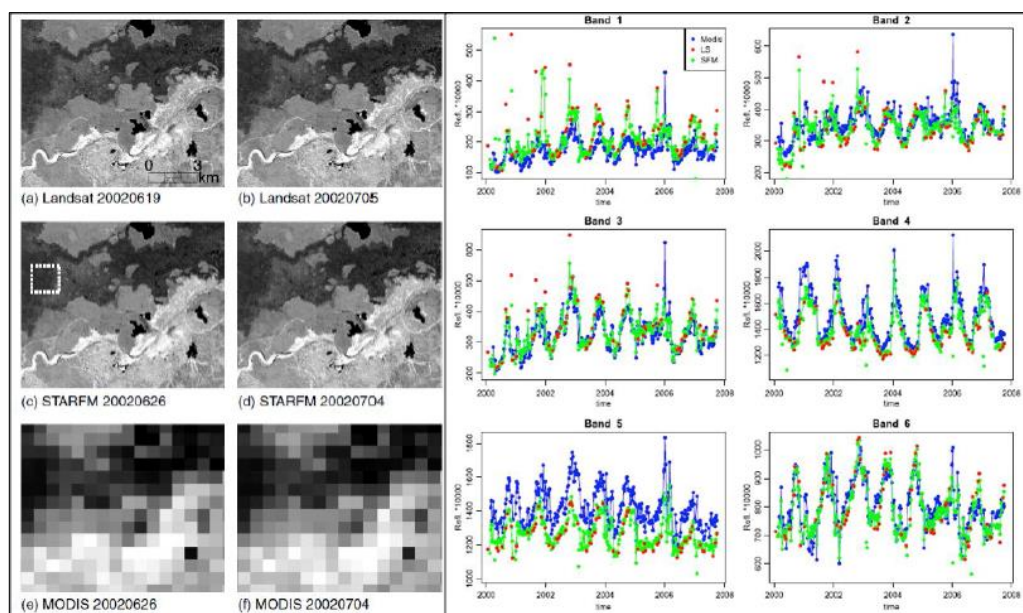
2.4.5 Mass Processing

Μέσω της μαζικής επεξεργασίας γίνεται προσπάθεια να επιλυθεί το «πρόβλημα» της κλίμακας και να πραγματοποιηθεί συνδυασμός δεδομένων που χαρακτηρίζονται από μικρό χρονικό βήμα και μικρή γεωμετρική διακριτική ικανότητα (π.χ. MODIS) με δεδομένα που χαρακτηρίζονται από μεγαλύτερο χρονικό βήμα και υψηλότερη γεωμετρική διακριτική ικανότητα (π.χ. Landsat). Με το συγκεκριμένο τρόπο, μέσω διαδικασιών συγχώνευσης (fusion) δίνεται η δυνατότητα παραγωγής «μαθηματικών» εικόνων που συνδυάζουν τα χαρακτηριστικά των δύο δορυφόρων. Για το σκοπό αυτό, είναι απαραίτητη η συλλογή, επεξεργασία και ανάλυση μακρών χρονοσειρών έτσι ώστε να εξαχθεί η απαραίτητη πληροφορία. Όμοια, παρέχεται η δυνατότητα συγχώνευσης δεδομένων

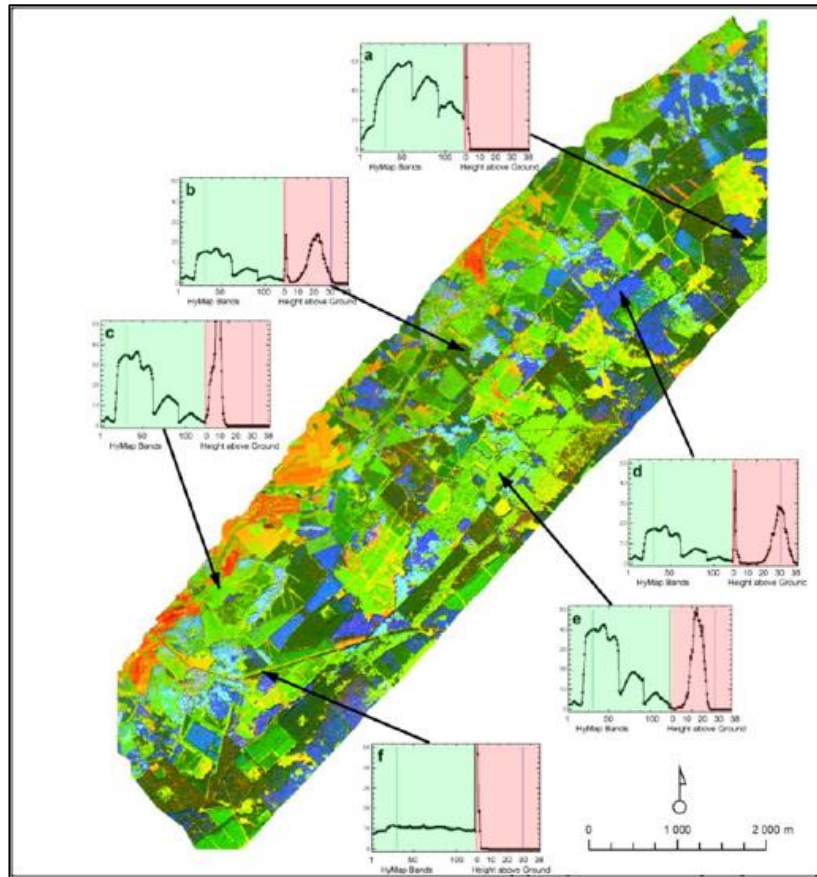
που αναπαριστούν διαφορετική πληροφορία, όπως είναι για παράδειγμα οι υπερφασματικοί δέκτες και τα συστήματα σάρωσης με λέιζερ. [Schmidt et al., 2012] Στις εικόνες 2.9 και 2.10 αναπαριστώνται οι σχηματικές απεικονίσεις της εφαρμογής του αλγορίθμου STARFM [Gao et al., 2010] και στην εικόνα 2.11 αναπαριστάται ένα παράδειγμα μαζικής επεξεργασίας δεδομένων από συστήματα σάρωσης με λέιζερ και υπερφασματικούς δέκτες.



Εικόνα 2.9: Εφαρμογή STARFM algorithm σε NDVI
Udelhoven, 2012



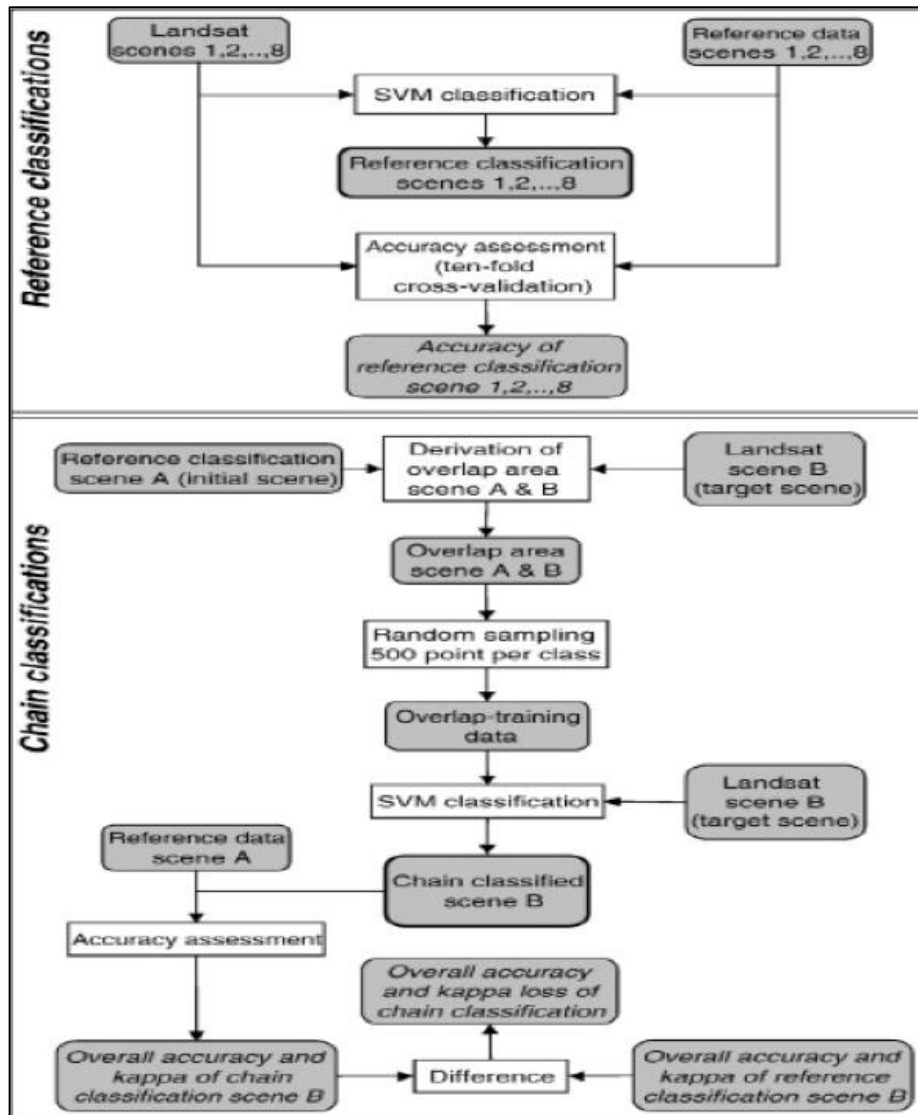
Εικόνα 2.10: Εφαρμογή STARFM algorithm σε time series reflectance
Udelhoven, 2012



Εικόνα 2.11: Συγχώνευση laser-scanning and hyperspectral reflective data
Buddenbaum et al., 2011

2.4.6 Συγχώνευση Πολυμορφικών Δεδομένων

Σημαντικό πρόβλημα που αντιμετωπίζεται πολύ συχνά σε διαδικασίες ανάλυσης με τηλεπισκοπικά δεδομένα είναι η έλλειψη πληροφορίας και γνώσης εδάφους σε συγκεκριμένες περιοχές μελέτης. Κατά συνέπεια, καθίσταται σχεδόν αδύνατη η διαδικασία ταξινόμησης και αξιολόγησης σε τέτοιες περιοχές. Με την αξιοποίηση, ωστόσο, των μεγάλων τηλεπισκοπικών δεδομένων τόσο από διαφορετικούς δέκτες όσο και από τους ίδιους δέκτες παρέχεται η δυνατότητα εργασίας σε γειτονικές περιοχές, που υπάρχει γνώση εδάφους, και μετέπειτα προσπάθεια για εξαγωγή και δημιουργία δεδομένων για την περιοχή που δεν υπήρχε πληροφορία. Στη συνέχεια, παρουσιάζεται η σχηματική απεικόνιση του αλγορίθμου της «αλυσιδωτής» ταξινόμησης (chain classification) σύμφωνα με τους *J. Knorn et al. (2009)*.



Εικόνα 2.12: Chain classifications
J. Knorn et al., 2009

2.5 Τεχνολογικά Προβλήματα – Τεχνικές Δυσκολίες

Η τεχνολογία των μεγάλων δεδομένων βρίσκει εφαρμογές σε πολλά επιστημονικά αντικείμενα καθώς και σε εφαρμογές που σχετίζονται με την καθημερινή ζωή. Ωστόσο, γίνεται αντιληπτό πως η διαχείριση και η επεξεργασία τόσο μεγάλου όγκου δεδομένων δημιουργεί σημαντικά προβλήματα και δυσκολίες στις εταιρείες που προσπαθούν να τα αξιοποιήσουν. Στο συγκεκριμένο κεφάλαιο παρουσιάζονται συνοπτικά τέτοιου είδους προβλήματα.

Αρχικά, απαιτείται αποτελεσματικό σύστημα διανομής αρχείων για τη συλλογή, αποθήκευση και ανάλυση της ροής του δικτύου έτσι ώστε να αντιμετωπιστούν οι πιθανές δυσκολίες. Επιπλέον, λόγω της διαρκούς συλλογής και της πολυπλοκότητας των δεδομένων απαιτείται αποτελεσματικό δίκτυο διαχείρισης, κατά τέτοιο τρόπο ώστε το πρόβλημα της ανάλυσης των μεγάλων δεδομένων να μπορεί να

αντιμετωπισθεί, λαμβάνοντας υπόψη τις παραμέτρους της αποτελεσματικότητας και τους κόστους. [Carlin & Curran, 2012] Σημαντικές δυσκολίες παρατηρούνται, επίσης, και στον τομέα της έρευνας στη δικτύωση υπολογιστών καθώς το κόστος επικοινωνίας αποτελεί μεγαλύτερη δυσκολία σε σύγκριση με την επεξεργασία των δεδομένων. Παράλληλα, το εύρος σύνδεσης καθώς και ο χρόνος αναμονής στα δίκτυα δημιουργεί προβλήματα στην άμεση επικοινωνία μεταξύ του πελάτη (client) και του διακομιστή (server). [Wong, Shen, Johnson, Chen, & Ross, 2012]

Σημαντικά προβλήματα παρουσιάζονται, επίσης, στην αποθήκευση των δεδομένων. Αν ληφθεί υπόψη πως η μέγιστη χωρητικότητα ανά δίσκο είναι περίπου τα 4 terabytes γίνεται αντιληπτό πως η αποθήκευση 1 exabyte δεδομένων θα απαιτούσε περίπου 25000 δίσκους. Από το παραπάνω γίνεται αντιληπτό πως χρειάζεται περισσότερος χρόνος για τη μετάδοση των δεδομένων από ένα σημείο συγκέντρωσης ή αποθήκευσης σε ένα σημείο επεξεργασίας από αυτόν που απαιτείται για την επεξεργασία των δεδομένων. Η διαχείριση, επίσης, των μεγάλων δεδομένων δημιουργεί ζητήματα πρόσβασης, αξιοποίησης και ενημέρωσης των υπαρχόντων δεδομένων καθώς απαιτείται πλήρη καταγραφή των μεταδεδομένων, έτσι ώστε να υπάρχουν διαφοροποιήσεις μεταξύ των ψηφιακών δεδομένων όπως είναι οι εικόνες, τα έγγραφα, τα βίντεο και άλλα. Παράλληλα, δημιουργούνται προβληματισμοί σχετικά με το ποια δεδομένα, τελικά, χρειάζονται να αποθηκευθούν, ποια χρειάζονται να χρησιμοποιηθούν σε διαδικασίες ανάλυσης καθώς και ποια δεδομένα αν χρησιμοποιηθούν θα οδηγήσουν στα μέγιστα δυνατά οφέλη. [Kaisler, Armour, Espinosa, & Money, 2013]

Αξίζει να σημειωθεί πως τέτοιου είδους προβλήματα αντιμετωπίζουν και όλοι οι οργανισμοί που συλλέγουν μεγάλα τηλεπισκοπικά δεδομένα (π.χ. USGS) καθώς η παράλληλη συλλογή, επεξεργασία, διανομή και ανάλυση δεδομένων από δορυφορικούς τηλεπισκοπικούς δέκτες, παγκόσμια δορυφορικά συστήματα πλοήγησης (GNSS), εναέριες λήψεις, ψηφιακές κάμερες, επίγειους δέκτες καθώς και από ραντάρ και lidar καθιστούν πολύ δύσκολο το χειρισμό τους.

2.6 Πλατφόρμες Διαχείρισης και Επεξεργασίας

Για να υπάρξει αποτελεσματική διαχείριση και επεξεργασία των μεγάλων δεδομένων έχουν αναπτυχθεί αρκετά λογισμικά, τα οποία εστιάζουν κυρίως στο χειρισμό και στην αποθήκευση αυτών προσφέροντας, παράλληλα, δυνατότητες εφαρμογής διαδικασιών ανάλυσης ανάλογα με τον τύπο των δεδομένων που υποστηρίζουν. Συνοπτικά, τα λογισμικά που επιλέγονται προς χρήση, προκειμένου να αντιμετωπίζουν αποτελεσματικά τις δυσκολίες των μεγάλων δεδομένων που προαναφέρθηκαν, πρέπει: [Singh, 2012]

- να αντιμετωπίζουν και τις τρεις διαστάσεις των μεγάλων δεδομένων
- να είναι αποδοτικά και να παρέχουν ασφάλεια, χρηστικότητα και αξιοπιστία

- να επιτρέπουν την ενσωμάτωση σύνθετων βάσεων δεδομένων
- να χαρακτηρίζονται από τις αρχές του ανοιχτού λογισμικού
- να χαρακτηρίζονται από υψηλές ταχύτητες επεξεργασίας μέσω της αποτελεσματικής αξιοποίησης του υλικού εξοπλισμού (hardware)
- να απαιτούν όσο το δυνατό λιγότερη συντήρηση
- να παρέχουν δυνατότητες αντιμετώπισης στοχευμένων (adhoc) ερωτημάτων

Στο κεφάλαιο 2.6.1 που ακολουθεί παρουσιάζεται συνοπτικά το λογισμικό ανοιχτού κώδικα Hadoop που αποτελεί μία από τις πιο αξιόπιστες λύσεις για τη διαχείριση των μεγάλων δεδομένων στον τομέα των επιχειρήσεων και οργανισμών. Τέλος, στο κεφάλαιο 2.6.2 παρουσιάζεται ο διεκπεραιωτής (server) rasdaman που χρησιμοποιείται για την διαχείριση μεγάλων τηλεπισκοπικών δεδομένων.

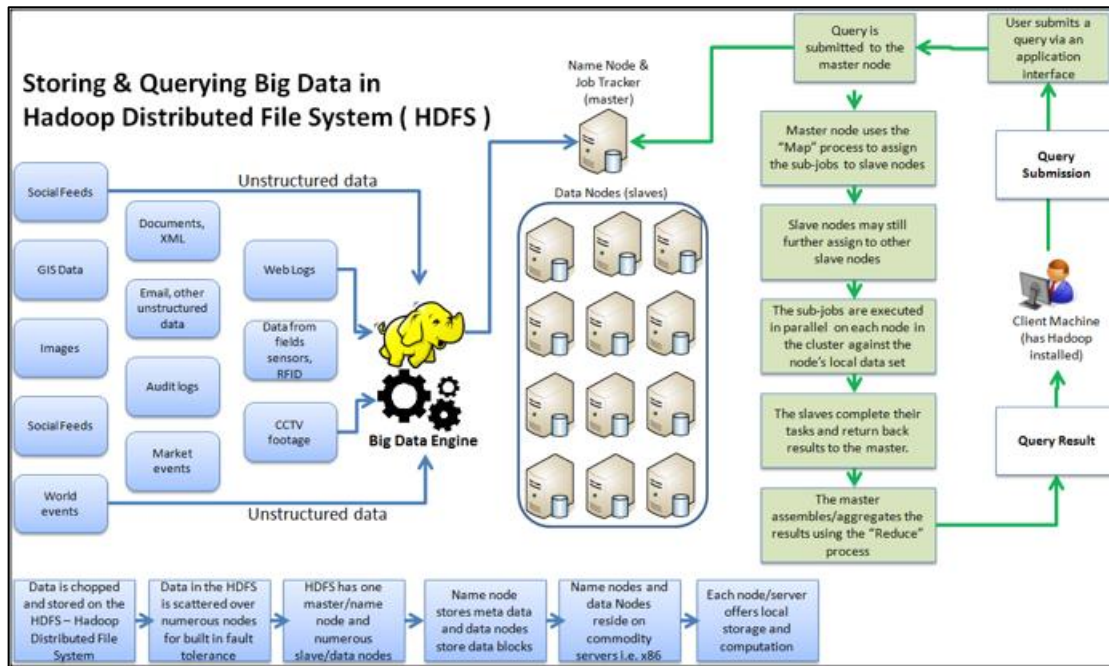
2.6.1 Hadoop (Apache Software Foundation)

Το λογισμικό ανοιχτού κώδικα Hadoop έχει αναπτυχθεί από την εταιρεία Apache Software Foundation και πρόκειται για ένα αξιόπιστο και κλιμακούμενο πρόγραμμα για κατανεμημένα υπολογιστικά συστήματα. Επιτρέπει την κατανεμημένη επεξεργασία μεγάλων δεδομένων με χρήση απλών μοντελών προγραμματισμού. Είναι σχεδιασμένο με τέτοιο τρόπο ώστε να επιτρέπει την αναβάθμιση από ένα μοναδικό διεκπεραιωτή (server) σε χιλιάδες υπολογιστικά συστήματα επιτρέποντας την τοπική αποθήκευση και επεξεργασία των δεδομένων. Επιπλέον, δεν βασίζεται αποκλειστικά στον υλικό εξοπλισμό (hardware) αλλά έχει σχεδιαστεί με τέτοιο τρόπο ώστε να διαχειρίζεται προβλήματα και σε επίπεδο εφαρμογών. (<http://hadoop.apache.org>, 2014)

Το λογισμικό αποτελείται από διάφορα εξωτερικά πρόσθετα, εκ των οποίων τα τέσσερα πιο σημαντικά είναι τα εξής [<http://hadoop.apache.org>, 2014]:

- **Hadoop Common:** Περιλαμβάνει τις βασικές και αναγκαίες λειτουργίες για να λειτουργούν αποτελεσματικά όλες οι ενότητες
- **Hadoop Distributed File System (HDFS™):** Πρόκειται για ένα κατανεμημένο σύστημα αρχείων που παρέχει υψηλής απόδοσης επεξεργασίες στα δεδομένα της κάθε εφαρμογής
- **Hadoop YARN:** Λογισμικό για τον προγραμματισμό των εσωτερικών λειτουργιών της εταιρείας
- **Hadoop MapReduce:** Στήριζεται στο πλαίσιο του YARN και ειδικεύεται στην παράλληλη επεξεργασία μεγάλων δεδομένων

Ακολουθεί η σχηματική απεικόνιση της αρχιτεκτονικής του λογισμικού Hadoop:



Εικόνα 2.13: Παράδειγμα Αρχιτεκτονικής Hadoop
[Prakash, 2012]

2.6.2 Rasdaman, Raster Data Manager

Η τεχνολογία του rasdaman επιτρέπει την αποθήκευση πολυδιάστατων πινάκων (δεδομένα σε μορφή εικόνας - raster data) απεριόριστου μεγέθους σε μια παραδοσιακή βάση δεδομένων που στηρίζεται στη γλώσσα προγραμματισμού ερωτήσεων SQL. Επιτρέπει την αποθήκευση και επεξεργασία μεγάλων γεωχωρικών δεδομένων που χρησιμοποιούνται σε διάφορους τομείς όπως είναι οι παρακάτω: (<http://www.rasdaman.com>, 2014)

- **Γεω-εφαρμογές:** διαδικτυακές απεικονίσεις, περιβαλλοντικές εφαρμογές, ωκεανογραφία, ανάλυση φυσικών καταστροφών, εφαρμογές σε καλλιέργειες και άλλα [Karmas et al., 2014]
- **Ιατρική:** απεικονίσεις εγκεφάλου, διάφοροι ιατρικοί σαρωτές όπως είναι οι ακτίνες X, PET, fMRI και άλλα
- **Αστρονομία:** οπτικές και ραδιοαστρονομικές εφαρμογές, κοσμολογικές προσομοιώσεις
- **Στατιστικά:** εφαρμογές σε n-διαστάσεις δεδομένων (είδος συστήματος MOLAP- Multidimensional Online Analytical Processing)

Το rasdaman είναι διαθέσιμο τόσο σε λογισμικό ανοιχτού κώδικα όσο και σε εμπορικό υποστηρίζοντας διάφορες παραλλαγές. Μέσα από τη διαρκή επεκτασιμότητά του, το σύστημα αντιμετωπίζει με άρτιο τρόπο τον αυξανόμενο όγκο δεδομένων, τον μεγάλο αριθμό χρηστών και την πολυπλοκότητα των ερωτημάτων (queries). Επιπλέον, μπορεί να χρησιμοποιηθεί παράλληλα με διαφορετικά συστήματα βάσεων δεδομένων και προσφέρει υψηλά επίπεδα βελτιστοποίησης υλικού (hardware) και λογισμικού. [<http://www.rasdaman.com>, 2014]

Στα πλαίσια του προγράμματος *rasdaman* έχουν αναπτυχθεί πολλές εφαρμογές που σχετίζονται με θέματα γεωπληροφορικής και τηλεπισκόπησης, τα οποία παρουσιάζονται συνοπτικά στη συνέχεια.

AIP-4: Σε συνεργασία με την EOX GmbH το πρόγραμμα *rasdaman* δημιούργησε μία δοκιμαστική έκδοση για στοχευμένη (adhoc) ανάλυση σε τηλεπισκοπικά δεδομένα στα πλαίσια του προγράμματος GEOSS. Στον τομέα της γεωπληροφορικής έχουν αναπτυχθεί διαδικασίες ανάλυσης για διαδικτυακή χαρτογράφηση, τηλεπισκόπηση, κλιματολογία, ωκεανογραφία και άλλα. [<http://www.earthlook.org/index.html>, 2014]

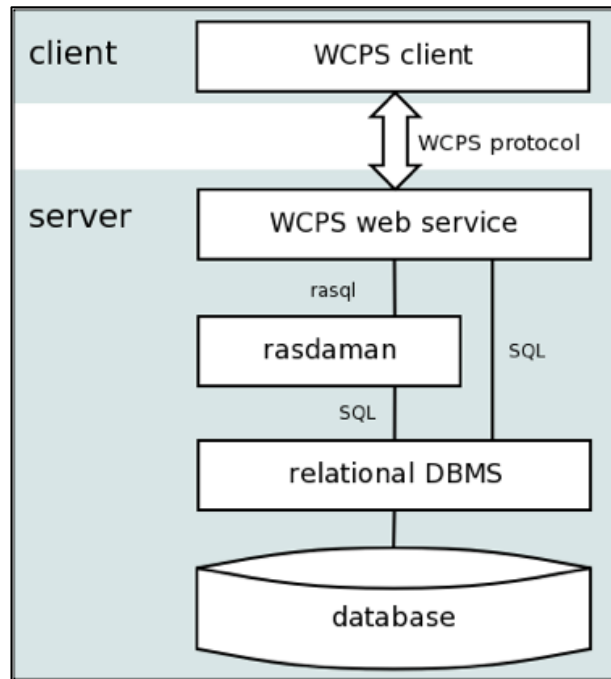
EarthServer: Το συγκεκριμένο έργο στοχεύει στην ελεύθερη πρόσβαση και ανάλυση τηλεπισκοπικών δεδομένων που στηρίζονται στα πρότυπα του OGC, στις υπηρεσίες του Web Coverage Service (WCS) καθώς και στο πρότυπο του Web Processing Service (WCPS). Σκοπός είναι μέσω της χρήσης γλωσσών προγραμματισμού, που εστιάζονται στις βάσεις δεδομένων, να παρέχεται πρόσβαση σε πολλαπλές πηγές χωροχρονικών δεδομένων με βάση τα ανοιχτά πρότυπα του OGC. Το πρόγραμμα στοχεύει από την πλευρά του διακομιστή να δημιουργηθεί μια παράλληλη και κατανομημένη επεξεργασία ερωτημάτων που θα εξασφαλίσει τη διαρκή επεκτασιμότητα του όγκου των δεδομένων. [www.earthserver.eu/, 2014]

OWS-9: Το πρόγραμμα OWS-9 (OGC Web Services, Phase 9) σε συνεργασία με το *rasdaman* δημιούργησαν κάλυψη των υπηρεσιών που σχετίζονται με την χρήση προηγμένης τεχνολογίας και την επέκταση των προδιαγραφών του προγράμματος WCS. Επιπλέον, δημιουργήθηκε συνεργασία με τις Ηνωμένες Πολιτείες Αμερικής, την NASA, NGA, OPeNDAP και άλλα ιδρύματα σε παγκόσμιο επίπεδο. [<http://www.rasdaman.com>, 2014]

TeraPro-Tera Scale Image Processing: Πρόκειται για ένα εξειδικευμένο σύστημα επεξεργασίας εικόνων (proStack) που συνδέεται άμεσα με μια βάση δεδομένων, μορφής στήλης, για τον βέλτιστο συνδυασμό διαδικασιών ανάλυσης και επιτρέπει τη δημιουργία κατάλληλων δυναμικών (on the fly) ερωτημάτων. [<http://www.rasdaman.com>, 2014]

RemoteAgri: Πρόκειται για μια διαδικτυακή εφαρμογή που αναπτύχθηκε στη σχολή των Αγρονόμων και Τοπογράφων Μηχανικών, στο εργαστήριο Τηλεπισκόπησης, και έχει ως στόχο τη διαχείριση, επεξεργασία, ανάλυση μεγάλων γεωχωρικών δεδομένων για εφαρμογές γεωργίας χρησιμοποιώντας δεδομένα από το δορυφόρο Landsat 8. [*Karmas et al.*, 2014]

Ακολουθεί παράδειγμα της αρχιτεκτονικής του *rasdaman*:



Εικόνα 2.14: Παράδειγμα Αρχιτεκτονικής Rasdaman

<http://www.rasdaman.org/attachment/wiki/Technology/wcps-stack.png>

2.7 Υπολογιστικά Συστήματα και Απαιτήσεις

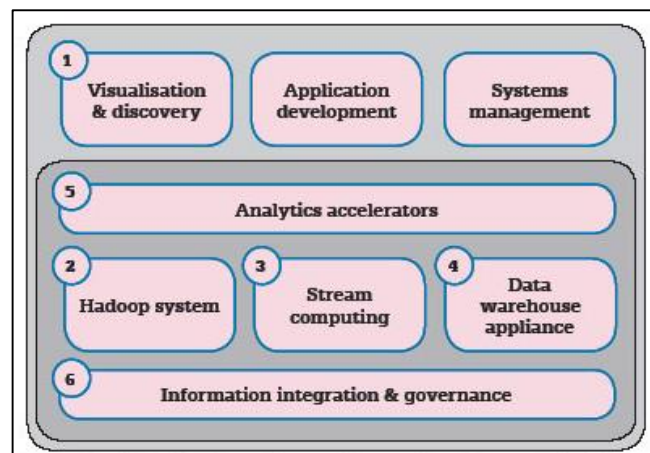
Στο συγκεκριμένο κεφάλαιο παρουσιάζονται οι προοπτικές ανάπτυξης καθώς και οι τελευταίες τεχνολογικές εξελίξεις στον τομέα της διαχείρισης και επεξεργασίας των μεγάλων δεδομένων. Στο κεφάλαιο 2.7.1 παρουσιάζονται οι σύγχρονες τεχνολογίες στον τομέα του υπολογιστικού υλικού και των εφαρμογών και στο κεφάλαιο 2.7.2 παρουσιάζονται οι προοπτικές ανάπτυξης καθώς και οι απαιτήσεις για εφάρμογες των μεγάλων δεδομένων στον τομέα της γεωπληροφορικής και της τηλεπισκόπησης.

2.7.1 Συστήματα

Από το κεφάλαιο 2.5 γίνεται αντιληπτό πως η αποθήκευση και ακόμα περισσότερο η επεξεργασία των μεγάλων δεδομένων απαιτεί τη χρήση υπερσύγχρονων υπολογιστών προκειμένου να μην δημιουργούνται προβλήματα και δυσκολίες στους οργανισμούς και στους χρηστές. Λόγω της έλλειψης υπολογιστικής δύναμης σε επίπεδο ατομικών υπολογιστικών μονάδων οι οργανισμοί-εταιρείες δημιουργούν συστήματα στηριζόμενα στην τεχνολογία cloud έτσι ώστε να δημιουργείται η κατάλληλη ισχύς που μπορεί να επεξεργαστεί τέτοιου όγκου δεδομένα. Σημαντικό παράδειγμα χρήσης της τεχνολογίας cloud και των μεγάλων γεωχωρικών δεδομένων είναι το σύστημα ανάλυσης Cloudbased system, το οποίο ενσωματώνει την τεχνολογία NetApp αντιμετωπίζοντας άρτια διαδικασίες ανάλυσης που σχετίζονται με την εξόρυξη δεδομένων, σύνδεση δικτύων, γεωχωρικών δυνατοτήτων και άλλα. [Dasgupta, 2013]

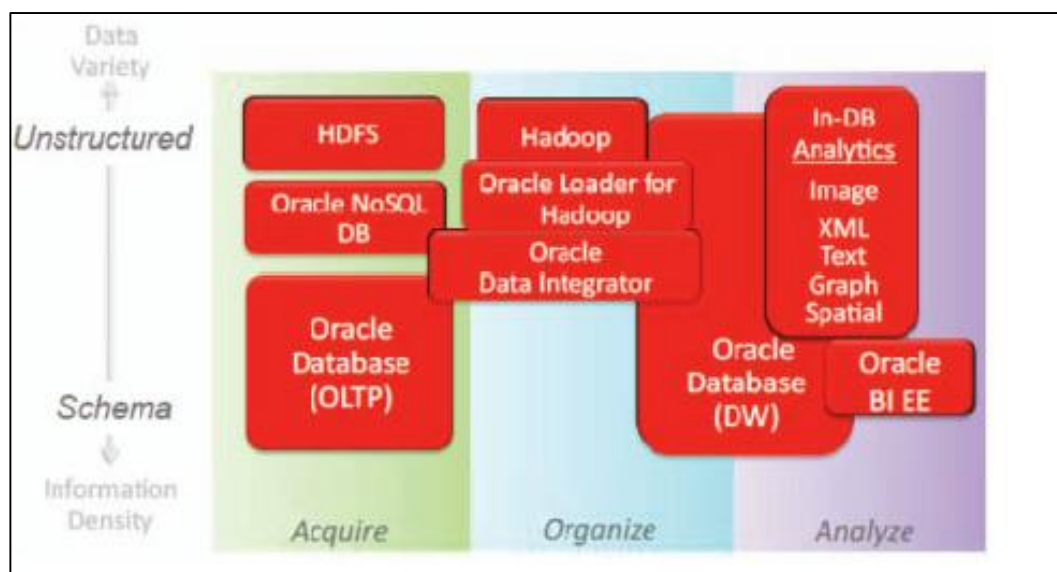
Σε συνάρτηση με τη χρήση της τεχνολογίας cloud, που εκμεταλλεύεται την παράλληλη επεξεργασία, οι πιο σύγχρονες διαδικασίες ανάλυσης χρησιμοποιούν τεχνικές που εκμεταλλεύονται τις μονάδες επεξεργασίας γραφικών (GPU) και όχι αποκλειστικά την κεντρική μονάδα επεξεργασίας (CPU). Προτείνεται, επίσης, η παράλληλη ενσωμάτωση βάσεων δεδομένων γνωστών και εως NoSQL (όχι μόνο SQL), όπου έχει αποδειχτεί πως συμβάλλουν στη βελτιστοποίηση του χρόνου επεξεργασίας δεδομένων κατά 10 έως 100 φορές. Σημαντικό πλεονέκτημα, ακόμα, της χρήσης NoSQL είναι η ενσωμάτωση ερωτημάτων σε πραγματικό χρόνο καθώς η χρήση αποκλειστικά της SQL καθιστά σχεδόν αδύνατη τη συγκεκριμένη διαδικασία. [Dasgupta, 2013]

Ακολουθούν διαγραμματικά παραδείγματα που περιγράφουν παραστατικά την ενσωμάτωση των προαναφερθέντων τεχνολογιών στην αρχιτεκτονική συστημάτων:



Εικόνα 2.15: IBM Big Data Platform

Paul C. Zikopoulos & Al, 2013



Εικόνα 2.16: Oracle Integrate Software Solution Stack

[Dasgupta, 2013]

2.7.2 Απαιτήσεις

Η επιστήμη της γεωπληροφορικής και πιο συγκεκριμένα οι επιστήμες που ασχολούνται με την παρατήρηση της επιφάνειας της Γης, όπως είναι για παράδειγμα η τηλεπισκόπηση, είναι υποχρεωμένες να χρησιμοποιούν και να επεξεργάζονται μεγάλα γεωχωρικά δεδομένα. Όσον αφορά τις διαδικασίες που σχετίζονται με τα μεγάλα δεδομένα στο αντικείμενο της παρατήρησης της Γης, πρέπει να καλύπτονται οι παρακάτω συνθήκες, έτσι ώστε να υπάρξει αρτιότερη εκμετάλλευση του τεράστιου όγκου πληροφορίας. [ESA, 2013]

- Γρήγορη και αξιόπιστη προσπέλαση δεδομένων
- Συνεχείς αναλύσεις για την εύρεση χαρακτηριστικών (patterns) και τάσεων (trends)
- Δημιουργία κέντρων διάθεσης δεδομένων σε παγκόσμιο επίπεδο
- Εύρεση συσχετίσεων και σχέσεων μεταξύ των διάφορων δεδομένων
- Εφαρμογή διαδικασιών εξόρυξης (data mining) για τον εμπλουτισμό της πληροφορίας
- Μαζική επεξεργασία δεδομένων (mass processing)
- Παράλληλη χρήση δεδομένων από δορυφόρους με διαφορετική γεωμετρική, ραδιομετρική και φασματική διακριτική ικανότητα
- Τεχνική βελτιστοποίηση στους τομείς της αυτόματης απόκτησης, ομαδοποίησης, χειρισμού, ασφάλειας και μεταφοράς δεδομένων και αλγορίθμων
- Χρήση ιδιωτικών ή δημόσιων πλατφόρμων cloud που προτείνονται ως πιστοποιημένα προϊόντα από την ΕΟ για την διευκόλυνση των χρηστών ώστε να πραγματοποιούνται δοκιμές των αλγορίθμων σε πραγματικό χρόνο

Για να υπάρξει αποτελεσματική χρήση των μεγάλων γεωχωρικών δεδομένων είναι απαραίτητη η συμμετοχή αναλυτών από διαφορετικούς τομείς. Πιο συγκεκριμένα, είναι απαραίτητη η συνεργασία μαθηματικών, προγραμματιστών υπολογιστών, γεωγράφων, μηχανικών, κοινωνικών επιστημόνων, στελεχών εταιρειών καθώς και απλών χρηστών. Είναι απαραίτητο, επίσης, να υπάρχει διεπιστημονική επαφή έτσι ώστε να αναλυθούν σωστά τα δεδομένα ως προς τις χωροταξικές αλληλεπιδράσεις και τις χωρικές συμπεριφορές με τον ταυτόχρονο συνδυασμό των παραδοσιακών τεχνικών στα γεωχωρικά δεδομένα και των στατιστικών μοντέλων που προκύπτουν από την ανάλυση τους. [Dasgupta, 2013]

Κεφάλαιο 3: Πρόγραμμα Δορυφόρων Landsat (USGS)

3.1 Γενικά Στοιχεία

Οι δορυφόροι Landsat αποτελούν παγκοσμίως τη μεγαλύτερη συλλογή διαστημικών τηλεπισκοπικών δεδομένων, μεσαίας ανάλυσης, για εφαρμογές παρατήρησης της επιφάνειας της Γης. Παρέχουν δεδομένα για τέσσερις συνεχόμενες δεκαετίες σε τομείς που ασχολούνται με τη γεωργία, τη γεωλογία, τη χωροταξία, την εκπαίδευση, τη χαρτογράφηση και την παγκόσμια έρευνα για τη διαρκή παρακολούθηση των αλλαγών στην επιφάνεια της Γης. Τα δεδομένα από τους δορυφόρους Landsat έχουν χρησιμοποιηθεί, επίσης, για την αντιμετώπιση καταστάσεων έκτακτης ανάγκης καθώς και για την ανίχνευση φυσικών καταστροφών.

[http://landsat.usgs.gov/about_project_descriptions.php, 2014]

Το πρόγραμμα δορυφόρων Landsat ξεκίνησε τη λειτουργία του με τον πρώτο δορυφόρο με ονομασία Landsat 1 στις 23 Ιουλίου του 1972 τελειώνοντας τη λειτουργία του στις 6 Ιανουαρίου του 1978, παρέχοντας δεδομένα για διάστημα περίπου 6 ετών. [http://landsat.usgs.gov/about_landsat1.php, 2014] Στις 22 Ιανουαρίου του 1975 εκτοξεύθηκε η δεύτερη σειρά δορυφόρου με ονομασία Landsat 2 και με χαρακτηριστικά περίπου όμοια με τα αντίστοιχα του πρώτου δορυφόρου. Ο Landsat 2 παρείχε δεδομένα για διάστημα περίπου 8 χρόνων τερματίζοντας τη λειτουργία του στις 27 Ιουλίου του 1983. [http://landsat.usgs.gov/about_landsat2.php, 2014] Παράλληλα με τον Landsat 2, ξεκίνησε τη λειτουργία του σε κοντινό χρονικό διάστημα και ο Landsat 3 στις 5 Μαρτίου του 1978 λειτουργώντας για 5 χρόνια και τερματίζοντας τη λειτουργία του στις 7 Σεπτεμβρίου του 1983. [http://landsat.usgs.gov/about_landsat3.php, 2014] Τους προηγούμενους δύο δορυφόρους τους αντικατέστησε ο Landsat 4, ο οποίος παρείχε δεδομένα για διάστημα 11 ετών, καθιστώντας τον έναν από τους πιο επιτυχημένους δορυφόρους της δεκαετίας. Πιο συγκεκριμένα, η λειτουργία του Landsat 4 διήρκησε από τις 26 Ιουλίου του 1982 έως τις 14 Δεκεμβρίου του 1993. [http://landsat.usgs.gov/about_landsat4.php, 2014]

Τη δεκαετία του '80 ωστόσο η εταιρεία USGS στα πλαίσια του προγράμματος Landsat εκτόξευσε την 1 Μαρτίου του 1984 το μακροβιότερο δορυφόρο στην ιστορία των δορυφόρων παρατήρησης της Γης με ονομασία Landsat 5 και με διάρκεια ζωής τα 29 χρόνια, τερματίζοντας τη λειτουργία του τον Ιανουάριο του 2013. [http://landsat.usgs.gov/about_landsat5.php, 2014] Τη χρονική περίοδο που τερμάτισε ο Landsat 4 τη λειτουργία του έγινε προσπάθεια για εκτόξευση του δορυφόρου Landsat 6, ο οποίος θα λειτουργούσε παράλληλα με τον Landsat 5 αλλά δεν κατάφερε να πιάσει τροχιά και πρακτικά δεν παρείχε ποτέ δεδομένα. [http://landsat.usgs.gov/about_landsat6.php, 2014] Η σειρά ολοκληρώνεται με τους δύο, τελευταίας τεχνολογίας, δορυφόρους Landsat 7 και Landsat 8, όπου ο πρώτος έχει ξεκινήσει τη λειτουργία του στις 15 Απριλίου του 1999 και ο δεύτερος στις 11 Φεβρουαρίου του 2013. Αξίζει να σημειωθεί πως ο δορυφόρος Landsat 7 παρέχει

δεδομένα μέχρι και σήμερα λειτουργώντας μέχρι στιγμής περίπου 15 χρόνια.
[http://landsat.usgs.gov/about_landsat7.php, 2014]

Στα πλαίσια της συγκεκριμένης εργασίας, όσον αφορά τα δορυφορικά τηλεπισκοπικά δεδομένα χρησιμοποιήθηκαν δορυφορικές εικόνες από το δορυφόρο Landsat 8, ο οποίος αποτελεί το πιο σύγχρονο δέκτη του προγράμματος Landsat προσφέροντας δεδομένα για πληθώρα εφαρμογών. Στο κεφάλαιο που ακολουθεί περιγράφονται αναλυτικά τα χαρακτηριστικά του δορυφόρου Landsat 8.

3.2 Δορυφορικό Πρόγραμμα Landsat 8

Στο συγκεκριμένο κεφάλαιο παρουσιάζεται ο δορυφόρος Landsat 8 και δίνονται πληροφορίες σχετικά με το οπτικό φάσμα που καλύπτει, τα τεχνικά του χαρακτηριστικά καθώς και για τη γεωμετρική και χρονική διακριτική του ικανότητα.

3.2.1 Βασικά στοιχεία

Ο δορυφόρος Landsat 8 είναι ο πιο σύγχρονος δορυφόρος της USGS που ξεκίνησε τη λειτουργία του τον Φεβρουάριο του 2013. Αποτελείται από δύο διαφορετικούς δέκτες, τον Operational Land Imager (OLI) και τον Thermal Infrared Sensor (TIRS) που αποτελούνται από 9 και 2 κανάλια αντίστοιχα. Τα δεδομένα που χρησιμοποιούνται χωρίζονται σε δύο διαφορετικές κατηγορίες. Τα πρώτα είναι τα Pre-WRS-2 Data Products που αφορούν τη χρονική διάρκεια από τις 11 Φεβρουαρίου 2013 μέχρι τις 10 Απριλίου 2013. Από τις 10 Απριλίου 2013 και μετά χρησιμοποιούνται τα νέα δεδομένα, τα οποία σε σχέση με τα παλιά έχουν βελτιωμένη γεωμετρία και ραδιομετρία. Σημαντικό στοιχείο είναι πως από τα παλιά δεδομένα μπορούν να χρησιμοποιηθούν μόνο τα αντίστοιχα του δέκτη TIRS καθώς υπήρξαν σημαντικές αλλαγές στο δέκτη OLI από τις 10 Απριλίου 2013 και μετά.

Τα προϊόντα του δορυφόρου Landsat 8 λαμβάνονται ελεύθερα από τους ιστοχώρους GloVis και Earthexplorer καθώς και από διαδικτυακούς χώρους που χρησιμοποιούν την τεχνολογία του cloud (π.χ. google, amazon)

[Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013]

3.2.2 Φάσμα που αναλύει

Ακολουθεί πίνακας που περιγράφει τα κανάλια του Landsat 8 για κάθε δέκτη και δίνονται πληροφορίες σχετικά με το οπτικό φάσμα που καλύπτει καθένα από αυτά. Στη συνέχεια, παρουσιάζονται διάφοροι συνδυασμοί καναλιών ανάλογα με τις εφαρμογές που καλύπτουν.

Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) Launched February 11, 2013	Bands	Wavelength (micrometers)
	Band 1 - Coastal aerosol	0.43 - 0.45
	Band 2 - Blue	0.45 - 0.51
	Band 3 - Green	0.53 - 0.59
	Band 4 - Red	0.64 - 0.67
	Band 5 - Near Infrared (NIR)	0.85 - 0.88
	Band 6 - SWIR 1	1.57 - 1.65
	Band 7 - SWIR 2	2.11 - 2.29
	Band 8 - Panchromatic	0.50 - 0.68
	Band 9 - Cirrus	1.36 - 1.38
	Band 10 - Thermal Infrared (TIRS) 1	10.60 - 11.19
	Band 11 - Thermal Infrared (TIRS) 2	11.50 - 12.51

Πίνακας 3.1: Δέκτες και Κανάλια Landsat 8

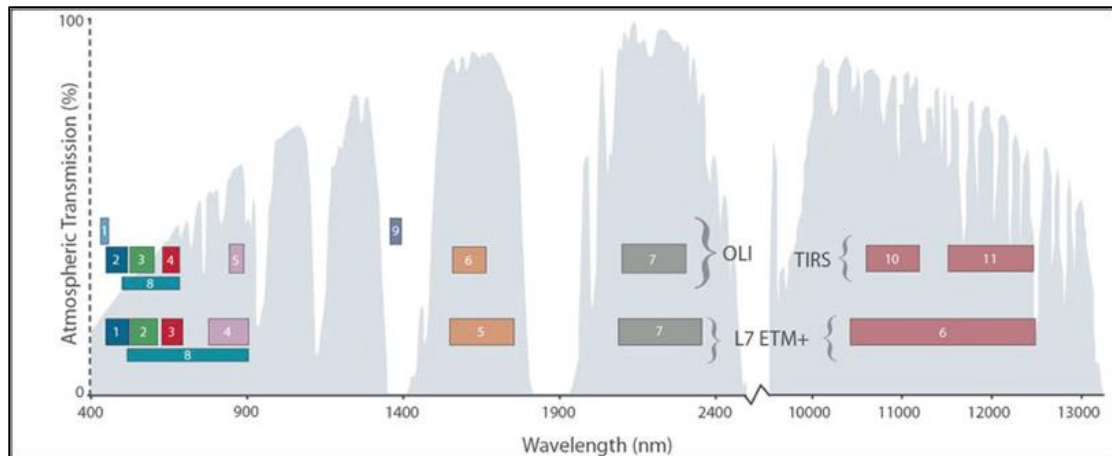
Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013

Color Composite	RGB	Applications (in general..)
	4 3 2	Natural Color
	7 6 4	False Color (urban)
	5 4 3	Color Infrared (vegetation)
	6 5 2	Agriculture
	7 6 5	Atmospheric Penetration
	5 6 2	Healthy Vegetation
	5 6 4	Land/ Water
	7 5 3	Natural With Atmospheric Removal
	7 5 4	Shortwave Infrared
	6 5 4	Vegetation Analysis

Πίνακας 3.2: Συνδυασμοί Καναλιών Landsat 8

Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013

Ακολουθεί διάγραμμα που απεικονίζει τις διαφορές στα μήκη κύματος για τα διάφορα κανάλια μεταξύ του Landsat 8 και του Landsat 7:



Διάγραμμα 3.1: Σύγκριση Καναλιών Landsat 7-Landsat 8
Landsat Missions, http://landsat.usgs.gov/L8_band_combos.php 2013

3.2.3 Γεωμετρική και χρονική διακριτική ικανότητα

Όσον αφορά τη γεωμετρική διακριτική ικανότητα του Landsat 8 παρατηρούνται διαφοροποιήσεις τόσο μεταξύ των δεκτών όσο και μεταξύ των καναλιών. Πιο συγκεκριμένα, τα κανάλια 1-7, 9 του δέκτη OLI χαρακτηρίζονται από 30m γεωμετρικής ικανότητας, ενώ το κανάλι 8 του δέκτη OLI, το οποίο είναι το πανχρωματικό, χαρακτηρίζεται από 15m. Όσον αφορά τα θερμικά κανάλια 10, 11 του δέκτη TIRS συλλέγονται στα 100m, ωστόσο πραγματοποιούνται ενδιάμεσες επεξεργασίες προκειμένου να παραχθούν στα 30m, ώστε να είναι σε αντιστοιχία με τα προϊόντα του δέκτη OLI. [*Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013*]

Η χρονική διάρκεια που παράγονται τα προϊόντα είναι ανά 16 μέρες και σε διάστημα 8 ημερών από τα προϊόντα που προσφέρει η προηγούμενη γενιά δορυφόρων του Landsat 7. [*Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013*]

3.2.4 Τεχνικά χαρακτηριστικά προϊόντων

Τα προϊόντα που παρέχονται από το δορυφόρο Landsat 8 είναι 16-bit εικόνες σε μορφή GeoTIFF. Πρακτικά, πρόκειται για το κλασσικό μορφή εικόνων TIFF σε συνδυασμό με την απαραίτητη πληροφορία γεωαναφοράς. Το σύστημα αναφοράς που χρησιμοποιείται είναι το παγκόσμιο γεωδαιτικό σύστημα (WGS84) στο προβολικό σύστημα UTM. Επιπλέον, τα τελικά δεδομένα που παράγονται και είναι διαθέσιμα έχουν υποστεί γεωμετρική επεξεργασία μέσω κυβικής συνέλιξης (cubic convolution) και είναι προσανατολισμένα προς το βορρά. Τα προϊόντα παραλαμβάνονται σε συμπιεσμένη μορφή .tar.gz και καταλαμβάνουν 1GB συμπιεσμένα και 2GB ασυμπίεστα. [*Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013*]

Ακολουθεί πίνακας που συνοψίζει τα χαρακτηριστικά των δεδομένων που παρέχει ο δορυφόρος Landsat 8.

Μέγεθος εικονοστοιχείου κατά άξονα x στο έδαφος (m)	Band 1-7,9 (ms): 30 m Band 8 (pan): 15 m Band 10,11 (thermal): 30 m (resampled from 100m)
Μέγεθος εικονοστοιχείου κατά άξονα y στο έδαφος (m)	Band 1-7,9 (ms): 30 m Band 8 (pan): 15 m Band 10,11 (thermal): 30 m (resampled from 100m)
Τυπικός αριθμός γραμμών ανά εικόνα	7300
Αριθμός εικονοστοιχείων ανά γραμμή (στήλες)	7500
Κανάλια που περιέχονται στην εικόνα (αριθμητικά)	11
Τυπος ψηφιακών τιμών (προσημασμένοι ή όχι ακέραιοι αριθμοί, πραγματικοί αριθμοί, κλπ)	μη προσημασμένοι ακέραιοι 16-bit
Τύπος δεδομένων (vector,raster)	raster
Τυπικό μέγεθος ψηφιακού αρχείου	2 GB ασυμπίεστα
Τύπος ψηφιακού αρχείου (format raster)	Gtiff
Τυπικό εμβαδό περιοχής που καλύπτει μία εικόνα στο έδαφος (m ²)	$4.9 \cdot 10^{10}$

Πίνακας 3.3: Χαρακτηριστικά Δεδομένων Landsat 8
Landsat Missions, <http://landsat.usgs.gov/landsat8.php>, 2013

3.2.5 Παραδείγματα εφαρμογής έγχρωμων σύνθετων Landsat 8

Στη συγκεκριμένη ενότητα παρουσιάζονται διάφορα έγχρωμα σύνθετα που δημιουργήθηκαν με σκοπό την ανάδειξη συγκεκριμένων κάθε φορά χαρακτηριστικών ή θεματικών ενοτήτων των πολυφασματικών εικόνων.

Έγχρωμο σύνθετο R:4 G:3 B:2: Με το συγκεκριμένο έγχρωμο σύνθετο αποδίδεται το φυσικό χρώμα στις εικόνες. Οι διάφορες θεματικές κατηγορίες παρουσιάζονται με το πραγματικό τους χρώμα, διευκολύνοντας έτσι τη διαδικασία της φωτοερμηνείας.



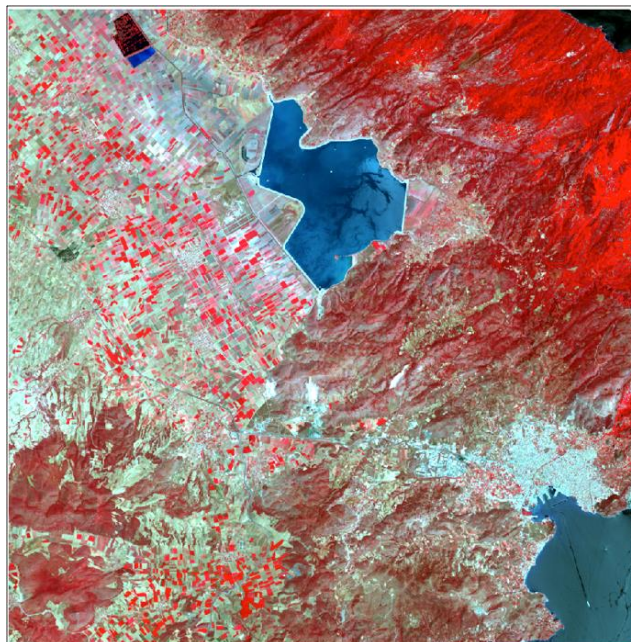
Εικόνα 3.1: Έγχρωμο σύνθετο 4-3-2 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία
Δημιουργία στο QGIS

Έγχρωμο σύνθετο R:7 G:6 B:4: Με το συγκεκριμένο έγχρωμο σύνθετο αποδίδεται το ψευδές χρώμα στις εικόνες και πραγματοποιείται εύκολα η φωτοερμηνεία του αστικού ιστού και του αστικού εξοπλισμού με ελαφρύ μωβ αποχρώσεις.



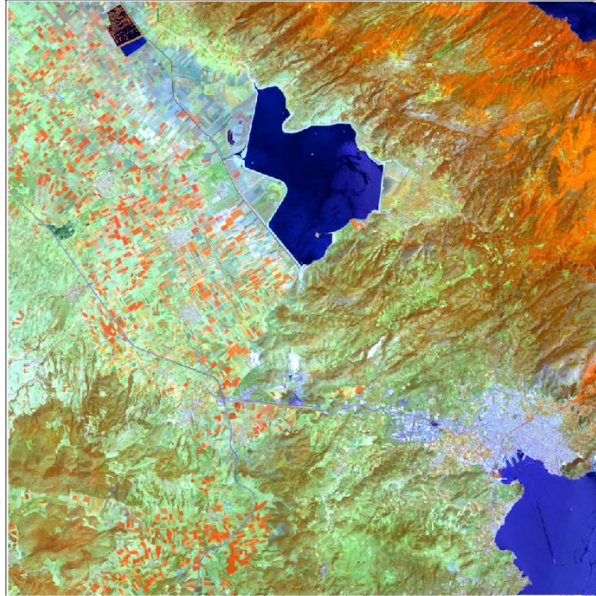
Εικόνα 3.2: Έγχρωμο σύνθετο 7-6-4 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία
Δημιουργία στο QGIS

Έγχρωμο σύνθετο R:5 G:4 B:3: Με το συγκεκριμένο έγχρωμο σύνθετο αποδίδεται το υπέρυθρο χρώμα, το οποίο χρησιμοποιείται για τη φωτοερμηνεία της βλάστησης απεικονίζοντάς την με κόκκινο χρώμα.



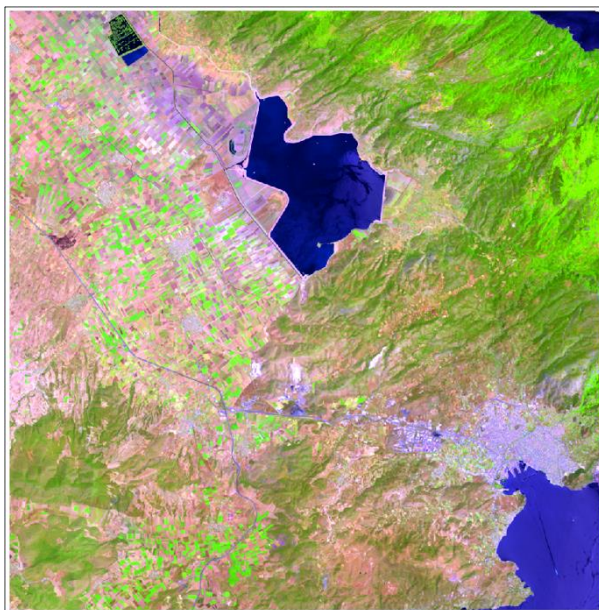
Εικόνα 3.3: Έγχρωμο σύνθετο 5-4-3 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία
Δημιουργία στο QGIS

Έγχρωμο σύνθετο R:5 G:6 B:2: Με το συγκεκριμένο έγχρωμο σύνθετο πραγματοποιείται η φωτοερμηνεία της υγιούς βλάστησης. Πιο συγκεκριμένα, οι περιοχές με αποχρώσεις του πορτοκαλί και του κόκκινου, σε αντίθεση με αυτές που έχουν αποχρώσεις του πράσινου, χαρακτηρίζονται ως υγιείς.



Εικόνα 3.4: Έγχρωμο σύνθετο 5-6-2 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία
Δημιουργία στο QGIS

Έγχρωμο σύνθετο R:6 G:5 B:2: Με το συγκεκριμένο έγχρωμο σύνθετο πραγματοποιείται η φωτοερμηνεία των καλλιεργειών, όπου οι πιο φωτεινές αποχρώσεις του πράσινου αναδεικνύουν τις καλλιέργειες που βρίσκονται σε υψηλό στάδιο ωρίμανσης και είναι υγιείς.



Εικόνα 3.5: Έγχρωμο σύνθετο 6-5-2 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία
Δημιουργία στο QGIS

3.3 Βασικές Εφαρμογές

Στο συγκεκριμένο κεφάλαιο εμφανίζεται πίνακας που παρουσιάζει συνοπτικά ενδεικτικές εφαρμογές του προγράμματος Landsat σε διάφορους επιστημονικούς τομείς.

Agriculture, Forestry and Range Resources	Land Use and Mapping	Geology	Hydrology	Coastal Resources	Environmental Monitoring
Discriminating vegetative, crop and timber types	Classifying land uses	Mapping major geologic features	Determining water boundaries and surface water areas	Determining patterns and extent of turbidity	Monitoring deforestation
Measuring crop and timber acreage	Cartographic mapping and map updating	Revising geologic maps	Mapping floods and flood plain characteristics	Mapping shoreline changes	Monitoring volcanic flow activity
Precision farming land management	Categorizing land capabilities	Recognizing land classifying certain rock types	Determining area extent of snow and ice coverage	Mapping shoals, reefs and shallow areas	Mapping and monitoring water pollution
Monitoring crop and forest harvests	Monitoring urban growth	Delineating unconsolidated rocks and soils	Measuring changes and extent of glacial features	Mapping and monitoring sea ice in shipping lanes	Determining effects of natural disasters
Determining range readiness, biomass and health	Aiding regional planning	Mapping volcanic surface deposits	Measuring turbidity and sediment patterns	Tracking beach erosion and flooding	Assessing drought impact
Determining soil conditions and associations	Mapping transportation networks	Mapping geologic landforms	Delineating irrigated fields	Monitoring coral reef health	Tracking oil spills

Πίνακας 3.4: Παραδείγματα εφαρμογών ανά τομέα προγράμματος Landsat
Landsat Applications, <http://landsat.gsfc.nasa.gov/?p=3501>, 2014

3.4 WELD: Web – ENABLED LANDSAT DATA

Στη συγκεκριμένη ενότητα παρουσιάζεται μία εφαρμογή που έχει αναπτυχθεί στα πλαίσια αξιοποίησης μεγάλων γεωχωρικών δεδομένων από τα δορυφορικά τηλεπισκοπικά δεδομένα του προγράμματος Landsat.

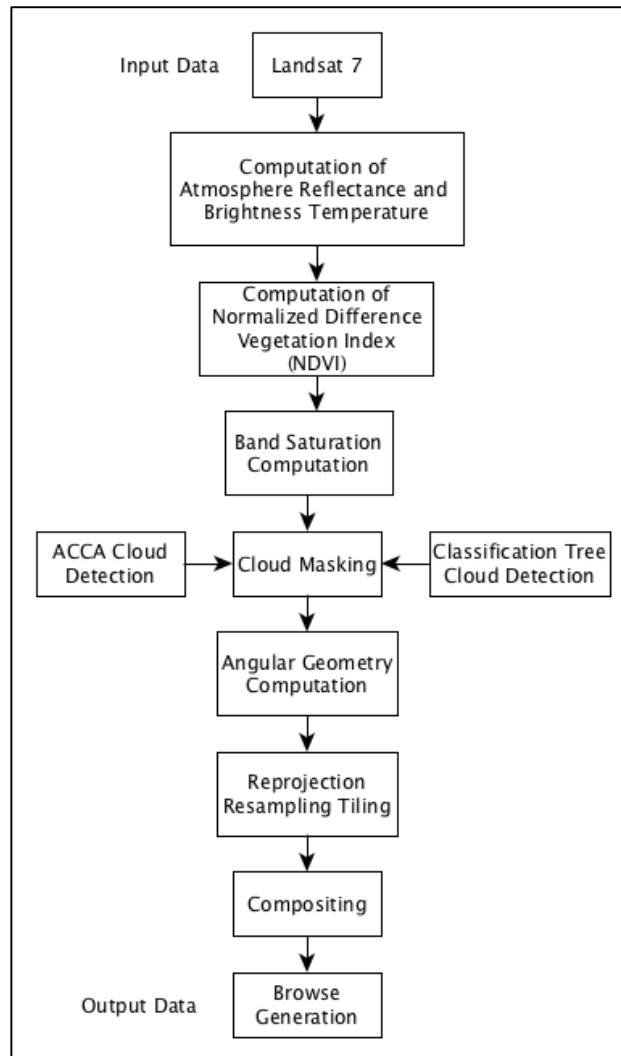
Ο σκοπός του προγράμματος της Nasa με ονομασία Making Earth System Data Records for Use in Research Environments (MEaSUREs) είναι να υποστηρίξει προγράμματα που σχετίζονται με τις επιστήμες της γης προμηθεύοντας αυτά με άρτια επεξεργασμένα δεδομένα. Στα πλαίσια του συγκεκριμένου προγράμματος

έχει αναπτυχθεί η διαδικτυακή πλατφόρμα WELD που στοχεύει στην επεξεργασία υψηλής ανάλυσης δεδομένων χρησιμοποιώντας τις τεχνικές τελευταίας τεχνολογίας (state of the art). Πιο συγκεκριμένα, στόχος του προγράμματος είναι η παραγωγή συνθετικών και απαλλαγμένων από σύννεφα μωσαϊκών χρησιμοποιώντας παράλληλα δεδομένα από MODIS και Landsat Enhanced Thematic Mapper plus (ETM+) και παράγοντας προϊόντα σε εβδομαδιαία, μηνιαία και χρονιαία χρονική βάση. Στα πλαίσια του συγκεκριμένου προγράμματος αναμένεται η δημιουργία παρόμοιων ποιοτικών προϊόντων για καλλιεργήσιμες και υδάτινες εκτάσεις.

[Roy, Ju, Kommareddy, & al, 2011]

Τα προϊόντα του προγράμματος WELD δημιουργούνται έτσι ώστε οι χρήστες να μην είναι υποχρεωμένοι να επεξεργάζονται ραδιομετρικά και γεωμετρικά τα δεδομένα. Το πρόγραμμα αυτό παρέχει τελικά προϊόντα σε τιμές ανακλαστικότητας με στόχο την αξιοποίησή τους σε πλήθος εφαρμογών που απαιτούν τη χρήση διαχρονικών δεδομένων. Μέχρι στιγμής εφαρμόζονται τεχνικές συγχώνευσης και βελτιστοποίησης σε δεδομένα Modis και Landsat με κάλυψη συννέφων μικρότερη του 80% για τις περιοχές των Ηνωμένων Εθνών και της Αλάσκας. Τα παραπάνω δεδομένα παρέχονται ελεύθερα στους χρήστες και στους οργανισμούς για παραπάνω από 8 χρόνια από το 2005. Τελικός στόχος του προγράμματος αποτελεί η παραγωγή και διάθεση μωσαϊκών απαλλαγμένων από σύννεφα σε παγκόσμια κλίμακα, διατηρώντας έτσι τα προϊόντα στο ίδιο υψηλό επίπεδο. *[Roy, Ju, Kommareddy, & al, 2011]*

Ακολουθεί το λογικό διάγραμμα που παρουσιάζει τη διαδικασία ανάλυσης για την παραγωγή του τελικού προϊόντος:



Διάγραμμα 3.2: WELD - Διαδικασία Παραγωγής Τελικού Προϊόντος

Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011

Στο στάδιο της σύνθεσης (compositing) πραγματοποιείται μια σειρά διαδικασιών σε επίπεδο εικονοστοιχείου λαμβάνοντας υπόψη διαχρονικά δεδομένα προκειμένου να πραγματοποιηθεί η τελική εξάλειψη των συννέφων, των σκιών καθώς και η ανάπλαση των «χαμένων» τιμών από το σύνολο των προηγούμενων βημάτων. Στα πλαίσια της συγκεκριμένης διαδικασίας ακολουθείται η σύνθεση του τελικού προϊόντος λαμβάνοντας υπόψη διάφορα κριτήρια μεταξύ εικόνων από διαδοχικές λήψεις. Στον πίνακα 3.5 παρουσιάζονται τα κριτήρια και στον πίνακα 3.6 παρουσιάζονται τα τελικά προϊόντα.

Priority	Compositing Criteria
1	If either fill: Select non-fill
2	If either saturated: Select unsaturated
3	If both saturated: Select the one with maximal Brightness Temperature
4	If one cloudy and one non-cloudy: Select non-cloudy
5	If one cloudy and one uncertain cloud: Select uncertain cloud if it has maximal Brightness Temperature or maximal NDVI, else select cloudy
6	If one non-cloudy and one uncertain cloud: Select non-cloud if it has maximal Brightness Temperature or maximal NDVI, else select uncertain cloud
7	If either or both “unvegetated” and both have NDVI < 0.5: Select the one with maximal Brightness Temperature
8	Select the one with maximal NDVI

Πίνακας 3.5: Κριτήρια Σύνθεσης Συγκρίνοντας δύο διαδοχικές λήψεις εικονοστοιχείου
Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011

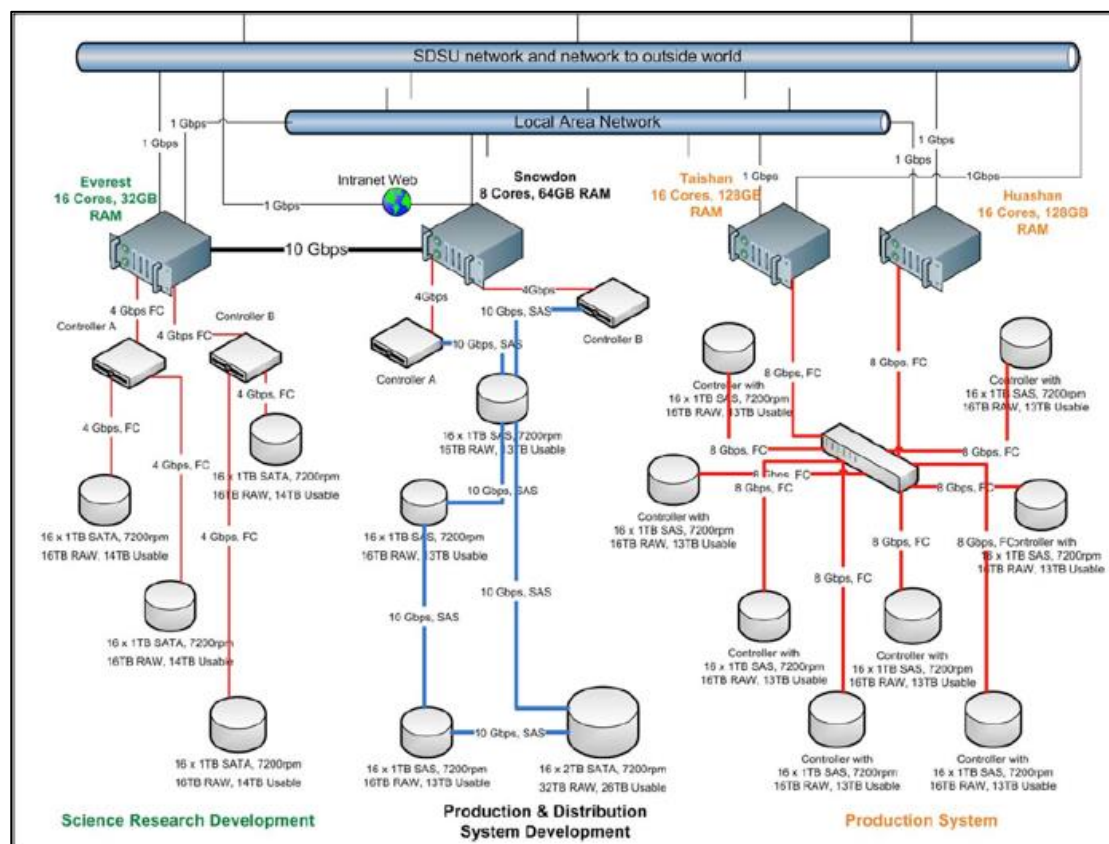
Product Type	Temporal Definition
<i>Annual</i>	The preceding year's December through the current year's November.
<i>Seasonal:</i>	
<i>Winter</i>	December, January, February
<i>Spring</i>	March, April, May
<i>Summer</i>	June, July, August
<i>Autumn</i>	September, October, November
<i>Monthly</i>	The days in each calendar month
<i>Weekly</i>	Consecutive 7-day products with <i>Week01</i> : January 1 to January 7, <i>Week02</i> : January 8 to January 14, ... , <i>Week 52</i> : December 24 to December 30 (non-leap years) or December 23 to December 29 (leap years), <i>Week 53</i> : December 30th to December 31st (leap years) or December 31st (non-leap years).

Πίνακας 3.6: Τελικά Προϊόντα Προγράμματος WELD
Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011

Όσον αφορά τη διαχείριση των δεδομένων, τα προϊόντα WELD δημιουργούνται σε μορφή Hierarchical Data Format (HDF) χρησιμοποιώντας σπειροειδή προσέγγιση ανάπτυξης μέσω προγραμμάτων σε γλώσσα προγραμματισμού C και σε λειτουργικό σύστημα Linux. Τα προγράμματα που παράγονται σχεδιάζονται με τέτοιο τρόπο ώστε να εκμεταλλεύονται στο μεγαλύτερο δυνατό βαθμό την κεντρική μονάδα επεξεργασίας (CPU) καθώς και τη μνήμη ανάγνωσης (RAM). Προς το παρόν, η διαδικασία στο σύνολό της καθώς και τα επιμέρους προγράμματα απαιτούν την παρέμβαση του μελετητή και η διαδικασία δεν είναι πλήρως αυτοματοποιημένη

λόγω της πολυπλοκότητας της διαδικασίας ανάλυσης που διεξάγεται. Ιδιαίτερη προσοχή δίνεται στην κατηγοριοποίηση των προϊόντων καθώς και των μεταδεδομένων έτσι ώστε να βελτιστοποιούνται οι διαδικασίες αποθήκευσης και επεξεργασίας.

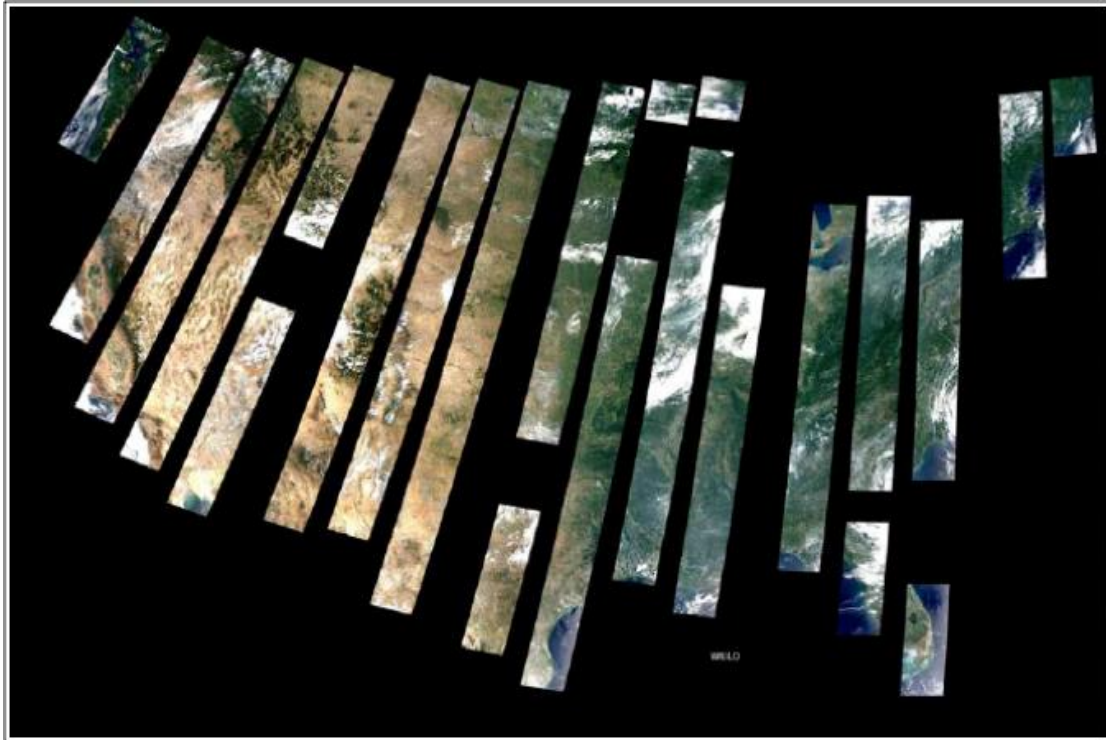
Προκειμένου να παραχθούν τα συγκεκριμένα προϊόντα είναι απαραίτητη η χρήση της τεχνολογίας cloud καθώς και η λειτουργία υπερ-υπολογιστών για την παραγωγή και διάθεση των δεδομένων στους χρήστες. Στην εικόνα που ακολουθεί παρουσιάζεται η αρχιτεκτονική του υπολογιστικού υλικού που χρησιμοποιείται στα πλαίσια του προγράμματος WELD. [Roy, Ju, Kommareddy, & al, 2011]



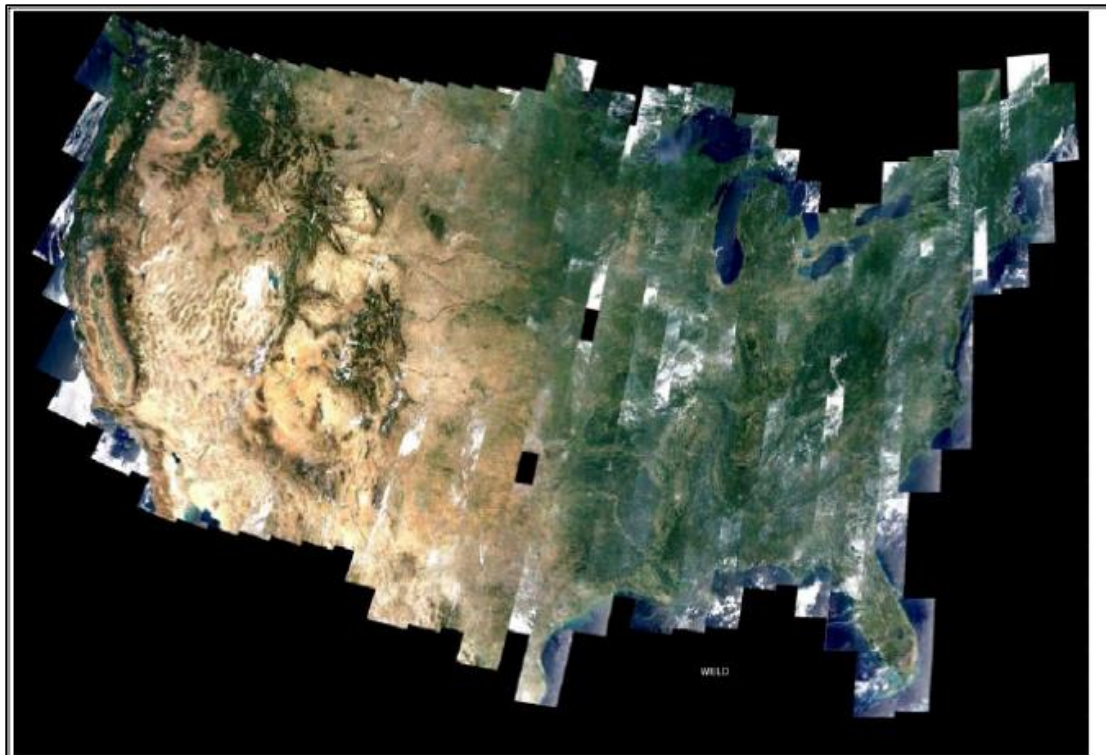
Εικόνα 3.6: Αρχιτεκτονική Υπολογιστικού Συστήματος WELD

Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011

Ακολουθούν παραδείγματα των προϊόντων WELD σε εβδομαδιαία, μηνιαία και χρονιαία χρονική κλίμακα.



Εικόνα 3.7: Παράδειγμα προϊόντος WELD CONUS σε εβδομαδιαία κλίμακα (15-21 Ιουλίου, 2008)
Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011



Εικόνα 3.8: Παράδειγμα προϊόντος WELD CONUS σε μηνιαία κλίμακα (Ιούλιος, 2008)
Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011



Εικόνα 3.9: Παράδειγμα προϊόντος WELD CONUS σε χρονιαία κλίμακα (έτος 2008)
Web Enabled Landsat Data (WELD) Products, David Roy, Junchang Ju, Indrani Kommareddy et al, 2011

Κεφάλαιο 4: Τεχνολογίες Αιχμής στην Ταξινόμηση και Ανίχνευση Μεταβολών

4.1 Εισαγωγή

Ως ανίχνευση μεταβολών (change detection) ορίζεται η τεχνολογία μέσω της οποίας εξακριβώνονται μεταβολές ειδικών χαρακτηριστικών σε ορισμένο χρονικό διάστημα και ανιχνεύονται χωρικές κατανομές των αλλαγών τόσο με βάση τη χρήση ποιοτικής όσο και ποσοτικής πληροφορίας. Όσον αφορά την ποσοτική ανάλυση για τον προσδιορισμό των χαρακτηριστικών που μεταβλήθηκαν χρησιμοποιούνται δεδομένα και διαδικασίες από την επιστήμη της τηλεπισκόπησης. Για τον ακριβή προσδιορισμό των μεταβολών γίνεται αφενός χρήση πολυφασματικών εικόνων τηλεπισκόπησης και αφετέρου ψηφιακών μοντέλων εδάφους λαμβάνοντας, παράλληλα, υπόψη την ποιοτική πληροφορία για την εκάστοτε περιοχή. [Kandare, 2010]

Η ανίχνευση μεταβολών μπορεί να εφαρμοστεί για τον εντοπισμό και προσδιορισμό αλλαγών σε διαφορετικές χρονικές και χωρικές κλίμακες. Πιο συγκεκριμένα, εξετάζονται περιπτώσεις βραχυπρόθεσμων αλλαγών (π.χ. καιρικά φαινόμενα), κυκλικών μεταβολών (π.χ. εποχικής φαινολογίας – seasonal phenology), κατευθυντήριων αλλαγών (π.χ. αστική ανάπτυξη) καθώς και αλλαγών που στηρίζονται σε κάποιο γεγονός (π.χ. μετά από πυρκαγιά). Για κάθε διαφορετική χρονική και χωρική κλίμακα εφαρμόζονται διαφορετικές διαδικασίες ανάλυσης λαμβάνοντας κάθε φορά υπόψη τα ιδιαίτερα χαρακτηριστικά της περιοχής και του γεγονότος που συντέλεσε στη δημιουργία των μεταβολών. [Herold, 2010]

Για τη βέλτιστη εφαρμογή διαδικασιών ανίχνευσης μεταβολών προτείνεται η χρήση διαδοχικών τηλεπισκοπικών δεδομένων (χρονοσειρές) καθώς και δεδομένων από διαφορετικούς δέκτες. Ωστόσο, είναι απαραίτητο να ακολουθούνται τα κατάλληλα βήματα προεργασίας προκειμένου να ελαττωθούν τα ραδιομετρικά, ατμοσφαιρικά και γεωμετρικά σφάλματα μεταξύ των δεκτών. Πιο συγκεκριμένα, οι διαδικασίες που πρέπει να πραγματοποιηθούν πριν την ανίχνευση μεταβολών παρουσιάζονται παρακάτω: [Campbell & Wynne, 2011]

- Χρήση δεδομένων από ίδιους ή βαθμονομημένους τηλεπισκοπικούς δέκτες
- Χρήση δεδομένων από την ίδια ώρα της ημέρας, με το ίδιο στιγμιαίο οπτικό πεδίο και γωνία
- Για αναλύσεις που αναφέρονται σε χρονική κλίμακα είναι απαραίτητο η απόκτηση δεδομένων κατά τη διάρκεια της ίδιας εποχής για να ελαχιστοποιηθούν τα σφάλματα λόγω φαινολογικών αλλαγών
- Εφαρμογή coregistration και κατά προτίμηση εντός δύο δεκάτων του pixel ή λιγότερο [Dai & Khorram, 1999]
- Τα δεδομένα που θα χρησιμοποιηθούν για τη διαδικασία ανάλυσης πρέπει να είναι απαλλαγμένα από σύννεφα
- Χρήση ραδιομετρικά διορθωμένων δεδομένων σε top-of-atmosphere reflectance, ή ακόμα καλύτερα στην πραγματική επιφάνεια

- Χρήση δεδομένων που δεν έχουν επηρεαστεί από άλλες συνθήκες, οι οποίες δεν θεωρούνται μέρος την ανάλυσης (π.χ. παρουσία υγρασίας)

Παρόλο που είναι πολύ δύσκολο πολλές φορές να καλύπτονται όλες οι παραπάνω συνθήκες ταυτόχρονα, είναι απαραίτητο να αναφερθούν καθώς η αδυναμία για ικανοποίησής τους οδηγεί σε εσφαλμένα συμπεράσματα στο τέλος της διαδικασίας ανάλυσης.

4.2 Μέθοδοι Ανίχνευσης Μεταβολών

Υπάρχουν πολλοί τύποι μεθόδων ανίχνευσης μεταβολών χρησιμοποιώντας πολυφασματικά τηλεπισκοπικά δεδομένα. Σε γενικές γραμμές, οι διάφορες μέθοδοι μπορούν να ταξινομηθούν σε τρεις κατηγορίες: ανάλυση φασματικών χαρακτηριστικών, διανυσματική ανάλυση καθώς και ανάλυση χρονοσειρών. Μεγαλύτερη εφαρμογή στον τομέα της παρακολούθησης των αλλαγών στην επιφάνεια της γης έχουν οι μέθοδοι ανάλυσης χρονοσειρών που βασίζονται σε δεδομένα τηλεπισκοπικών δεκτών και λιγότερο στην αξιοποίηση ποιοτικής πληροφορίας. [Kandare, 2010] Στα κεφαλαία που ακολουθούν παρουσιάζονται οι σημαντικότερες από τις παραπάνω μεθόδους με αντίστοιχα παραδείγματα. [Karantzalos, 2014]

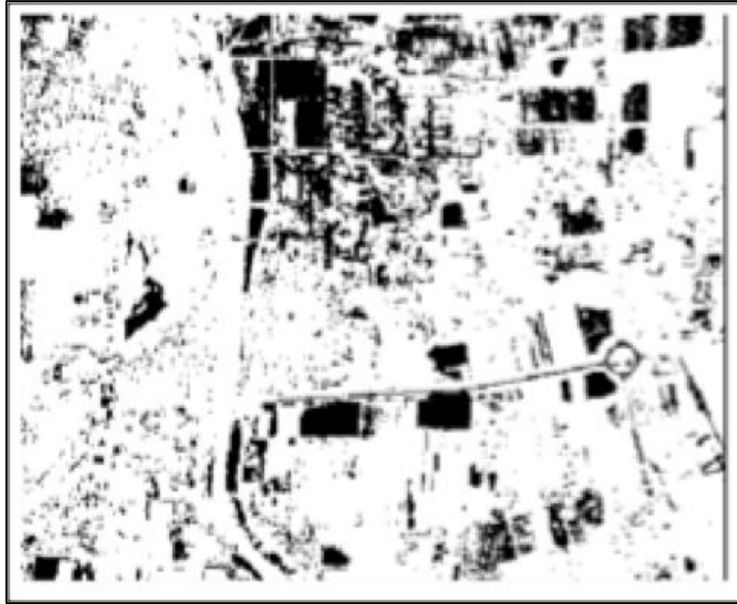
4.2.1 Μέθοδοι άγλεβρας εικόνας

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται οι μέθοδοι ανίχνευσης μεταβολών που στηρίζονται σε αλγεβρικές πράξεις μεταξύ διαχρονικών δεδομένων. Πιο συγκεκριμένα, παρουσιάζεται η μέθοδος αφαίρεσης εικόνων, η μέθοδος λόγου εικόνων καθώς και η μέθοδος «change vector analysis».

Αφαίρεση εικόνων

Πρόκειται για μια μέθοδο που έχει εκτεταμένη εφαρμογή σε πολλούς τύπους γεωχωρικών προβλημάτων λόγω της απλότητας και της αποτελεσματικότητας της διαδικασίας. Ο καθορισμός των περιοχών που άλλαξαν και των αντίστοιχων που παρέμειναν αμετάβλητες εξαρτάται από την εφαρμογή κατάλληλων ορίων των γκρίζων τόνων της εικόνας. Η αφαίρεση των τόνων του γκρι υποδεικνύει, στην εικόνας αφαίρεσης, τα εικονοστοιχεία που μεταβλήθηκαν. Πλεονέκτημα της συγκεκριμένης μεθόδου είναι πως ο καθορισμός του κατωφλίου εφαρμόζεται μόνο μία φορά, ωστόσο σημαντικό μειονέκτημα αποτελεί το γεγονός ότι η επιλογή του κατωφλίου επηρεάζει σημαντικά τη διαδικασία ανάλυσης καθώς και ότι η συγκεκριμένη μέθοδος δεν παρουσιάζει τις αλλαγές στις χρήσεις γης. Έτσι, η μέθοδος αυτή δεν προτείνεται σε εφαρμογές ανίχνευσης μεταβολών σε περιοχές όπου υπάρχουν συχνές εναλλαγές χρήσεων γης καθώς αδυνατεί να ανιχνεύσει τη φύση των αλλαγών και απαιτείται περεταίτω ανάλυση. Στην εικόνα που δημιουργείται μέσω της πράξης της αφαίρεσης, οι διαφορές στις περιοχές με λίγη ή

καθόλου αλλαγή είναι μικρές ενώ σε μεγάλες αλλαγές οι διαφορές είναι πολύ μεγάλες με θετικό ή αρνητικό πρόσημο. [Kandare, 2010] Ακολουθεί παράδειγμα εικόνας που προέκυψε μετά την εφαρμογή της πράξης της αφαίρεσης.



Εικόνα 4.1: Παράδειγμα εφαρμογής πράξης αφαίρεσης

The time series change detection methods of remote sensing, Kaja Kandare, isprs 2010

Λόγος εικόνων

Πρόκειται για άλλη μία μέθοδο που ανήκει στην κατηγορία της άλγεβρας εικόνων. Πιο συγκεκριμένα, στη συγκεκριμένη μέθοδο δημιουργείται μία νέα εικόνα από την πράξη της διαίρεσης. Η τιμή της μονάδας δηλώνει πως δεν υπάρχει μεταβολή ενώ οι τιμές που είναι μεγαλύτερες ή μικρότερες της μονάδας δηλώνουν μεταβολές. Προκειμένου να προσδιοριστεί το κατώφλι για την ερμηνεία του μεγέθους της αλλαγής απαιτείται ο προσδιορισμός ενός κατώτατου ορίου που επιλέγεται ανάλογα με την περιοχή μελέτης, τις συνθήκες λήψης της εικόνας καθώς την αντίστοιχη εφαρμογή. Μια συνήθης μέθοδος προσδιορισμού του κατωφλίου αποτελεί η χρήση του ιστογράμματος του λόγου της εικόνας που δημιουργήθηκε. Έχει παρατηρηθεί πως η συγκεκριμένη μέθοδος έχει ικανοποιητικά αποτελέσματα για την ανίχνευση μεταβολών στις καλλιέργειες καθώς και σε εφαρμογές που κύριο ενδιαφέρον έχει η υφή των αντικειμένων. Ωστόσο, παρατηρούνται αδυναμίες και ευαισθησίες της συγκεκριμένης διαδικασίας που σχετίζονται κυρίως με τη γωνία του ήλιου, την ύπαρξη θορύβου στις αρχικές εικόνες καθώς και στην επιρροή της ατμόσφαιρας στη μετάδοση της ακτινοβολίας. Επιπλέον, όμοια με τη μέθοδο της αφαίρεσης εικόνων, στη συγκεκριμένη περίπτωση δεν είναι δυνατόν να εντοπιστούν οι αλλαγές των μεταβολών και τα χαρακτηριστικά αυτών. [Kandare, 2010] Ακολουθεί παράδειγμα εικόνας που προέκυψε μετά την εφαρμογή της πράξης του λόγου.

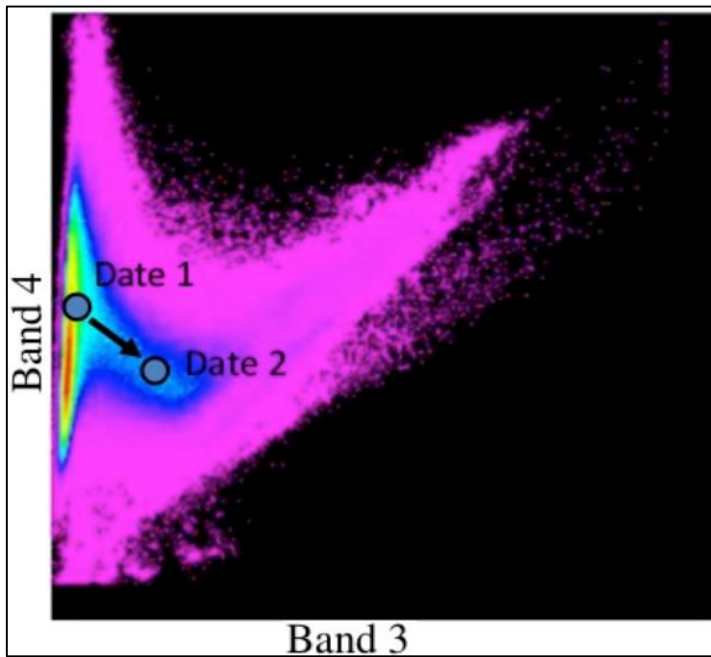


Εικόνα 4.2: Παράδειγμα εφαρμογής πράξης λόγου

The time series change detection methods of remote sensing, Kaja Kandare, isprs 2010

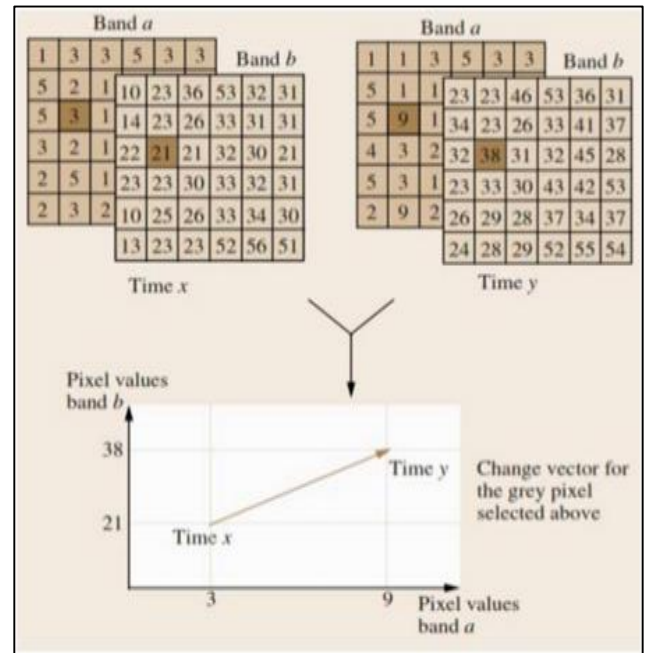
Change vector analysis

Η ανάλυση αλλαγών του φορέα (CVA) είναι μια πολυδιάστατη επέκταση της μεθόδου αφαίρεσης εικόνων. Οι διαφορές μεταξύ των ραδιομετρικών τιμών σε πολλαπλά κανάλια παρουσιάζουν τους φορείς αλλαγής κατεύθυνσης (change direction) και το μέγεθος της αλλαγής (change intensity). Εξετάζονται δύο περιπτώσεις όπου στην πρώτη υπολογίζεται το όριο αλλαγής, που υποδεικνύεται με ένα γραμμικό βέλος (vector arrow), ενώ στη δεύτερη απεικονίζεται το ζευγάρι των εικονοστοιχείων, των οποίων οι ραδιομετρικές τιμές ανήκουν εντός μιας ζώνης ορίου και ταξινομούνται ως εικονοστοιχεία χωρίς μεταβολή. Με αυτόν τον τρόπο, παράγεται μία εικόνα που υποδεικνύει τόσο την περίπτωση μεταβολών όσο και την ταξινόμηση της κάθε μεταβολής. Σε αντίθεση με τις προηγούμενες αλγεβρικές μεθόδους η συγκεκριμένη μέθοδος είναι ικανή να ανιχνεύσει το είδος της μεταβολής και να εξαχθούν συμπεράσματα σε πολλαπλά κανάλια. Ωστόσο, και η συγκεκριμένη μέθοδος αντιμετωπίζει το σημαντικό μειονέκτημα των αλγεβρικών μεθόδων, που σχετίζεται με την κατάλληλη εύρεση της τιμής του κατωφλίου. [Herold, 2010]



Εικόνα 4.3: Παράδειγμα CVA σε δύο κανάλια

A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012



Εικόνα 4.4: Άλγεβρική πράξη CVA

A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012

Ανασκόπηση Μεθόδων Άλγεβρας Εικόνας

Ακολουθεί συγκεντρωτικός πίνακας που παρουσιάζει τα κύρια χαρακτηριστικά, τα πλεονεκτήματα και μειονεκτήματα καθώς και χαρακτηριστικά παραδείγματα εφαρμογής των μεθόδων που παρουσιάστηκαν παραπάνω.

Μέθοδος	Χαρακτηριστικά	Πλεονεκτήματα	Μειονεκτήματα	Παραδείγματα Εφαρμογής
Εικόνα Αφαίρεση	Αφαίρεση της πρώτης χρονολογικά εικόνας από τη δεύτερη σε επίπεδο εικονοστοιχείου	Απλή, γρήγορη, εύκολη ερμηνεία αποτελεσμάτων	Δεν παρουσιάζει αναλυτικά χαρακτηριστικά των μεταβολών, προϋποθέτει την επιλογή κατωφλίου	Δασικές εκτάσεις, γενικές κατηγορίες χρήσεων γης
Εικόνα Λόγου	Υπολογίζει το λόγο εικόνων ανά κανάλι και εικονοστοιχείο	Μείωση σφαλμάτων από τη γωνία του ήλιου, λόγω σκιών και αναγλύφου	Αναποτελεσματικότητα σε συγκεκριμένες κατηγορίες χρήσεων γης	Γενικές κατηγορίες χρήσεων γης
CVA	Εύρεση πιθανών μεταβολών και έντασής τους με τη μορφή διανυσματικών βέλων	Αναλυτική παρουσίαση μεταβολών σε πολλαπλά κανάλια	Δυσκολία στην ερμηνεία των αποτελεσμάτων	Εφαρμογές σε διάφορες κατηγορίες π.χ. δασικές εκτάσεις, αστικές περιοχές, καλλιέργειες κτλ

Πίνακας 4.1: Ανασκόπηση μεθόδων ανίχνευσης μεταβολών με άλγεβρα εικόνας

A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012

4.2.2 Μέθοδοι μετασχηματισμού δεδομένων

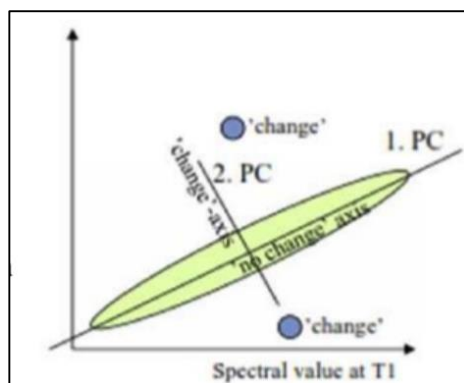
Στο συγκεκριμένο κεφάλαιο παρουσιάζονται δύο μέθοδοι μετασχηματισμού δεδομένων, που αξιοποιούνται στη διαδικασία ανάλυσης για την ανίχνευση μεταβολών σε διαχρονικά δεδομένα. Πιο συγκεκριμένα, παρουσιάζεται ο μετασχηματισμός της ανάλυσης κύριων συνιστωσών και ο μετασχηματισμός Tasseled cap.

Ανάλυση κύριων συνιστωσών (PCA)

Η ανάλυση κύριων συνιστωσών είναι μία στατιστική διαδικασία που χρησιμοποιεί ένα ορθογώνιο μετασχηματισμό για να μετατρέψει ένα σύνολο παρατηρήσεων, συνήθως συσχετισμένων, σε ένα σύνολο γραμμικών ασυσχέτιστων, που ονομάζονται κύριες συνιστώσες. Ο αριθμός των κύριων συνιστωσών είναι μικρότερος ή ίσος των αρχικών παρατηρήσεων, με την πρώτη κύρια συνιστώσα να έχει τη μεγαλύτερη διακύμανση ενώ κάθε επόμενη συνιστώσα χαρακτηρίζεται από υψηλή διακύμανση, αλλά πρέπει παράλληλα να είναι ασυσχέτιστη με την προηγούμενη. Ωστόσο, επισημαίνεται πως οι κύριες συνιστώσες είναι ευαίσθητες καθώς επηρεάζονται από τα αρχικά δεδομένα.

[http://en.wikipedia.org/wiki/Principal_component_analysis, 2014]

Μέσω της ανάλυσης κύριων συνιστωσών προκύπτουν νέα ασυσχέτιστα δεδομένα, τα οποία όχι μόνο μειώνουν τον όγκο των δεδομένων, αλλά με τις κατάλληλες διαδικασίες ανάλυσης οδηγούν σε πιο ασφαλή και έγκυρα συμπεράσματα. Στα πλαίσια των μεθόδων ανίχνευσης μεταβολών η ανάλυση κύριων συνιστωσών μπορεί να χρησιμοποιηθεί με ποικίλους τρόπους. Πιο συγκεκριμένα, είναι δυνατή η εφαρμογή της ανάλυσης των κύριων συνιστωσών στο σύνολο των δεδομένων, σε όλες τις χρονικές στιγμές, και η μετέπειτα ανάλυση της μικρότερης σημασίας συνιστώσας που προέκυψε από το συνδυασμό των δεδομένων. Επιπλέον, είναι δυνατή η εφαρμογή της ανάλυσης κύριων συνιστωσών για κάθε χρονική περίοδο ξεχωριστά και μετέπειτα η εφαρμογή κάποιας αλγεβρικής μεθόδου (π.χ. διαφορά) για την ανίχνευση των μεταβολών.



Εικόνα 4.5: Ανίχνευση μεταβολών με PCA

A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012

Tasseled cap

Όμοια με την ανάλυση κύριων συνιστωσών ο μετασχηματισμός Tasseled Cap μειώνει τα δεδομένα συμπιέζοντας τα φασματικά δεδομένα τους σε λιγότερα κανάλια, ανάλογα με τα φυσικά χαρακτηριστικά τους (Crist & Cicone, 1984). Στο μετασχηματισμό Tasseled Cap η αρχική εικόνα αναλύεται μέσω ενός γραμμικού μετασχηματισμού ανάλογα με τις ψηφιακές τιμές κάθε καναλιού. Οι συντελεστές των γραμμικών εξισώσεων αναπτύχθηκαν μέσω της ανάλυσης πολλών εικόνων και για κάθε αισθητήρα ξεχωριστά. Όμοια με την ανάλυση κύριων συνιστωσών, ο μετασχηματισμός Tasseled Cap οδηγεί στη δημιουργία μιας νέας εικόνας με τον ίδιο ή μικρότερο αριθμό καναλιών σε σχέση με την αρχική εικόνα κρατώντας, ωστόσο, τα πιο σημαντικά χαρακτηριστικά. Οι τρεις πρώτες συνιστώσες ανταποκρίνονται σε φαινόμενα όπως είναι η μεταβολή της φωτεινότητας, του πρασίνου και της υγρασίας. [Crist & Kauth, 1984]

Ανασκόπηση Μεθόδων Μετασχηματισμού Δεδομένων

Ακολουθεί συγκεντρωτικός πίνακας που παρουσιάζει τα κύρια χαρακτηριστικά, τα πλεονεκτήματα και μειονεκτήματα καθώς και χαρακτηριστικά παραδείγματα εφαρμογής των μεθόδων μετασχηματισμού που παρουσιάστηκαν παραπάνω στα πλαίσια της ανίχνευσης μεταβολών.

Μέθοδος	Χαρακτηριστικά	Πλεονεκτήματα	Μειονεκτήματα	Παραδείγματα Εφαρμογής
PCA	Δημιουργία ασυσχέτιστων δεδομένων και εφαρμογή αλγεβρικών πράξεων ή περαιτέρω ανάλυση	Μείωση δεδομένων, ασυσχέτιστα κανάλια	Τα παραγόμενα δεδομένα εξαρτώνται από τα αρχικά, αδυναμία ερμηνείας των μεταβολών	Τροπικά δάση, γενικές κατηγορίες χρήσεων γης, αστικές περιοχές
Tasseled Cap	Δημιουργία ασυσχέτιστων δεδομένων και εφαρμογή αλγεβρικών πράξεων ή περαιτέρω ανάλυση	Μείωση δεδομένων, ασυσχέτιστα κανάλια	Αδυναμία ερμηνείας των μεταβολών, διαφοροποίηση των συντελεστών του γραμμικού μετασχηματισμού για κάθε δέκτη	Δασικές εκτάσεις, γενικές κατηγορίες χρήσεων γης, καλλιεργήσιμες εκτάσεις

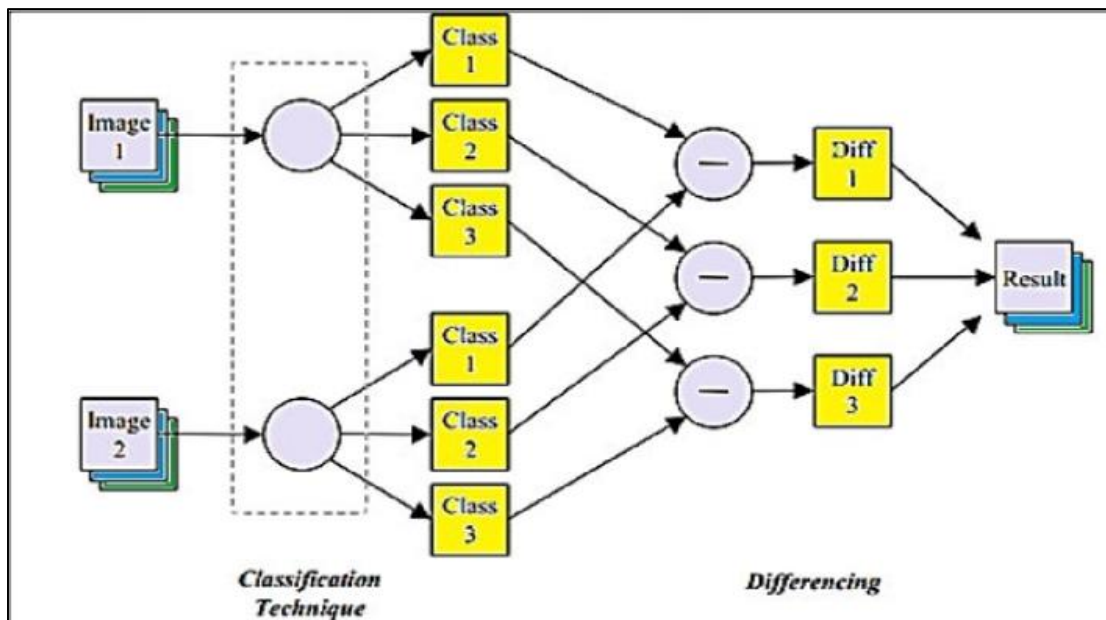
Πίνακας 4.2: Ανασκόπηση μεθόδων ανίχνευσης μεταβολών με εφαρμογή μετασχηματισμού
A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012

4.2.3 Μέθοδοι στηριζόμενοι σε ταξινόμηση

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται δύο μέθοδοι ανίχνευσης μεταβολών που στηρίζονται σε διαδικασίες ταξινόμησης. Πιο συγκεκριμένα, παρουσιάζεται μία μέθοδος που στηρίζεται σε αλγορίθμους επιβλεπόμενης ταξινόμησης καθώς και μία μέθοδος με εφαρμογή μη επιβλεπόμενης ταξινόμησης.

Ανάλυση μετα-ταξινόμησης

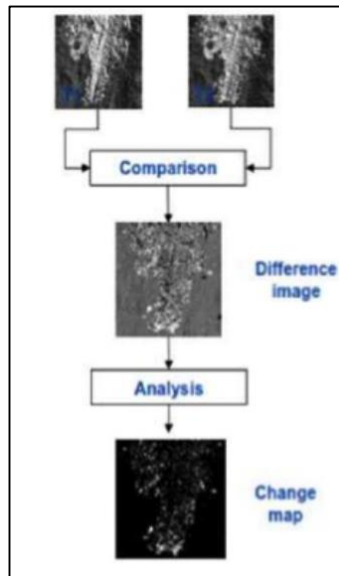
Στη συγκεκριμένη μέθοδο ταξινομείται ξεχωριστά κάθε εικόνα για κάθε χρονική στιγμή και στη συνέχεια συγκρίνονται οι τιμές κάθε τάξης σε επίπεδο εικονοστοιχείου. Η αποτελεσματικότητα της συγκεκριμένης μεθόδου εξαρτάται από την επιλογή του κατάλληλου αλγορίθμου ταξινόμησης καθώς και του αντίστοιχου μοντέλου εκπαίδευσης. Τα πλεονεκτήματα της συγκεκριμένης μεθόδου εστιάζονται στο γεγονός ότι δεν είναι απαραίτητη η εφαρμογή ραδιομετρικών διορθώσεων, εφόσον χρησιμοποιούνται δεδομένα από τον ίδιο δέκτη, καθώς και στο ότι γίνονται αντιληπτές τόσο οι μεταβολές όσο και τα χαρακτηριστικά τους. Μειονεκτήματα της μεθόδου είναι η διαφορετικότητα των αποτελεσμάτων ανάλογα από τον αλγόριθμο ταξινόμησης που θα επιλεγεί καθώς και η αναγκαιότητα για δεδομένα εδάφους (ground truth) έτσι ώστε να αξιολογηθεί η διαδικασία. [Bhatt, 2012]



Διάγραμμα 4.1: Post classification analysis in change detection
Change detection process and techniques, Decker, 2013

Μη επιβλεπόμενες τεχνικές

Ακολουθείται η ίδια διαδικασία με την προηγούμενη μέθοδο με τη διαφορά ότι στη συγκεκριμένη γίνεται επιλογή αλγορίθμου μη επιβλεπόμενης ταξινόμησης και έτσι δεν είναι απαραίτητη η δημιουργία δεδομένων εκπαίδευσης. Στο διάγραμμα 4.2 που ακολουθεί απεικονίζεται η διαδικασία της ανίχνευσης μεταβολών με αλγορίθμους μη επιβλεπόμενης ταξινόμησης και στον πίνακα 4.3 παρουσιάζονται οι κύριες διαφορές μεταξύ επιβλεπόμενων και μη επιβλεπόμενων αλγορίθμων όσον αφορά στην ανίχνευση μεταβολών.



Διάγραμμα 4.2: Unsupervised classification analysis in change detection

A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012

	Επιβλεπόμενη	Μη επιβλεπόμενη
Επίπεδο ανίχνευσης μεταβολών	Σε επίπεδο απόφασης που καθορίζεται από τον μελετητή	Σε επίπεδο δεδομένων
Πληροφορία μεταβολής	Ονομασία κατηγοριών και διαφοροποιήσεις	Ξεχωρίζει αλλαγές ή όχι
Υπολογισμοί για ανίχνευση μεταβολών	Απευθείας από τα δεδομένα εκπαίδευσης	Χρειάζεται περαιτέρω επεξεργασία
Γνώση εδάφους	Απαιτείται	Δεν απαιτείται
Φασματική πληροφορία	Σε πολυφασματικά	Κυρίως σε κάθε κανάλι
Αναγκαία δεδομένα	Πολύγωνα εκπαίδευσης	Δεν χρειάζονται

Πίνακας 4.3: Σύγκριση επιβλεπόμενης και μη επιβλεπόμενης ταξινόμησης για ανίχνευση μεταβολών

A review of change detection techniques, Indian Institute of Technology, Abhishek Bhatt, 2012

4.3 Ανίχνευση Μεταβολών σε Μεγάλα Γεωχωρικά Δεδομένα

Για την ανίχνευση μεταβολών αξιοποιώντας μεγάλα γεωχωρικά δεδομένα οι συμβατικές μέθοδοι δεν αντιμετωπίζουν αποτελεσματικά το πρόβλημα. Στις περισσότερες περιπτώσεις απαιτείται η εφαρμογή πολλαπλών συνδυαστικών μεθόδων σε όλα τα στάδια της διαδικασίας ανάλυσης. Επιπλέον, είναι αναγκαία η εφαρμογή σύνθετων διαδικασιών προεργασίας δεδομένων καθώς και η αξιοποίηση διαφορετικού τύπου δεδομένων (διανυσματικών, εικόνων από διάφορους δέκτες καθώς και ποιοτικής πληροφορίας). Για την ανίχνευση μεταβολών είναι απαραίτητη η χρήση εξειδικευμένων αλγορίθμων ταξινόμησης που στηρίζονται σε αλγορίθμους υπολογιστικής νοημοσύνης. Οι συγκεκριμένοι αλγόριθμοι εκμεταλλεύονται διαφόρου τύπου δεδομένων και μέσω διαδικασιών εξόρυξης δεδομένων (data mining) δημιουργούν στατιστικά μοντέλα που αντιμετωπίζουν αποτελεσματικότερα τα προβλήματα ταξινόμησης. Παραδείγματα τέτοιου είδους αλγορίθμων είναι ο

SVM, τα στατιστικά μοντέλα decision trees, random forest και άλλα. [Zhe Zhu et al, 2014]

Στα κεφάλαια που ακολουθούν παρουσιάζονται οι κυριότεροι αλγόριθμοι ταξινόμησης που χρησιμοποιούν την τεχνολογία της μηχανικής μάθησης (machine learning) και χρησιμοποιούνται σε εφαρμογές ανίχνευσης μεταβολών σε μεγάλα διαχρονικά γεωχωρικά δεδομένα. Οι συγκεκριμένοι αλγόριθμοι εντάσσονται στην ευρύτερη κατηγορία των διαδικασιών εξόρυξης δεδομένων (data mining). [S. Lhermitte et al, 2011]

4.4 Data Mining

Η εξόρυξη δεδομένων (data mining) είναι μία αναλυτική διαδικασία που σχεδιάστηκε για την ανάλυση μεγάλων δεδομένων με σκοπό την αναζήτηση προτύπων ή συστηματικών σχέσεων μεταξύ των μεταβλητών. Στη συνέχεια, επικυρώνονται τα παραπάνω πορίσματα για την ανίχνευση μοτίβων σε νέα υποσύνολα δεδομένων σε εφαρμογές που απαιτείται πρόβλεψη χαρακτηριστικών. Η διαδικασία της εξόρυξης δεδομένων αποτελείται από τρία στάδια: 1) αρχική έρευνα, 2) μοντελοποίηση και επικύρωση, 3) ανάπτυξη και εφαρμογή του μοντέλου σε νέα δεδομένα ώστε να παραχθούν προβλέψεις.

[<https://www.statsoft.com/Textbook/Data-Mining-Techniques>, 2014]

Στις εφαρμογές εξόρυξης δεδομένων χρησιμοποιούνται σε ευρεία κλίμακα διαδικασίες μηχανικής μάθησης μέσω αλγορίθμων ταξινόμησης. Παραδείγματα τέτοιου είδους αλγορίθμων ταξινόμησης είναι τα παρακάτω:

[http://en.wikipedia.org/wiki/Statistical_classification, 2014]

- Linear classifiers
 - Fisher's linear discriminant
 - Logistic regression
 - Naive Bayes classifier
 - Perceptron
- Support vector machines
 - Least squares support vector machines
- Quadratic classifiers
- Kernel estimation
 - k-nearest neighbor
- Boosting (meta-algorithm)
- Decision trees
 - Random forests
- Neural networks

- Gene Expression Programming
- Bayesian networks
- Hidden Markov models

Ακολουθεί πίνακας που παρουσιάζει τα πιο γνωστά και αποτελεσματικά ελεύθερα λογισμικά καθώς και τις βιβλιοθήκες που χρησιμοποιούνται για διαδικασίες υπολογιστικής μάθησης.

Ελεύθερα λογισμικά			
Ονομασία	Κύρια χαρακτηριστικά	Αλγόριθμοι Ταξινόμησης	Ιστότοπος
Apache Mahout	java cross platform Apache 2.0 licence	Logistic Regression Naive Bayes Random Forest Hidden Markov Models Multilayer Perceptron	http://mahout.apache.org
mlpy	Python, C, C++ Cross platform GPL licence	Linear Discriminant Analysis Logistic Regression SVM KNN Classification Tree	http://mlpy.sourceforge.net
OpenCV	C, C++ Cross platform BSD licence	Boosting Decision tree KNN Naive Bayes classifier ANN SVM etc	http://opencv.org
R	C, Fortran, R language Cross platform GNU licence	Neural Networks Random Forests Recursive Partitioning Boosting SVM Bayesian methods etc	http://www.r-project.org
Scikit-learn	Python, C, C++ Cross platform BSD Licence	Svm Generalized Linear Models Stochastic Gradient Descent KNN Decision Tree Naive Bayes etc	http://scikit-learn.org/stable/
Weka	Java Cross platform GNU licence	AdaBoost Bayesian methods SVM Naive Bayes Random Forest ANN etc	http://www.cs.waikato.ac.nz/ml/weka/

Πίνακας 4.4: Ελεύθερα λογισμικά για διαδικασίες data mining

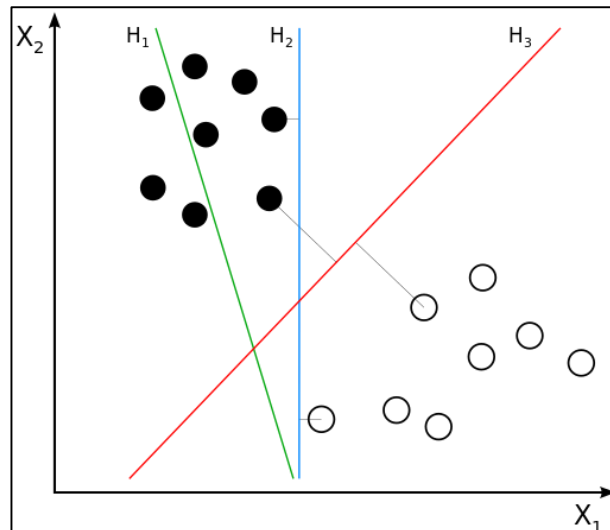
4.5 Μέθοδοι Ταξινόμησης και Μηχανική Μάθηση

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται διάφοροι αλγόριθμοι ταξινόμησης που βασίζονται σε διαδικασίες μηχανικής μάθησης και χρησιμοποιούνται ευρέως σε εφαρμογές που απαιτούν χρήση μεγάλων και ανομοιογενών δεδομένων. Πιο συγκεκριμένα, περιγράφονται οι εξής μέθοδοι: SVM, Decision tree, K-nearest neighbors και παρουσιάζονται παράλληλα παραδείγματα εφαρμογών τους, χρησιμοποιώντας αλγορίθμους από το orfeo toolbox και τη βιβλιοθήκη OpenCV.

4.5.1 SVM

Οι μηχανές υποστήριξης διανυσμάτων (Support Vector Machines) είναι επιβλεπόμενα μοντέλα μάθησης που συνδυάζονται με αλγορίθμους εκμάθησης, με στόχο την ανάλυση δεδομένων και την αναγνώριση προτύπων για διαδικασίες ταξινόμησης και κατάτμησης (regression). Παρέχοντας ένα σύνολο δεδομένων εκπαίδευσης, το καθένα από αυτά διαχωρίζεται σε μία από τις δύο κατηγορίες και κάθε φορά ο αλγόριθμος εκπαίδευσης, SVM, χτίζει ένα μοντέλο που διαχωρίζει τα νέα δεδομένα στη μία ή στην άλλη κατηγορία δημιουργώντας ένα μη πιθανοτικό, γραμμικό και δυαδικό μοντέλο. Το μοντέλο SVM, πρακτικά, είναι μια αναπαράσταση των δεδομένων στο χώρο, τα οποία απεικονίζονται ως κατηγορίες διαχωρισμένες από ένα σαφές όριο, το οποίο χαρακτηρίζεται από συγκεκριμένο εύρος κάθε φορά. Τα νέα δεδομένα αντιστοιχίζονται στο συγκεκριμένο διάστημα-σύνολο που ανήκουν με βάση την πλευρά του ορίου που βρίσκονται. Εκτός από τη γραμμική μορφή ταξινόμησης, τα μοντέλα SVM μπορούν να εκτελέσουν αποτελεσματικά μη γραμμικές ταξινομήσεις χρησιμοποιώντας εξειδικευμένα τεχνάσματα του πυρήνα τους (kernel) και αντιμετωπίζοντας πολύπλοκα προβλήματα. [Τζώτσος, 2014]

Οι μηχανές υποστήριξης διανυσμάτων κατασκευάζουν ένα υπερεπίπεδο (hyperplane) ή ένα σύνολο υπερεπιπέδων σε έναν υψηλό ή άπειρων διαστάσεων χώρο, το οποίο μπορεί να χρησιμοποιηθεί για ταξινόμηση, κατάτμηση ή άλλες εφαρμογές. Ένας καλός διαχωρισμός επιτυγχάνεται από το υπερεπίπεδο που έχει τη μεγαλύτερη δυνατή απόσταση από το πλησιέστερο σημείο των δεδομένων εκπαίδευσης, καθώς όσο μεγαλύτερο είναι το περιθώριο τόσο μικρότερη είναι η πιθανότητα λάθους από τον ταξινομητή. [Vapnik & Cortes, 1995]



Εικόνα 4.6: Παράδειγμα ταξινομητή SVM
http://en.wikipedia.org/wiki/Support_vector_machine

Στο παραπάνω παράδειγμα γίνεται αντιληπτό πως το όριο H1 δεν διαχωρίζει τις κλάσεις. Αντίθετα το όριο H2 διαχωρίζει τις κλάσεις παρέχοντας ωστόσο μικρό περιθώριο, ενώ το όριο H3 διαχωρίζει τις κλάσεις με το μέγιστο δυνατό περιθώριο.

Linear SVM

Έστω κάποια δεδομένα εκπαίδευσης D και μία σειρά n σημείων της μορφής:

$$D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, \quad y_i \in \{-1, 1\}\}_{i=1}^n$$

όπου τα y_i είναι 1 ή -1, υποδεικνύοντας την κλάση στην οποία το σημείο x_i ανήκει.

Κάθε σημείο x_i είναι ένα διάνυσμα p -διαστάσεων. Στόχος είναι να βρεθεί το μέγιστο περιθώριο του υπερεπιπέδου που διαχωρίζει τα σημεία με τιμή $y_i = 1$ από αυτά που έχουν τιμή $y_i = -1$. Κάθε υπερεπίπεδο μπορεί να γραφτεί ως το σύνολο των σημείων x που ικανοποιούν την παρακάτω μαθηματική σχέση:

$$w \cdot x - b = 0$$

όπου w το «κανονικό» διάνυσμα (normal vector) του υπερεπιπέδου.

Η παράμετρος $\frac{b}{\|w\|}$ δηλώνει την απόκλιση (offset) του υπερεπιπέδου από την αρχή και κατά μήκος του «κανονικού» διανύσματος w .

Εάν τα δεδομένα εκπαίδευσης είναι γραμμικώς διακριτά, είναι δυνατή η επιλογή δύο υπερεπιπέδων με τέτοιο τρόπο ώστε τα δεδομένα να διαχωρίζονται και να μην υπάρχουν σημεία μεταξύ τους. Στη συνέχεια, γίνεται προσπάθεια μεγιστοποίησης

της απόστασής τους. Η περιοχή που οριοθετείται από αυτά ονομάζεται περιθώριο (margin). Τα παραπάνω υπερεπίπεδα μπορούν να περιγραφούν από τις εξής εξισώσεις:

$$w \cdot x - b = 1$$

και

$$w \cdot x - b = -1$$

Με εφαρμογή γεωμετρίας βρίσκεται η απόσταση μεταξύ των δύο υπερεπιπέδων (hyperplanes) που είναι $\frac{2}{\|w\|}$ και στόχος είναι η ελαχιστοποίηση του όρου $\|w\|$. Ωστόσο, επειδή πρέπει να τηρείται και ο περιορισμός των δεδομένων, δηλαδή να μην βρεθούν αυτά στο περιθώριο (margin), προστίθεται ένας παραπάνω περιορισμός για κάθε σημείο i :

$$w \cdot x_i - b \geq 1, \quad \text{για κάθε } x_i \text{ της πρώτης κλάσης}$$

ή

$$w \cdot x_i - b \leq -1, \quad \text{για κάθε } x_i \text{ της δεύτερης κλάσης}$$

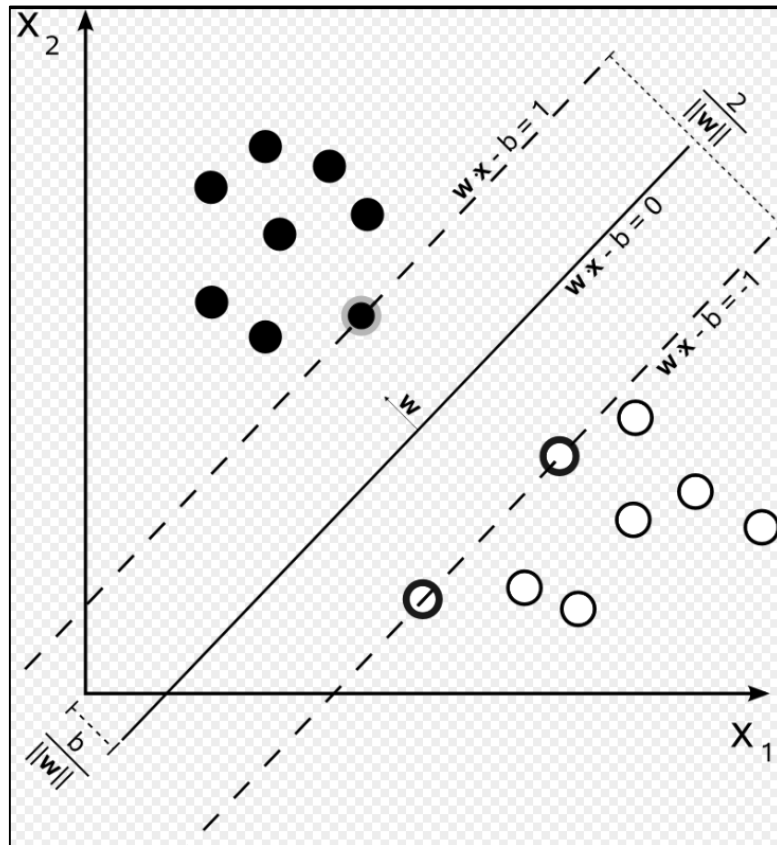
Οι παραπάνω δύο περιορισμοί μπορούν να γραφτούν και ως εξής:

$$y_i(w \cdot x_i - b) \geq 1, \quad 1 \leq i \leq n$$

Χρησιμοποιώντας τις παραπάνω σχέσεις καταστρώνεται το πρόβλημα βελτιστοποίησης:

Ελαχιστοποίηση w, b υπό τους περιορισμούς $y_i(w \cdot x_i - b) \geq 1, \quad 1 \leq i \leq n$

[http://en.wikipedia.org/wiki/Support_vector_machine, 2014]



Εικόνα 4.7: Πρόβλημα βελτιστοποίησης περιθωρίου σε γραμμικό ταξινομητή SVM

http://en.wikipedia.org/wiki/Support_vector_machine

Για την επίλυση του προβλήματος της βελτιστοποίησης που περιγράφηκε παραπάνω έχουν προταθεί διάφορες λύσεις, ωστόσο στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας περιγράφονται δύο κλασσικές μέθοδοι βελτιστοποίησης που έχουν χρησιμοποιηθεί για το σκοπό αυτό.

▪ Primal form

Η δυσκολία του συγκεκριμένου προβλήματος βελτιστοποίησης έγκειται στο γεγονός πως η επίλυση του στηρίζεται στον όρο $\|w\|$, ο οποίος απαιτεί τη χρήση τετραγωνικής ρίζας. Για το συγκεκριμένο λόγο πραγματοποιείται αντικατάσταση της σχέσης $\|w\|$ με τη σχέση $\frac{1}{2}\|w\|^2$ για τη διευκόλυνση των μαθηματικών πράξεων, χωρίς όμως να αλλάζει το αποτέλεσμα της επίλυσης. Η επίλυση του προβλήματος της βελτιστοποίησης ανάγεται σε ένα πρόβλημα βελτιστοποίησης τετραγωνικού προγραμματισμού (quadratic programming optimization). (Nocedal, Jorge, Wright, & Stephen, 2006)

Πιο συγκεκριμένα ισχύουν οι εξής σχέσεις:

$$\operatorname{argmin}_{(w,b)} \frac{1}{2} \|w\|^2, \text{ υπό τους περιορισμούς } y_i(w \cdot x_i - b) \geq 1, \quad 1 \leq i \leq n$$

Εντάσσοντας τους πολλαπλασιαστές του Lagrange α , το προηγούμενο πρόβλημα περιορισμών μπορεί να εκφραστεί ως εξής:

(http://en.wikipedia.org/wiki/Lagrange_multiplier)

$$\operatorname{argmin}_{(w,b)} \max_{a \geq 0} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i [y_i(w \cdot x_i - b) - 1] \right\}$$

Ο μετασχηματισμός του αρχικού προβλήματος βελτιστοποίησης στην παραπάνω σχέση οδηγεί στην επίλυσή του με τεχνικές τετραγωνικού προγραμματισμού, χρησιμοποιώντας γραμμικούς συνδυασμούς μεταξύ των διανυσμάτων εκπαίδευσης.

$$w = \sum_{i=1}^n a_i y_i x_i$$

Μόνο ορισμένα a_i θα είναι μεγαλύτερα του μηδενός. Τα αντίστοιχα x_i είναι διανύσματα υποστήριξης, τα οποία βρίσκονται στο περιθώριο και καλύπτουν τη σχέση $y_i(w \cdot x_i - b) = 1$. Από το προηγούμενο εξάγεται το συμπέρασμα πως τα συγκεκριμένα διανύσματα θα καλύπτουν επίσης τη σχέση:

$$w \cdot x_i - b = \frac{1}{y_i} = y_i \leftrightarrow b = w \cdot x_i - y_i$$

μέσω της οποίας είναι δυνατός ο προσδιορισμός της απόκλισης (offset) b , χρησιμοποιώντας το μέσο όρο όλων των διανυσμάτων υποστήριξης από τη σχέση:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (w \cdot x_i - y_i)$$

(http://en.wikipedia.org/wiki/Support_vector_machine)

▪ Dual form

Γράφοντας τον κανόνα ταξινόμησης στη διπλή μορφή του γίνεται αντιληπτό πως το μέγιστο περιθώριο του υπερεπιπέδου και συνεπώς της διαδικασίας της ταξινόμησης είναι μία συνάρτηση των διανυσμάτων υποστήριξης και του υποσυνόλου των δεδομένων εκπαίδευσης που βρίσκονται στο περιθώριο.

Χρησιμοποιώντας τη σχέση $\|w\|^2 = w \cdot w$ και αντικαθιστώντας την τιμή του w με $\sum_{i=1}^n (w \cdot x_i - y_i)$ γίνεται αντιληπτό πως οδηγούμαστε στο παρακάτω πρόβλημα βελτιστοποίησης:

Ελαχιστοποίηση του a_i : $\tilde{L}(\alpha) = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i,j} a_i a_j y_i y_j k(x_i, x_j)$

υπό τους περιορισμούς: $0 \leq a_i \leq C$, $1 \leq i \leq n$ και $\sum_{i=1}^n a_i y_i = 0$

[Vapnik & Cortes, 1995]

Nonlinear SVM

Η δημιουργία μη γραμμικών ταξινομητών πραγματοποιείται με την εφαρμογή τεχνάσματος στον πυρήνα (kernel) που στοχεύει στην μεγιστοποίηση του περιθωρίου στα υπερεπίπεδα. (Aizerman, Mark A. Braverman, Emmanuel M, Rozonoer, 1964) Ο αλγόριθμος που προκύπτει είναι τυπικά ο ίδιος, εκτός από το ότι κάθε εσωτερικό γινόμενο αντικαθίσταται από μία μη γραμμική συνάρτηση του πυρήνα. Αυτό επιτρέπει στον αλγόριθμο να «χωρέσει» το μέγιστο περιθώριο του υπερεπιπέδου σε ένα μετασχηματισμένο χώρο χαρακτηριστικών. [Boser, Guyon, & Vapnik, 1992]

Μερικοί από τους πιο συνηθισμένους πυρήνες είναι οι εξής:

- Polynomial (ομογενείς): $k(x_i, x_j) = (x_i \cdot x_j)^d$
- Polynomial (ανομοιογενείς): $k(x_i, x_j) = (x_i \cdot x_j + 1)^d$
- Gaussian radial basis function: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$
- Hyperbolic tangent: $k(x_i, x_j) = \tanh(k x_i \cdot x_j + c)$, $k > 0$ $c < 0$

Multiclass SVM

Η μηχανή υποστήριξης διανυσμάτων σε πολλαπλές κλάσεις έχει ως στόχο την εκχώρηση ετικετών (labels) σε περιπτώσεις (instances), όπου οι ετικέτες έχουν προέλθει από ένα πεπερασμένο σύνολο στοιχείων. Η κυρίαρχη προσέγγιση είναι ο μετασχηματισμός του εννιαίου προβλήματος πολλαπλών κλάσεων σε πολλαπλά δυαδικά προβλήματα ταξινόμησης. [Duan & Keerthi, 2005]

Ακολουθούν τρεις από τις πιο συνηθισμένες μεθόδους για την αντιμετώπιση του παραπάνω προβλήματος:

- Δημιουργία δυαδικών ταξινομητών που διακρίνονται μεταξύ τους σε δύο κατηγορίες: 1) κάθε ετικέτα συγκρίνεται με τις άλλες (one-versus-one) ή 2) ζεύγη ετικετών συγκρίνονται μεταξύ τους (one-versus-one). Η ταξινόμηση στην πρώτη κατηγορία στηρίζεται στη λογική winner-takes-all, στην οποία ο ταξινομητής με την υψηλότερη συνάρτηση εξόδου καταχωρεί την κατηγορία, ενώ στη δεύτερη η ταξινόμηση στηρίζεται στη στρατηγική max-win voting, όπου η κατηγορία με τους περισσότερους «ψήφους» καταχωρεί την τιμή της σε κάθε περίπτωση. [Hsu, Chih-Wei, & Lin, 2002]

- Directed acyclic graph [Platt, John, Cristianini, & Shawe-Taylor, 2000]
- Error-correcting output coded [Dietterich, G, Bakiri, & Ghulum, 1995]

Ακολουθεί πίνακας που παρουσιάζει τα κύρια πλεονεκτήματα και μειονεκτήματα των μηχανών υποστήριξης διανυσμάτων:

Πλεονεκτήματα	Μειονεκτήματα
Αποτελεσματικές σε πολυδιάστατους χώρους	Εάν ο αριθμός των χαρακτηριστικών είναι πολύ μεγαλύτερος από τον αριθμό των δειγμάτων δεν δίνουν καλά αποτελέσματα
Δεν απαιτούν μεγάλη υπολογιστική ισχύ	
Ευέλικτες ως προς τον προσδιορισμό των πυρήνων	Δεν παρέχουν άμεσα εκτιμήσεις πιθανοτήτων, απαιτούν αντεπικύρωση (cross-validation)
Αποτελεσματικές και με σχετικά μικρό αριθμό δειγμάτων	

Πίνακας 4.5: Πλεονεκτήματα-Μειονεκτήματα SVM
<http://scikit-learn.org/stable/modules/svm.html>

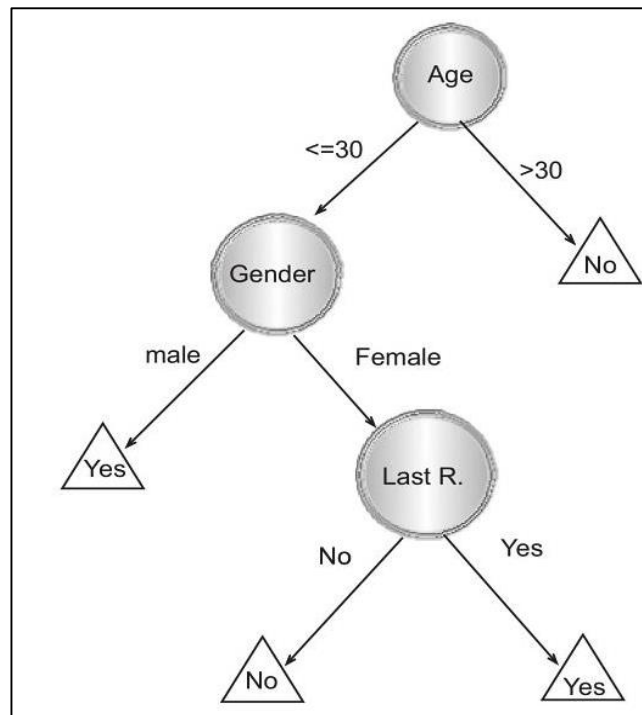
4.5.2 Decision tree

Στις τεχνικές εξόρυξης δεδομένων (data mining) το δέντρο αποφάσεων (decision tree) είναι ένα μοντέλο πρόβλεψης τόσο για ταξινόμηση όσο και για κατάτμηση. Πρόκειται για ένα ιεραρχικό μοντέλο αποφάσεων και συνεπειών που καλείται να βρει τη βέλτιστη στρατηγική για την επίτευξη του στόχου. Όταν ένα δέντρο αποφάσεων χρησιμοποιείται για διαδικασίες ταξινόμησης καλείται δέντρο ταξινόμησης (classification tree), ενώ όταν χρησιμοποιείται για διαδικασίες κατάτμησης καλείται δέντρο κατάτμησης (regression tree). Στη συνέχεια του κεφαλαίου, παρουσιάζονται οι ιδιότητες και τα χαρακτηριστικά των δέντρων ταξινόμησης, τα οποία χρησιμοποιούνται σε σημαντικό βαθμό σε τομείς που ασχολούνται με τα οικονομικά, το marketing, τις επιστήμες και την ιατρική. [Rokach, 2008]

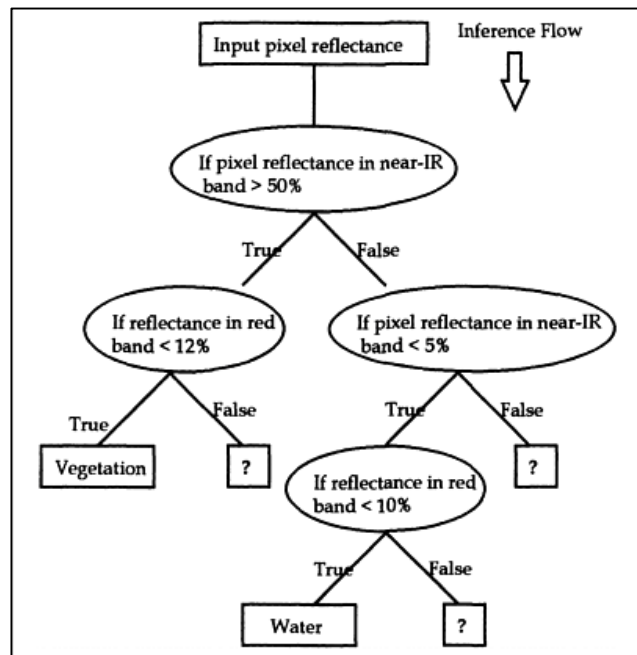
Το δέντρο απόφασης είναι ένας ταξινομητής που εκφράζεται ως μια αναδρομική διχοτόμηση του χώρου. Αποτελείται από κόμβους (nodes) που σχηματίζουν ένα ριζωμένο δέντρο (rooted tree), δηλαδή ένα δέντρο με κατεύθυνση και με έναν κόμβο που δεν έχει εισερχόμενες ακμές. Ο κόμβος με εξερχόμενες ακμές ονομάζεται «εσωτερικός» ή «δοκιμαστικός» κόμβος, ενώ όλοι οι άλλοι κόμβοι ονομάζονται «φύλλα» (leaves). Κάθε φύλλο αντιπροσωπεύεται είτε από μια κλάση με το πιο αντιπροσωπευτικό χαρακτηριστικό, είτε από ένα διάνυσμα πιθανότητας που μεταφράζεται ως η πιθανότητα να έχει το φύλλο μια συγκεκριμένη κλάση. Σε κάθε εσωτερικό κόμβο υπάρχει η δυνατότητα να δημιουργηθούν περισσότερα από δύο κλαδιά (branches). Κάθε κομβός αντιπροσωπεύεται από ένα αποκλειστικό χαρακτηριστικό, ενώ κάθε κλαδί αντιπροσωπεύεται από ένα εύρος τιμών. [Rokach, 2008]

Κάθε αντικείμενο ταξινομείται ξεκινώντας από την ρίζα του δέντρου μέχρι κάποιο φύλλο του, ανάλογα με τα αποτελέσματα των ελέγχων στη διαδρομή που ακολουθεί και σύμφωνα με τα χαρακτηριστικά του. Πιο συγκεκριμένα, η διαδικασία ξεκινά από την ρίζα του δέντρου, ελέγχεται στη συνέχεια ποιο κλαδί είναι πιο αντιπροσωπευτικό ως προς τα χαρακτηριστικά του αντικειμένου και τέλος ελέγχεται ο κόμβος στον οποίο ανήκει το συγκεκριμένο κλαδί. Η διαδικασία επαναλαμβάνεται μέχρι να υπάρξει αντιστοίχιση – ταξινόμηση του αντικειμένου σε κάποιο φύλλο. [Rokach, 2008] Στις περισσότερες περιπτώσεις επιλέγεται ένα όχι ιδιαίτερα πολύπλοκο δέντρο ώστε να είναι κατανοητό από τους μελετητές, ενώ λαμβάνεται υπόψη το γεγονός ότι η πολυπλοκότητα στα δέντρα απόφασης έχει επίδραση στην ακρίβεια της ταξινόμησης. [Breinman et al., 1984]

Ακολουθούν δύο διαγραμματικά παραδείγματα εφαρμογής του δέντρου απόφασης:



Εικόνα 4.8: Δέντρο απόφασης για την εύρεση συγκεκριμένων επιθέτων
Data mining with decision trees: Theory and Application, Lior Rokach, 2008



Εικόνα 4.9: Δέντρο απόφασης για την ταξινόμηση εικονοστοιχείων
Classification methods for remotely data, second edition, Tso B., Mather P.

Η φάση εξάπλωσης του δέντρου συνεχίζεται μέχρις ότου καλυφθούν κάποια κριτήρια τερματισμού. Οι συνθηκές που οδηγούν σε τερματισμό είναι οι εξής: [Rokach, 2008]

- Εάν όλα τα στοιχεία των δεδομένων εκπαίδευσης έχουν μια μοναδική τιμή γ
- Εάν έχει εκπληρωθεί η συνθήκη του μέγιστου βάθους του δέντρου
- Εάν ο αριθμός των περιπτώσεων στον τερματικό κόμβο είναι μικρότερος από τον ελάχιστο αριθμό περιπτώσεων που έχει τεθεί για τους κόμβους
- Εάν ο κόμβος έχει διαχωριστεί και ο αριθμός των περιπτώσεων εμφάνισης είναι μικρότερος από αυτόν που έχει οριστεί
- Εάν ο αριθμός βέλτιστων διαχωρισμών είναι μεγαλύτερος από το συγκεκριμένο κατώφλι που έχει τεθεί

Ακολουθεί παράδειγμα ψευδοκώδικα του αλγόριθμου για την επέκταση του δέντρου απόφασης:

```

TreeGrowing (S,A,y,SplitCriterion,StoppingCriterion)
Where:
S - Training Set
A - Input Feature Set
y - Target Feature
SplitCriterion - the method for evaluating a certain split
StoppingCriterion - the criteria to stop the growing process
Create a new tree T with a single root node.
IF StoppingCriterion(S) THEN
    Mark T as a leaf with the most
    common value of y in S as a label.
ELSE
     $\forall a_i \in A$  find a that obtain the best SplitCriterion( $a_i, S$ ).
    Label t with a
    FOR each outcome  $v_i$  of a:
        Set Subtreei = TreeGrowing ( $\sigma_{a=v_i} S, A, y$ ).
        Connect the root node of  $t_T$  to Subtreei with
        an edge that is labelled as  $v_i$ 
    END FOR
END IF
RETURN TreePruning (S,T,y)
TreePruning (S,T,y)
Where:
S - Training Set
y - Target Feature
T - The tree to be pruned
DO
    Select a node t in T such that pruning it
    maximally improve some evaluation criteria
    IF  $t \neq \emptyset$  THEN  $T = \text{pruned}(T, t)$ 
UNTIL  $t = \emptyset$ 
RETURN T

```

Εικόνα 4.10: Ψευδοκώδικας αλγορίθμου τερματισμού επέκτασης δέντρου απόφασης
Data mining with decision trees: Theory and Application, Lior Rokach, 2008

Ακολουθεί πίνακας που παρουσιάζει τα κύρια πλεονεκτήματα και μειονεκτήματα των δέντρων απόφασης:

Πλεονεκτήματα	Μειονεκτήματα
Εύκολα στην κατανόηση και δυνατότητα αναπαράστασής τους	Προβλήματα ακρίβειας σε σύνθετες μορφές δέντρων
Δεν απαιτούν μεγάλη προεργασία δεδομένων	Απαιτείται εμπειρία από το μελετητή για να δημιουργηθούν σωστές συνθήκες
Δυνατότητα ενσωμάτωσης ποσοτικών και ποιοτικών κριτηρίων	Τα αποτελέσματα επηρεάζονται σε σημαντικό βαθμό από τις συνθήκες τερματισμού
Δυνατότητα αξιολόγησής τους μέσα από στατιστικούς ελέγχους	Απαιτούν αρκετά μεγάλο όγκο δεδομένων εκπαίδευσης
Δεν απαιτούν μεγάλη υπολογιστική ισχύ	

Πίνακας 4.6: Πλεονεκτήματα-Μειονεκτήματα decision tree
<http://scikit-learn.org/stable/modules/tree.html>

4.5.3 K-nearest neighbors

Στην αναγνώριση προτύπων, ο αλγόριθμος k-πλησιέστερων γειτόνων (k-nearest neighbors ή KNN) είναι μία μη παραμετρική μέθοδος που χρησιμοποιείται για ταξινόμηση και κατάτμηση. Και στις δύο περιπτώσεις ο αλγόριθμος στηρίζεται στα

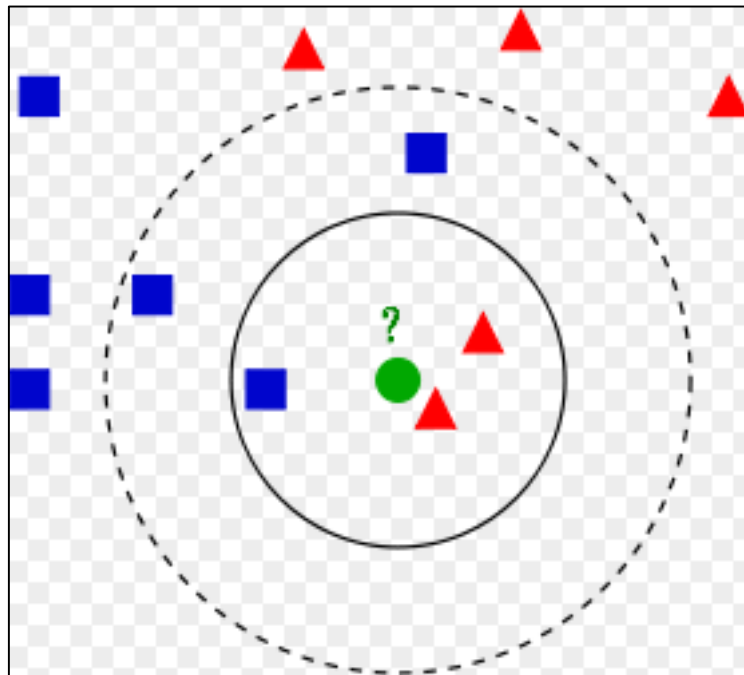
k-κοντινά δεδομένα εκπαίδευσης στο χώρο των αντικειμένων. Όσον αφορά την εφαρμογή του αλγορίθμου σε διαδικασίες ταξινόμησης, η έξοδος είναι η κατηγοριοποίηση των δεδομένων σε κλάσεις με βάση την πλειοψηφία των γειτόνων (majority voting). Ο αριθμός των γειτόνων k είναι ένας θετικός ακέραιος αριθμός, όπου αν $k=1$ τότε το αντικείμενο απλά αποδίδεται στην κατηγορία που ανήκει ο πλησιέστερος γείτονας. Ο αλγόριθμος KNN ανήκει στην κατηγορία των instance-based learning algorithms ή αλλιώς lazy learning algorithms καθώς η συνάρτηση απόδοσης τιμής και οι υπολογισμοί είναι κατά προσέγγιση σε τοπικό επίπεδο. Σε σύγκριση με άλλους αλγορίθμους υπολογιστικής μάθησης ο KNN είναι ο πιο απλός. [Altman, 1992]

Επιπλέον, ο αλγόριθμος KNN δεν χρησιμοποιεί τα δεδομένα εκπαίδευσης για να κάνει οποιαδήποτε γενίκευση, οπότε δεν υπάρχει ρητή φάση εκπαίδευσης ή αυτή είναι ελάχιστη. Με τη μη γενίκευση των δεδομένων εκπαίδευσης ο αλγόριθμος κρατά όλα τα δεδομένα για εκπαίδευση σε όλες τις φάσεις δοκιμών σε αντίθεση με άλλους αλγορίθμους εκπαίδευσης, όπως είναι ο SVM, όπου υπάρχει η δυνατότητα απόρριψης δεδομένων. Κατά συνέπεια, ο αλγόριθμος KNN είναι πιο δαπανηρός από άποψη υπολογιστικής ισχύος καθώς αποθηκεύει και χρησιμοποιεί το σύνολο των δεδομένων για τη λήψη απόφασης. [Thirumuruganathan, 2010]

Όπως αναφέρθηκε, ο αλγόριθμος KNN είναι μη παραμετρικός και το μόνο που απαιτείται είναι η επιλογή του αριθμού των γειτόνων k . Η καλύτερη επιλογή του k εξαρτάται από τα δεδομένα και σε γενικές γραμμές έχει παρατηρηθεί πως όσο μεγαλύτερη είναι η τιμή του k τόσο περισσότερο μειώνεται ο θόρυβος στην ταξινόμηση κάνοντας παράλληλα τα όρια των τάξεων λιγότερο διακριτά. [Everitt, Landau, Leese, & Stahl, 2011]

Για τη βέλτιστη επιλογή του αριθμού k πραγματοποιούνται διάφοροι αλγόριθμοι βελτιστοποίησης (π.χ. υπερπαραμετρική βελτιστοποίηση) για κάθε διαφορετική εφαρμογή. Στην ειδική περίπτωση που ο αριθμός k επιλέγεται να ισούται με τη μονάδα τότε ο αλγόριθμος ονομάζεται πλησιέστερου γείτονα.

Ακολουθεί ένα παράδειγμα του αλγορίθμου KNN:



Εικόνα 4.11: Παράδειγμα εφαρμογής αλγορίθμου KNN
KnnClassification, Antti Ajaniki, 2007

Στο παραπάνω παράδειγμα γίνεται αντιληπτό πως ο πράσινος κύκλος πρέπει να ταξινομηθεί είτε στην κατηγορία μπλε τετράγωνο είτε στην κατηγορία κόκκινο τρίγωνο. Αν επιλεγεί ο αριθμός γειτόνων να είναι τρεις, $k=3$ (με τη συνεχόμενη γραμμή), τότε θα επιλεγεί η κατηγορία κόκκινο τρίγωνο αφού μέσα στον κύκλο υπάρχουν 2 κόκκινα τρίγωνα και 1 μπλε τετράγωνο. Αντίθετα, αν επιλεγεί ο αριθμός γειτόνων να είναι πέντε, $k=5$ (με τη διακεκομμένη γραμμή) τότε θα επιλεγεί η κατηγορία μπλε τετράγωνο αφού στο όριο υπάρχουν 3 μπλε τετράγωνα και 2 κόκκινα τρίγωνα.

Ακολουθεί ο ψευδοκώδικας του αλγορίθμου KNN:

```

TRAIN-KNN( $\mathbf{C}, \mathbf{ID}$ )
1  $\mathbf{ID}' \leftarrow \text{PREPROCESS}(\mathbf{ID})$ 
2  $k \leftarrow \text{SELECT-K}(\mathbf{C}, \mathbf{ID}')$ 
3 return  $\mathbf{ID}', k$ 

APPLY-KNN( $\mathbf{C}, \mathbf{ID}', k, d$ )
1  $S_k \leftarrow \text{COMPUTENEARESTNEIGHBORS}(\mathbf{ID}', k, d)$ 
2 for each  $c_j \in \mathbf{C}$ 
3 do  $p_j \leftarrow |S_k \cap c_j|/k$ 
4 return  $\arg \max_j p_j$ 
  
```

Εικόνα 4.12: Ψευδοκώδικας αλγορίθμου KNN
Introduction to information retrieval, lecture by Christopher D. Manning et al, 2008

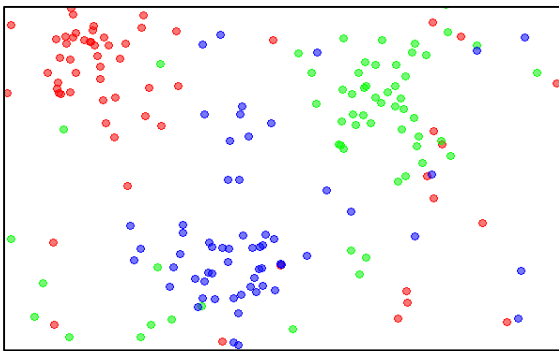
Όσον αφορά στον αλγόριθμο KNN, έχει προταθεί μία βελτίωσή του με ονομασία CNN (condensed nearest neighbor), η οποία σχεδιάστηκε για να μειώσει το σύνολο

των δεδομένων για τον αλγόριθμο ταξινόμησης KNN. Ο αλγόριθμος επιλέγει το σύνολο των προτύπων U από τα δεδομένα εκπαίδευσης, έτσι ώστε ο αλγόριθμος πλησιέστερου γείτονα με $k=1$ να μπορεί να ταξινομή το σύνολο των δεδομένων με την ακρίβεια που θα έκανε ο 1NN σε μικρότερα παραδείγματα. Πιο συγκεκριμένα, ακολουθείται η εξής διαδικασία: [Hart, 1968]

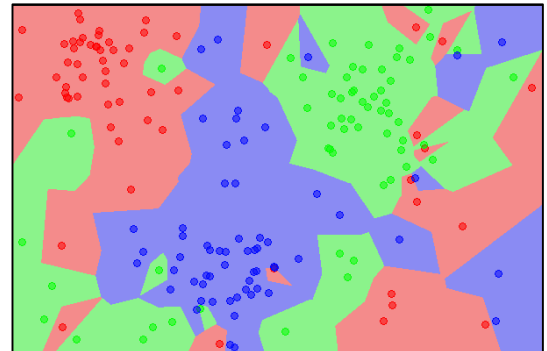
Έστω τα δεδομένα εκπαίδευσης X :

- Σάρωση όλων των στοιχείων X , ψάχνοντας το στοιχείο x του οποίου το κοντινότερο πρότυπο από το U έχει διαφορετική ετικέτα από το x
- Αφαίρεση του x από το X και ένταξη του στο U
- Επανάληψη της διαδικασίας μέχρις ότου κανένα πρότυπο να μην εντάσσεται στο U
- Χρήση του U αντί του X για ταξινόμηση

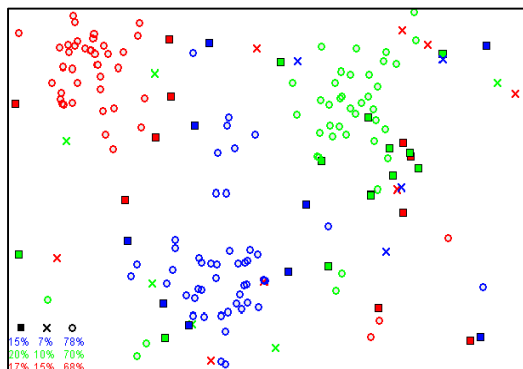
Ακολουθεί παράδειγμα που παρουσιάζει τα δεδομένα εκπαίδευσης και το αποτέλεσμα της ταξινόμησης χωρίς τη χρήση του αλγορίθμου CNN καθώς και αυτά με τη χρήση του αλγορίθμου CNN.



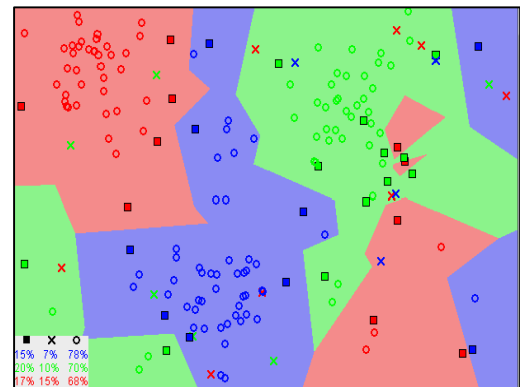
Εικόνα 4.13: Δεδομένα εκπαίδευσης χωρίς την εφαρμογή του CNN
http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm



Εικόνα 4.14: 1NN ταξινόμηση χωρίς εφαρμογή του CNN
http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm



Εικόνα 4.15: Δεδομένα εκπαίδευσης με εφαρμογή του CNN
http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm



Εικόνα 4.16: 1NN ταξινόμηση με εφαρμογή του CNN
http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Ακολουθεί πίνακας που παρουσιάζει τα κύρια πλεονεκτήματα και μειονεκτήματα των κ-πλησιέστερων γειτόνων:

Πλεονεκτήματα	Μειονεκτήματα
Εύκολα στην κατανόηση και εκσφαλμάτωσή τους	Απαιτούν μεγάλη υπολογιστική ισχύ και είναι αρκετά αργά
Αντιμετώπιση θορύβου στις ταξινομήσεις	Ευαίσθητα στην επιλογή γειτόνων k
Μη παραμετρικά	Δεν αντιμετωπίζουν αποτελεσματικά σύνθετα πρότυπα
Ευκολία αντιμετώπισης προβλημάτων με πολλά δεδομένα εκπαίδευσης	

Πίνακας 4.7: Πλεονεκτήματα-Μειονεκτήματα decision tree

[Cunningham & Delany, 2007]

4.5.4 Στατιστικά μοντέλα

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται κάποιοι δείκτες που χρησιμοποιούνται προκειμένου να αξιολογηθούν τα στατιστικά μοντέλα εκμάθησης που εφαρμόζονται στα πλαίσια των διαδικασιών εξόρυξης δεδομένων. Πιο συγκεκριμένα, παρουσιάζεται ο δείκτης της ακρίβειας (precision), ο αριθμός εύστοχων αποκρίσεων (recall), ο δείκτης f1 score, ο δείκτης accuracy και ο δείκτης kappa.

Πριν δοθούν οι ορισμοί των παραπάνω δεικτών είναι αναγκαίο να αναφερθούν τέσσερις έννοιες που είναι απαραίτητες για την πλήρη κατανόηση των δεικτών. Συγκεκριμένα, εξηγούνται οι έννοιες true positives, true negatives, false positives, false negatives.

Αρχικά, οι έννοιες positive (θετικό) και negative (αρνητικό) αποδίδουν την πρόβλεψη του ταξινομητή (ή προσδοκία) ενώ οι έννοιες αληθές (true) και ψευδές (false) αναφέρονται στο αν η παραπάνω πρόβλεψη ανταποκρίνεται σε εξωτερική κρίση (ή παρατήρηση). Συνδυάζοντας τις παραπάνω έννοιες προκύπτει πως με την έννοια true positives (TP) ορίζονται τα αντικείμενα που ταξινομήθηκαν σωστά στην τάξη, με την έννοια true negative (TN) ορίζονται τα αντικείμενα που δεν συμπεριλήφθηκαν σωστά στην τάξη, με την έννοια false positive (FP) ορίζονται τα αντικείμενα που ταξινομήθηκαν στην τάξη αλλά δεν έπρεπε, ενώ με την έννοια false negative (FN) ορίζονται τα αντικείμενα που δεν συμπεριλήφθηκαν στην τάξη ενώ έπρεπε. [Powers & David, 2011]

Precision

Σε εργασίες ταξινόμησης ο δείκτης precision (ακρίβεια) για μια συγκεκριμένη τάξη είναι ο αριθμός των αληθώς θετικών (true positives) αντικειμένων, δηλαδή ο αριθμός των αντικειμένων που επισημάνθηκαν σωστά ότι ανήκουν σε μία τάξη, διαιρούμενος με το συνολικό αριθμό των στοιχείων που επισημάνθηκαν ότι

ανήκουν στη θετική τάξη, δηλαδή το άθροισμα των αληθών και ψευδών θετικών αντικειμένων. [Powers & David, 2011]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall

Ο αριθμός εύστοχων αποκρίσεων (recall) ορίζεται ως ο αριθμός των αληθώς θετικών αντικειμένων διαιρούμενος με το συνολικό αριθμό των στοιχείων που πραγματικά ανήκουν στη θετική τάξη, δηλαδή το άθροισμα των πραγματικά αληθώς και ψευδώς αρνητικών αντικειμένων που έπρεπε να ανήκουν στη συγκεκριμένη τάξη αλλά δεν επισυμάνθηκαν. [Powers & David, 2011]

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F1-score

Πρόκειται για ένα μέτρο που συνδυάζει την ακρίβεια με την ευστοχία μέσω ενός αρμονικού μέσου σταθμίζοντας τις δύο έννοιες ομοιόμορφα. [Powers & David, 2011]

Πιο συγκεκριμένα:

$$F - \text{score} = 2 * \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Accuracy

Η έννοια της ακρίβειας (accuracy) εκφράζει το πηλίκο του αθροίσματος των αληθών αντικειμένων (θετικών και αρνητικών) με το σύνολο όλων των θετικών και αρνητικών αντικειμένων. Πιο συγκεκριμένα:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Kappa

Η στατιστική τιμή kappa είναι ένα μέτρο διαφοράς μεταξύ της πραγματικής συμφωνίας μεταξύ των επίγειων δεδομένων αναφοράς και ενός αυτόματου αλγορίθμου ταξινόμησης, και της τυχαίας συμφωνίας μεταξύ των επίγειων δεδομένων αναφοράς και ενός τυχαίου αλγορίθμου ταξινόμησης. Είναι μια ένδειξη του βαθμού, κατά τον οποίο οι ποσοστιαίες ορθές τιμές ενός πίνακα σφαλμάτων οφείλονται σε αληθή έναντι τυχαίας συμφωνίας. [Δημήτρης Π.Αργιαλάς, 1998]

$$k = \frac{\text{observed agreement} - \text{chance agreement}}{1 - \text{chance agreement}}$$

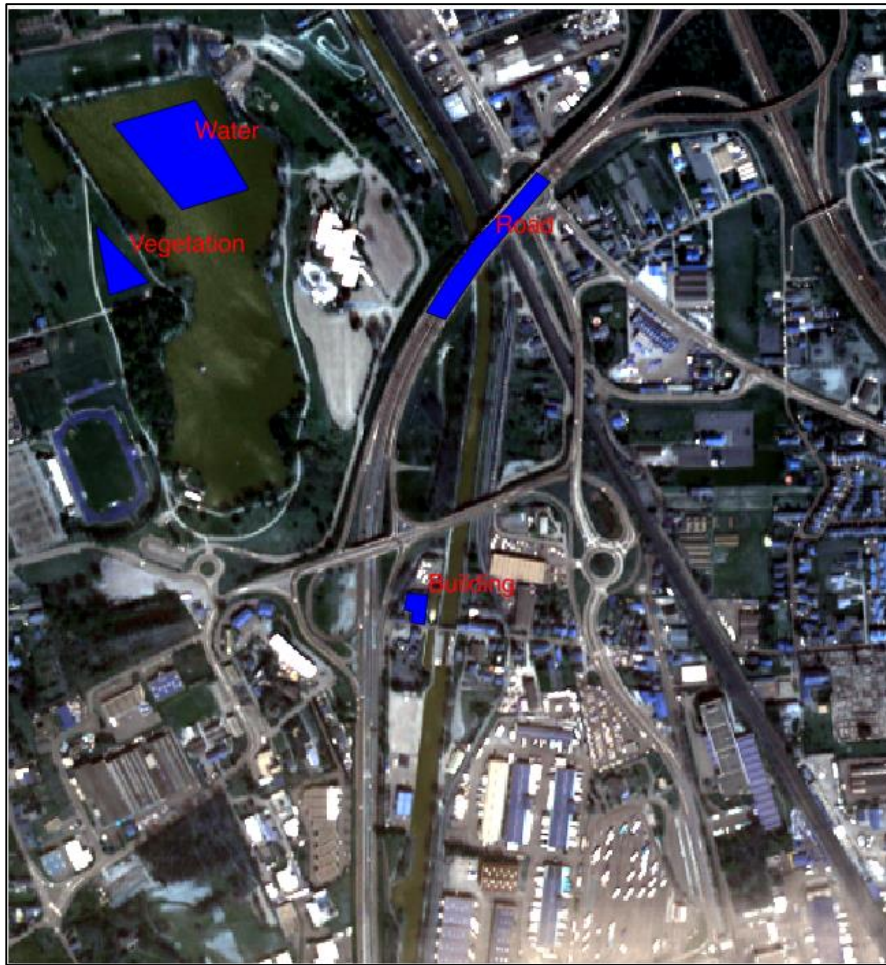
4.6 Παραδείγματα Εφαρμογών

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται εφαρμογές των αλγορίθμων ταξινόμησης που περιγράφηκαν στο κεφάλαιο 4.2. Για το σκοπό αυτό χρησιμοποιήθηκαν εικόνες και διανυσματικά δεδομένα από το φάκελο παραδειγμάτων του πηγαίου κώδικα του orfeo toolbox. Για τις διαδικασίες μηχανικής μάθησης και ταξινόμησης χρησιμοποιήθηκαν αλγόριθμοι από το orfeo toolbox καθώς και από τη βιβλιοθήκη του openCV χρησιμοποιώντας τη γραμμή εντολών σε περιβάλλον MacOS, τύπου Unix. Η έκδοση του orfeo toolbox που χρησιμοποιήθηκε ήταν η 4.0 και για τη βιβλιοθήκη openCV η 2.4.9.

Παρακάτω παρουσιάζονται οι εικόνες και τα πολύγωνα που χρησιμοποιήθηκαν, όπως αυτά λήφθηκαν από το φάκελο παραδειγμάτων του orfeo toolbox.



Εικόνα 4.17: Εικόνα QB_1_ortho.tif και πολύγωνα από QuickBird 4-καναλιών
Δεδομένα από φάκελο παραδειγμάτων orfeo-toolbox



Εικόνα 4.18: Εικόνα QB_2_ortho.tif και πολύγωνα από QuickBird 4-καναλιών
Δεδομένα από φάκελο παραδειγμάτων orfeo-toolbox

Προκειμένου να χρησιμοποιηθούν οι διαδικασίες μηχανικής μάθησης στη διαδικασία ταξινόμησης, αρχικά, δημιουργήθηκε ένα μοντέλο εκπαίδευσης χρησιμοποιώντας τα δεδομένα εκπαίδευσης της πρώτης εικόνας και έγινε προσπάθεια για την ταξινόμηση της δεύτερης εικόνας χωρίς να ληφθούν υπόψη τα δεδομένα εκπαίδευσης της δεύτερης. Στη συνέχεια, ελέγχθηκε η ακρίβεια της ταξινόμησης της δεύτερης εικόνας χρησιμοποιώντας ως δεδομένα ελέγχου της ταξινόμησης τα δεδομένα εκπαίδευσης της δεύτερης εικόνας. Πρέπει να τονιστεί πως οι εικόνες που χρησιμοποιήθηκαν είναι γεωμετρικά και ραδιομετρικά διορθωμένες, χωρίς σύννεφα και σκιές.

4.6.1 Παρουσίαση χρησιμοποιούμενης τεχνολογίας

Στο συγκεκριμένο κεφάλαιο παρουσιάζονται οι βιβλιοθήκες *orfeotoolbox* και *openCV*, οι οποίες χρησιμοποιήθηκαν για τις διαδικασίες της μηχανικής μάθησης και της ταξινόμησης και δίνεται έμφαση στους αντίστοιχους αλγορίθμους που χρησιμοποιήθηκαν.

Orfeo Toolbox

Το orfeo toolbox αποτελεί μία βιβλιοθήκη γραμμένη στη γλώσσα προγραμματισμού C++, που εστιάζεται στην επεξεργασία εικόνων καθώς και σε αλγόριθμους τηλεπισκόπησης. Δημιουργήθηκε το 2006 από το Γαλλικό Οργανισμό Διαστήματος και βασίζεται στις αρχές του ελεύθερου λογισμικού, κάτι που συντέλεσε στην άμεση ανάπτυξη και βελτίωση της βιβλιοθήκης σε πολύ μικρό χρονικό διάστημα. Το orfeo toolbox μπορεί να χρησιμοποιηθεί τόσο μέσω του πηγαίου κώδικα όσο και με τη μορφή γραμμών εντολών. Επιπλέον, οι αλγόριθμοι της βιβλιοθήκης μπορούν να εκτελεστούν μέσα από το γραφικό περιβάλλον με ονομασία Monteverdi. [<http://www.orfeo-toolbox.org/otb/2014>]

OpenCV

Η OpenCv είναι μία ελεύθερη βιβλιοθήκη επεξεργασίας εικόνας και βίντεο και είναι γραμμένη σε C/C++ γλώσσα προγραμματισμού. Υποστηρίζεται τόσο σε περιβάλλον Windows όσο και σε Unix, MacOS και εκτός των C/C++ υποστηρίζεται και σε άλλες γλώσσες προγραμματισμού όπως είναι η Python και η Java. Πρόκειται για μια διεπαφή που προσφέρει αίσθηση υψηλού επιπέδου γλώσσας προγραμματισμού και εκμεταλλεύεται σε σημαντικό βαθμό το σύστημα εξοπλισμού του υπολογιστή (hardware), προσφέροντας υψηλές ταχύτητες υλοποίησης σύνθετων προγραμμάτων. Επιπλέον, η βιβλιοθήκη μπορεί να χρησιμοποιηθεί σαν πρόσθετο και σε άλλες γλώσσες προγραμματισμού, όπως για παράδειγμα στο Matlab, octave, C# κ.α. [<http://opencv.org,2014>]

Στα πλαίσια της συγκεκριμένης εργασίας χρησιμοποιήθηκε η εφαρμογή otbTrainImagesClassifier της βιβλιοθήκης orfeo toolbox που χρησιμοποιεί παράλληλα τη βιβλιοθήκη OpenCV για τους διάφορους αλγόριθμους εκμάθησης. Πιο συγκεκριμένα, η βιβλιοθήκη του orfeo toolbox μέχρι την έκδοση 3.2 αντιμετώπιζε την εκμάθηση των μοντέλων αποκλειστικά με τον αλγόριθμο SVM, ωστόσο με την ένταξη της OpenCV χρησιμοποιεί πλέον πιο εξειδικευμένους αλγόριθμους εκμάθησης όπως είναι οι decision tree, KNN, Normal Bayes, Random Forests κ.α. Ο πηγαίος κώδικας της εφαρμογής otbTrainImagesClassifier βρίσκεται στο Παράρτημα Ι με ονομασία «Orfeo Toolbox – Train Images Classifier»

4.6.2 SVM

Όπως αναφέρθηκε παραπάνω, στόχος ήταν η προσπάθεια για ταξινόμηση μιας εικόνας χρησιμοποιώντας ένα στατιστικό μοντέλο ταξινόμησης που προέκυψε από δεδομένα εκπαίδευσης από άλλη εικόνα. Στο τέλος πραγματοποιήθηκε η αξιολόγηση της διαδικασίας με τα πολύγωνα εκπαίδευσης της ταξινομημένης εικόνας που δεν συμμετείχαν στο μοντέλο εκπαίδευσης.

Ταξινόμηση της εικόνας QB_2_ortho.tif με βάση το μοντέλο εκπαίδευσης από την QB_1_ortho.tif

Αρχικά υπολογίστηκαν τα στατιστικά της εικόνας QB_1_ortho.tif με την εξής εντολή:

```
otbcli_ComputeImagesStatistics -il orfeo_examples/QB_1_ortho.tif -out orfeo_examples/QB_1_stats.xml
```

όπου με `-il` ορίζεται η εικόνα εισόδου του αλγορίθμου και με `-out` τα στατιστικά εξόδου.

Στη συνέχεια πραγματοποιήθηκε η εκπαίδευση του μοντέλου από τα δεδομένα εκπαίδευσης της πρώτης εικόνας χρησιμοποιώντας γραμμικό μοντέλο μηχανών υποστήριξης διανυσμάτων:

```
otbcli_TrainImagesClassifier -io.il orfeo_examples/QB_1_ortho.tif -io.vd orfeo_examples/VectorData_QB1.shp -io.imstat orfeo_examples/QB_1_stats.xml -classifier libsvm -classifier.libsvm.k linear -sample.vfn Class -io.out orfeo_examples/model_1_svm
```

όπου με `-io.il` ορίζεται η εικόνα εισόδου, με `-io.vd` τα δεδομένα εκπαίδευσης εισόδου, με `-io.imstat` τα στατιστικά της εικόνας όπως υπολογίστηκαν παραπάνω, με `-classifier` η επιλογή του αλγορίθμου εκπαίδευσης, με `-classifier.libsvm.k` το είδος του ταξινομητή `svm`, με `-sample.vfn` το όνομα της στήλης του διανυσματικού αρχείου (shapefile) για την απόδοση τιμών στην ταξινομημένη εικόνα και με `-io.out` ορίζεται το μοντέλο εκπαίδευσης ως αρχείο εξόδου.

Ο πηγαίος κώδικας της δημιουργίας του στατιστικού μοντέλου εκμάθησης με τον αλγόριθμο SVM της βιβλιοθήκης `orfeo toolbox` βρίσκεται στο Παράρτημα I με ονομασία «Train Lib SVM».

Ακολουθούν τα στατιστικά του μοντέλου εκπαίδευσης όπως προέκυψαν από τον αλγόριθμο:

```
Application.logger (INFO) Number of training samples: 383
Application.logger (INFO) Size of training set: 383
Application.logger (INFO) Size of validation set: 360
Application.logger (INFO) Size of labeled training set: 383
Application.logger (INFO) Size of labeled validation set: 360
predicted list size == 360
validationLabeledListSample size == 360
Application.logger (INFO) training performances
Application.logger (INFO) Confusion matrix (rows = reference labels, columns = produced labels):
  [1] [2] [3] [4]
[1] 101  0  0  0
[2]  0  94  0  0
[3]  0  0  76  0
[4]  0  0  0  89
```

```

Application.logger (INFO) Precision of class [1] vs all: 1
Application.logger (INFO) Recall of class [1] vs all: 1
Application.logger (INFO) F-score of class [1] vs all: 1

Application.logger (INFO) Precision of class [2] vs all: 1
Application.logger (INFO) Recall of class [2] vs all: 1
Application.logger (INFO) F-score of class [2] vs all: 1

Application.logger (INFO) Precision of class [3] vs all: 1
Application.logger (INFO) Recall of class [3] vs all: 1
Application.logger (INFO) F-score of class [3] vs all: 1

Application.logger (INFO) Precision of class [4] vs all: 1
Application.logger (INFO) Recall of class [4] vs all: 1
Application.logger (INFO) F-score of class [4] vs all: 1

Application.logger (INFO) Global performance, Kappa index: 1

```

Το στατιστικό μοντέλο εκμάθησης όπως δημιουργήθηκε από τον αλγόριθμο παρουσιάζεται στο Παράρτημα Ι με ονομασία “Στατιστικό μοντέλο εκμάθησης svm από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif”.

Στη συνέχεια, υπολογίστηκαν τα στατιστικά της εικόνας QB_2_ortho.tif που πρόκειται να ταξινομηθεί:

```

otbcli_ComputeImagesStatistics -il orfeo_examples/QB_1_ortho.tif -out orfeo_examples/
QB_1_stats.xml

```

Στο επόμενο βήμα, πραγματοποιήθηκε η ταξινόμηση της δεύτερης εικόνας λαμβάνοντας υπόψη το στατιστικό μοντέλο εκπαίδευσης που δημιουργήθηκε από την πρώτη εικόνα:

```

otbcli_ImageClassifier -in orfeo_examples/QB_2_ortho.tif -imstat orfeo_examples/QB_2_stats.xml -
model orfeo_examples/model_1_svm -out orfeo_examples/QB_2_ortho_Class.tif

```

όπου με `-in` ορίζεται η εικόνα εισόδου προς ταξινόμηση, με `-imstat` τα στατιστικά εισόδου της εικόνας, με `-model` το μοντέλο εκπαίδευσης όπως δημιουργήθηκε με τον προηγούμενο αλγόριθμο και με `-out` ορίζεται το όνομα εξόδου της ταξινομημένης εικόνας.

Τέλος, υπολογίστηκε η ακρίβεια της ταξινόμησης χρησιμοποιώντας ως δεδομένα ελέγχου τα δεδομένα εκπαίδευσης της δεύτερης εικόνας ως εξής:

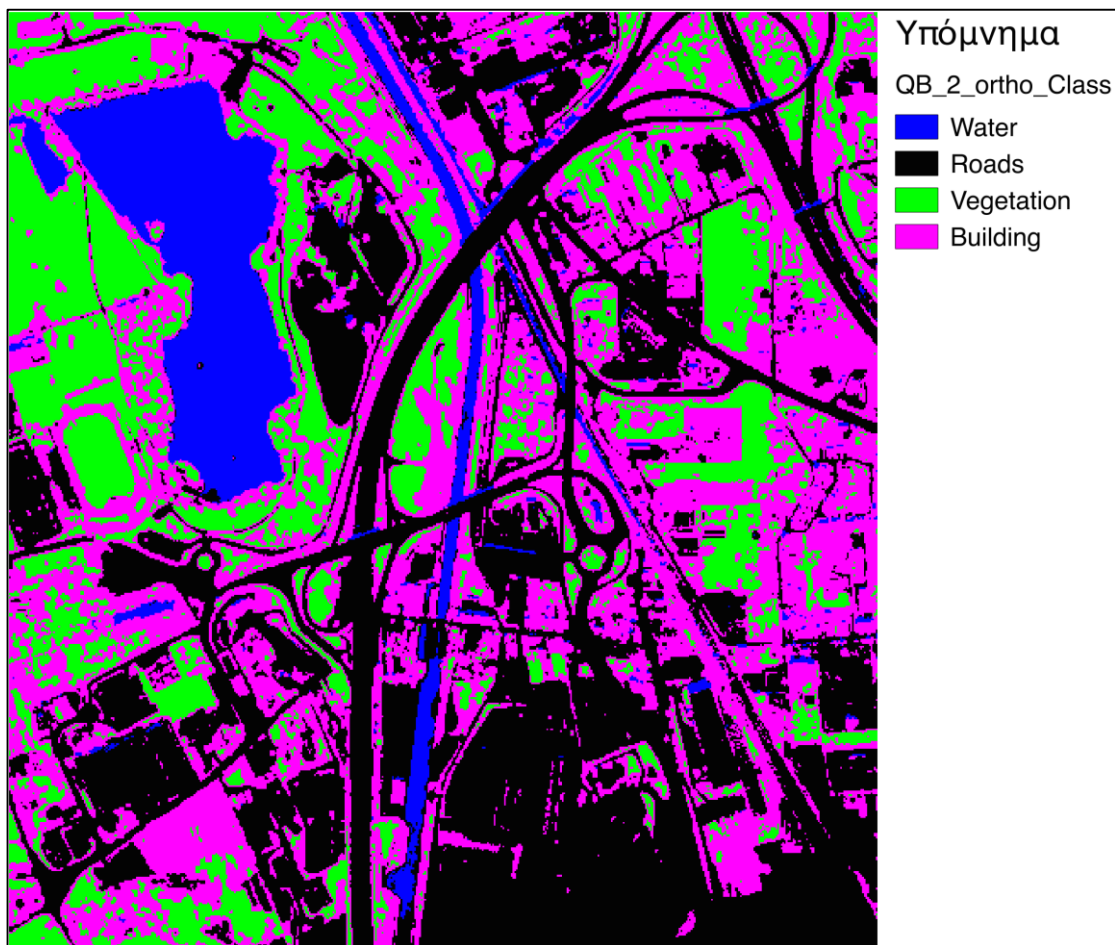
```

otbcli_ComputeConfusionMatrix -in orfeo_examples/QB_2_ortho_Class.tif -ref vector -ref.vector.in
orfeo_examples/VectorData_QB2.shp -ref.vector.field Class -out orfeo_examples/Qb2_conf.csv

```


όπου με `-in` ορίζεται η ταξινομημένη εικόνα εισόδου προς ταξινόμηση, με `-ref` vector ορίζεται πως η αξιολόγηση θα πραγματοποιηθεί με διανυσματικό αρχείο, με `-ref.vector.in` ορίζονται τα δεδομένα αξιολόγησης και με `-out` ορίζεται η έξοδος του αλγόριθμου που είναι ο πίνακας σύγχυσης.

Στη συνέχεια, παρουσιάζονται η ταξινομημένη εικόνα `QB_2_ortho_Class.tif` με τη μέθοδο των μηχανών υποστήριξης διανυσμάτων, τα στατιστικά όπως προέκυψαν από την αξιολόγηση καθώς και ο πίνακας σύγχυσης που προέκυψε από τα δεδομένα εκπαίδευσης της `QB_2_ortho.tif`, τα οποία δεν χρησιμοποιήθηκαν στη διαδικασία δημιουργίας του μοντέλου εκπαίδευσης.



Εικόνα 4.19: Ταξινομημένη εικόνα `QB_2_ortho.tif` με αλγόριθμο svm

Ακολουθούν τα στατιστικά αξιολόγησης όπως προέκυψαν από τον αλγόριθμο:

	[1]	[2]	[3]	[4]
[1]	2470	0	0	0
[2]	418	629	0	0
[3]	0	0	476	3
[4]	0	20	0	149

```

Application.logger (INFO) Precision of class [1] vs all: 0.855263
Application.logger (INFO) Recall of class [1] vs all: 1
Application.logger (INFO) F-score of class [1] vs all: 0.921986

Application.logger (INFO) Precision of class [2] vs all: 0.969183
Application.logger (INFO) Recall of class [2] vs all: 0.600764
Application.logger (INFO) F-score of class [2] vs all: 0.741745

Application.logger (INFO) Precision of class [3] vs all: 1
Application.logger (INFO) Recall of class [3] vs all: 0.993737
Application.logger (INFO) F-score of class [3] vs all: 0.996859

Application.logger (INFO) Precision of class [4] vs all: 0.980263
Application.logger (INFO) Recall of class [4] vs all: 0.881657
Application.logger (INFO) F-score of class [4] vs all: 0.928349

Application.logger (INFO) Precision of the different classes: [0.855263, 0.969183,
1, 0.980263]
Application.logger (INFO) Recall of the different classes: [1, 0.600764,
0.993737, 0.881657]
Application.logger (INFO) F-score of the different classes: [0.921986, 0.741745,
0.996859, 0.928349]

Application.logger (INFO) Kappa index: 0.802087
Application.logger (INFO) Overall accuracy index: 0.894118

```

4.6.3 Decision tree

Όμοια με το προηγούμενο κεφάλαιο, έγινε προσπάθεια για ταξινόμηση μιας εικόνας με άγνωστη γνώση εδάφους χρησιμοποιώντας ένα στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε από δεδομένα εκπαίδευσης άλλης εικόνας. Ακολουθεί παράδειγμα χρησιμοποιώντας το στατιστικό μοντέλο που δημιουργήθηκε από τον αλγόριθμο decision tree:

Ταξινόμηση της εικόνας QB_2_ortho.tif με βάση το μοντέλο εκπαίδευσης από την QB_1_ortho.tif

Αρχικά, υπολογίστηκαν τα στατιστικά της εικόνας QB_1_ortho.tif με την εξής εντολή:

```
otbcli_ComputeImagesStatistics -il orfeo_examples/QB_1_ortho.tif -out orfeo_examples/QB_1_stats.xml
```

Στη συνέχεια, πραγματοποιήθηκε η εκπαίδευση του μοντέλου από τα δεδομένα εκπαίδευσης της πρώτης εικόνας χρησιμοποιώντας τον αλγόριθμο του δέντρου απόφασης:


```
otbcli_TrainImagesClassifier -io.il orfeo_examples/QB_1_ortho.tif -io.vd
orfeo_examples/VectorData_QB1.shp -io.imstat orfeo_examples/QB_1_stats.xml -classifier dt -
classifier.dt.max 60000 -classifier.dt.min 1 -classifier.dt.f 5 -sample.vfn Class -io.out
orfeo_examples/model_1_dt
```

Για τα συγκεκριμένα δεδομένα εκπαίδευσης καθώς και για τις συγκεκριμένες εικόνες, ύστερα από δοκιμές στις παραμέτρους του μεγέθους του μέγιστου βάθους του δέντρου καθώς και της παραμέτρου για την ελάχιστη τιμή του αριθμού των δειγμάτων για κάθε κόμβο, παρατηρήθηκε πως τα καλύτερα αποτελέσματα, από άποψη ευστοχίας του στατιστικού μοντέλου και της αξιολόγησης της ταξινόμησης, έδωσαν οι τιμές 60000 στο μέγιστο βάθος του δέντρου καθώς και η τιμή 1 στον ελάχιστο αριθμό δειγμάτων ανά κόμβο. Ο πηγαίος κώδικας της δημιουργίας του στατιστικού μοντέλου εκμάθησης με τον αλγόριθμο Decision Tree της βιβλιοθήκης orfeo toolbox βρίσκεται στο Παράρτημα Ι με ονομασία «Train Decision Tree».

Ακολουθούν τα στατιστικά του μοντέλου εκπαίδευσης όπως προέκυψαν από τον αλγόριθμο:

```
Application.logger (INFO) Number of training samples: 365
Application.logger (INFO) Size of training set: 365
Application.logger (INFO) Size of validation set: 350
Application.logger (INFO) Size of labeled training set: 365
Application.logger (INFO) Size of labeled validation set: 350

Application.logger (INFO) training performances
Application.logger (INFO) Confusion matrix (rows = reference labels, columns = produced labels):
[1] [2] [3] [4]
[1] 90 0 0 0
[2] 0 78 0 3
[3] 0 0 94 0
[4] 0 2 0 83

Application.logger (INFO) Precision of class [1] vs all: 1
Application.logger (INFO) Recall of class [1] vs all: 1
Application.logger (INFO) F-score of class [1] vs all: 1

Application.logger (INFO) Precision of class [2] vs all: 0.975
Application.logger (INFO) Recall of class [2] vs all: 0.962963
Application.logger (INFO) F-score of class [2] vs all: 0.968944
Application.logger (INFO) Precision of class [3] vs all: 1
Application.logger (INFO) Recall of class [3] vs all: 1
Application.logger (INFO) F-score of class [3] vs all: 1

Application.logger (INFO) Precision of class [4] vs all: 0.965116
Application.logger (INFO) Recall of class [4] vs all: 0.976471
Application.logger (INFO) F-score of class [4] vs all: 0.97076

Application.logger (INFO) Global performance, Kappa index: 0.980931
```

Το στατιστικό μοντέλο εκμάθησης όπως δημιουργήθηκε από τον αλγόριθμο παρουσιάζεται στο Παράρτημα Ι με ονομασία “Στατιστικό μοντέλο εκμάθησης decision tree από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif”

Στη συνέχεια, υπολογίστηκαν τα στατιστικά της εικόνας QB_2_ortho.tif που πρόκειται να ταξινομηθεί:

```
otbcli_ComputeImagesStatistics -il orfeo_examples/QB_1_ortho.tif -out orfeo_examples/  
QB_1_stats.xml
```

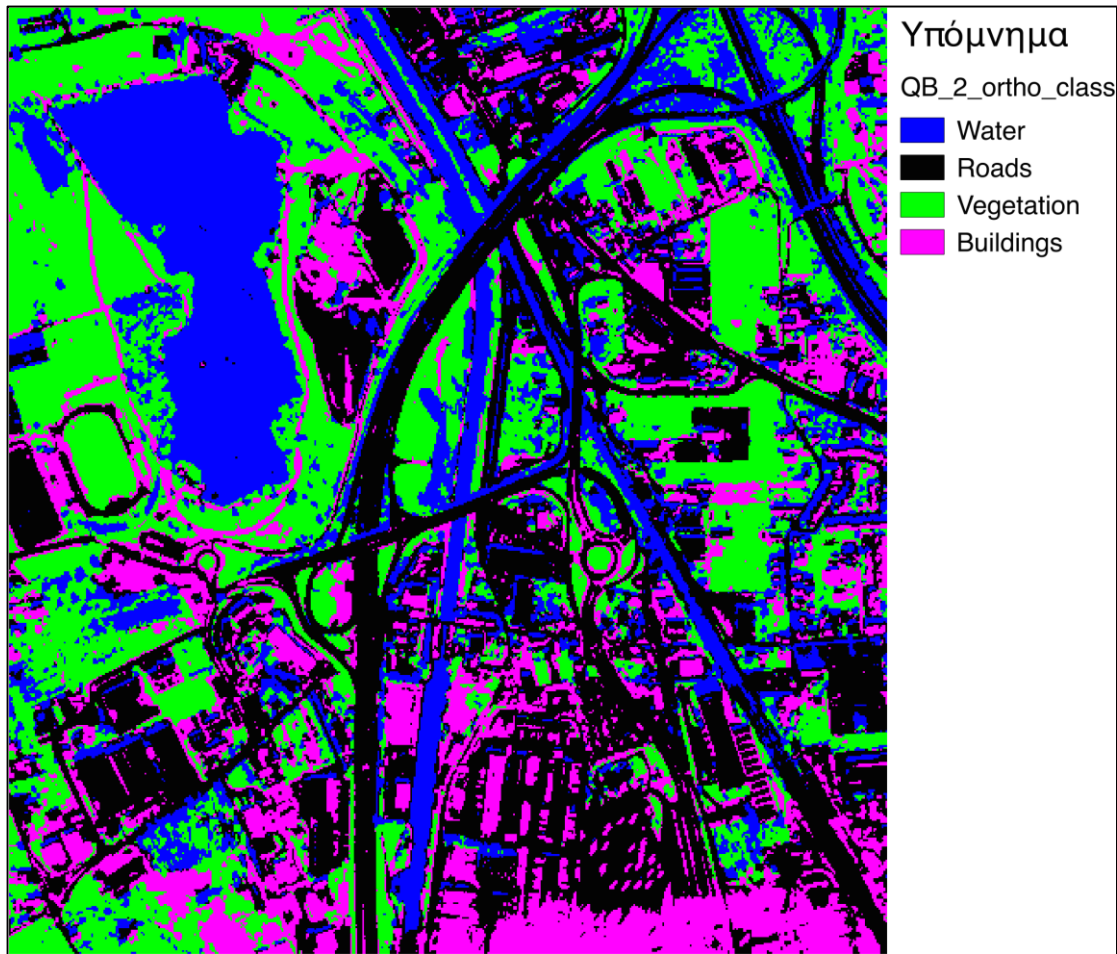
Στο επόμενο βήμα, πραγματοποιήθηκε η ταξινόμηση της δεύτερης εικόνας λαμβάνοντας υπόψη το στατιστικό μοντέλο εκπαίδευσης που δημιουργήθηκε από την πρώτη εικόνα:

```
otbcli_ImageClassifier -in orfeo_examples/QB_2_ortho.tif -imstat orfeo_examples/QB_2_stats.xml -  
model orfeo_examples/model_1_dt -out orfeo_examples/QB_2_ortho_Class.tif
```

Τέλος, υπολογίστηκε η ακρίβεια της ταξινόμησης χρησιμοποιώντας ως δεδομένα ελέγχου τα δεδομένα εκπαίδευσης της δεύτερης εικόνας ως εξής:

```
otbcli_ComputeConfusionMatrix -in orfeo_examples/QB_2_ortho_Class.tif -ref vector -ref.vector.in  
orfeo_examples/VectorData_QB2.shp -ref.vector.field Class -out orfeo_examples/Qb2_conf.csv
```

Στη συνέχεια, παρουσιάζονται η ταξινομημένη εικόνα QB_2_ortho_Class.tif με τη μέθοδο των δέντρων απόφασης, τα στατιστικά όπως προέκυψαν από την αξιολόγηση καθώς και ο πίνακας σύγχυσης που προέκυψε από τα δεδομένα εκπαίδευσης της QB_2_ortho.tif που δεν χρησιμοποιήθηκαν στη διαδικασία δημιουργίας του μοντέλου εκπαίδευσης.



Εικόνα 4.20: Ταξινομημένη εικόνα QB_2_ortho.tif με αλγόριθμο decision tree

Ακολουθούν τα στατιστικά αξιολόγησης όπως προέκυψαν από τον αλγόριθμο:

	[1]	[2]	[3]	[4]
[1]	2470	0	0	0
[2]	921	126	0	0
[3]	0	0	479	30
[4]	2	55	0	112

Application.logger (INFO) Precision of class [1] vs all: 0.727969

Application.logger (INFO) Recall of class [1] vs all: 1

Application.logger (INFO) F-score of class [1] vs all: 0.842572

Application.logger (INFO) Precision of class [2] vs all: 0.696133

Application.logger (INFO) Recall of class [2] vs all: 0.120344

Application.logger (INFO) F-score of class [2] vs all: 0.205212

Application.logger (INFO) Precision of class [3] vs all: 1

Application.logger (INFO) Recall of class [3] vs all: 1

Application.logger (INFO) F-score of class [3] vs all: 1

Application.logger (INFO) Precision of class [4] vs all: 1

```

Application.logger (INFO) Recall of class [4] vs all: 0.662722
Application.logger (INFO) F-score of class [4] vs all: 0.797153

Application.logger (INFO) Precision of the different classes: [0.727969, 0.696133,
1, 1]
Application.logger (INFO) Recall of the different classes: [1, 0.120344,
1, 0.662722]
Application.logger (INFO) F-score of the different classes: [0.842572, 0.205212,
1, 0.797153]

Application.logger (INFO) Kappa index: 0.522389

Application.logger (INFO) Overall accuracy index: 0.765186

```

4.6.4 K-nearest veighbors

Όμοια με τα προηγούμενα κεφάλαια, έγινε προσπάθεια για ταξινόμηση μιας εικόνας με άγνωστη γνώση εδάφους χρησιμοποιώντας ένα στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε από δεδομένα εκπαίδευσης άλλης εικόνας. Ακολουθεί παράδειγμα χρησιμοποιώντας το στατιστικό μοντέλο που δημιουργήθηκε από τον αλγόριθμο K-nearest neighbors:

Ταξινόμηση της εικόνας QB_2_ortho.tif με βάση το μοντέλο εκπαίδευσης από την QB_1_ortho.tif

Αρχικά, υπολογίστηκαν τα στατιστικά της εικόνας QB_1_ortho.tif με την εξής εντολή:

```
otbcli_ComputeImagesStatistics -il orfeo_examples/QB_1_ortho.tif -out orfeo_examples/
QB_1_stats.xml
```

Στη συνέχεια, πραγματοποιήθηκε η εκπαίδευση του μοντέλου από τα δεδομένα εκπαίδευσης της πρώτης εικόνας χρησιμοποιώντας τον αλγόριθμο των k-πλησιέστερων γειτόνων:

```
otbcli_TrainImagesClassifier -io.il orfeo_examples/QB_1_ortho.tif -io.vd
orfeo_examples/VectorData_QB1.shp -io.imstat orfeo_examples/QB_1_stats.xml --classifier knn --
classifier.knn.k 2 -sample.vfn Class -io.out orfeo_examples/model_1_knn
```

Για τα συγκεκριμένα δεδομένα εκπαίδευσης καθώς και για τις συγκεκριμένες εικόνες, ύστερα από δοκιμές στην παράμετρο k του αριθμού των γειτόνων, παρατηρήθηκε πως τα καλύτερα αποτελέσματα, από άποψη ευστοχίας του στατιστικού μοντέλου και της αξιολόγησης της ταξινόμησης, έδωσε η τιμή k=2. Ο πηγαίος κώδικας της δημιουργίας του στατιστικού μοντέλου εκμάθησης με τον αλγόριθμο SVM της βιβλιοθήκης orfeo toolbox βρίσκεται στο Παράρτημα I με ονομασία «Train KNN».

Ακολουθούν τα στατιστικά του μοντέλου εκπαίδευσης όπως προέκυψαν από τον αλγόριθμο:

```
Application.logger (INFO) Number of training samples: 338
Application.logger (INFO) Size of training set: 338
Application.logger (INFO) Size of validation set: 372
Application.logger (INFO) Size of labeled training set: 338
Application.logger (INFO) Size of labeled validation set: 372

Application.logger (INFO) training performances

Application.logger (INFO) Confusion matrix (rows = reference labels, columns = produced labels):
[1] [2] [3] [4]
[1] 102 0 0 0
[2] 0 92 0 0
[3] 0 0 87 0
[4] 0 0 0 91

Application.logger (INFO) Precision of class [1] vs all: 1
Application.logger (INFO) Recall of class [1] vs all: 1
Application.logger (INFO) F-score of class [1] vs all: 1

Application.logger (INFO) Precision of class [2] vs all: 1
Application.logger (INFO) Recall of class [2] vs all: 1
Application.logger (INFO) F-score of class [2] vs all: 1

Application.logger (INFO) Precision of class [3] vs all: 1
Application.logger (INFO) Recall of class [3] vs all: 1
Application.logger (INFO) F-score of class [3] vs all: 1

Application.logger (INFO) Precision of class [4] vs all: 1
Application.logger (INFO) Recall of class [4] vs all: 1
Application.logger (INFO) F-score of class [4] vs all: 1

Application.logger (INFO) Global performance, Kappa index: 1
```

Το στατιστικό μοντέλο εκμάθησης όπως δημιουργήθηκε από τον αλγόριθμο παρουσιάζεται στο Παράρτημα Ι με ονομασία “Στατιστικό μοντέλο εκμάθησης k-nearest neighbors από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif”.

Στη συνέχεια, υπολογίστηκαν τα στατιστικά της εικόνας QB_2_ortho.tif που πρόκειται να ταξινομηθεί:

```
otbcli_ComputeImagesStatistics -il orfeo_examples/QB_1_ortho.tif -out orfeo_examples/
QB_1_stats.xml
```

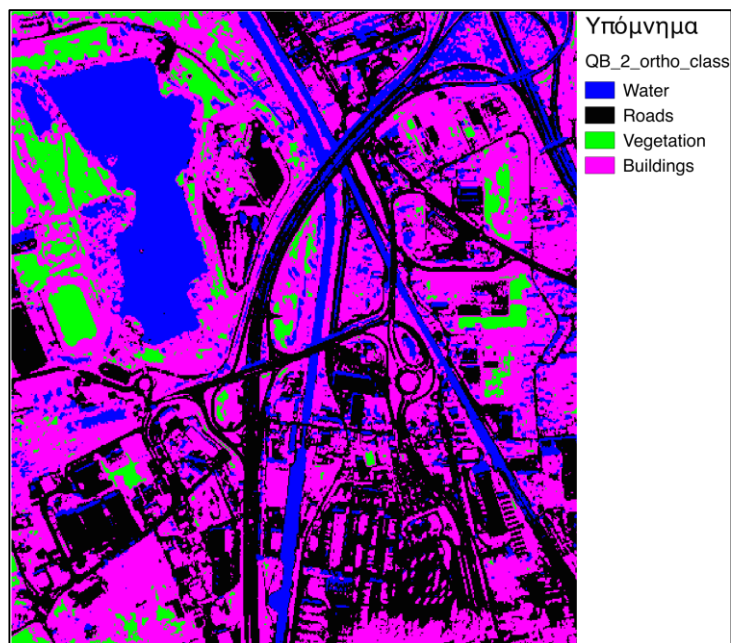
Στο επόμενο βήμα, πραγματοποιήθηκε η ταξινόμηση της δεύτερης εικόνας λαμβάνοντας υπόψη το στατιστικό μοντέλο εκπαίδευσης που δημιουργήθηκε από την πρώτη εικόνα:

```
otbcli_ImageClassifier -in orfeo_examples/QB_2_ortho.tif -imstat orfeo_examples/QB_2_stats.xml -
model orfeo_examples/model_1_knn -out orfeo_examples/QB_2_ortho_Class.tif
```

Τέλος, υπολογίστηκε η ακρίβεια της ταξινόμησης χρησιμοποιώντας ως δεδομένα ελέγχου τα δεδομένα εκπαίδευσης της δεύτερης εικόνας ως εξής:

```
otbcli_ComputeConfusionMatrix -in orfeo_examples/QB_2_ortho_Class.tif -ref vector -ref.vector.in
orfeo_examples/VectorData_QB2.shp -ref.vector.field Class -out orfeo_examples/Qb2_conf.csv
```

Στη συνέχεια, παρουσιάζονται η ταξινομημένη εικόνα QB_2_ortho_Class.tif με τη μέθοδο των k-πλησιέστερων γειτόνων, τα στατιστικά όπως προέκυψαν από την αξιολόγηση καθώς και ο πίνακας σύγχυσης που προέκυψε από τα δεδομένα εκπαίδευσης της QB_2_ortho.tif που δεν χρησιμοποιήθηκαν στη διαδικασία δημιουργίας του μοντέλου εκπαίδευσης.



Εικόνα 4.21: Ταξινομημένη εικόνα QB_2_ortho.tif με αλγόριθμο k-nearest neighbors

Ακολουθούν τα στατιστικά αξιολόγησης όπως προέκυψαν από τον αλγόριθμο:

	[1]	[2]	[3]	[4]
[1]	2470	0	0	0
[2]	570	471	0	6
[3]	0	0	452	27
[4]	10	7	0	152

Application.logger (INFO) Precision of class [1] vs all: 0.809836
 Application.logger (INFO) Recall of class [1] vs all: 1
 Application.logger (INFO) F-score of class [1] vs all: 0.894928

Application.logger (INFO) Precision of class [2] vs all: 0.985356
 Application.logger (INFO) Recall of class [2] vs all: 0.449857

```
Application.logger (INFO) F-score of class [2] vs all: 0.617705

Application.logger (INFO) Precision of class [3] vs all: 1
Application.logger (INFO) Recall of class [3] vs all: 0.943633
Application.logger (INFO) F-score of class [3] vs all: 0.970999

Application.logger (INFO) Precision of class [4] vs all: 0.821622
Application.logger (INFO) Recall of class [4] vs all: 0.899408
Application.logger (INFO) F-score of class [4] vs all: 0.858757

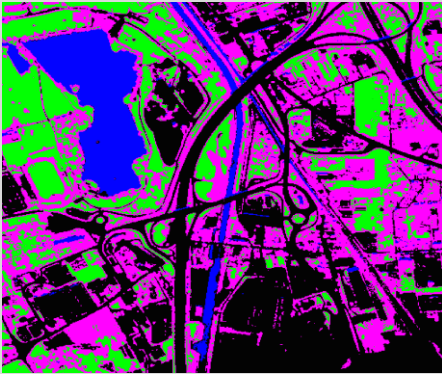
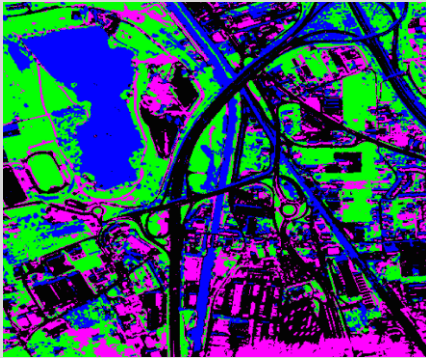
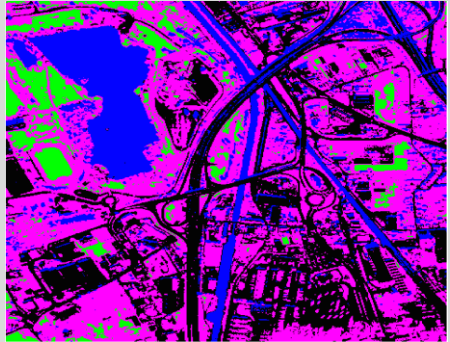
Application.logger (INFO) Precision of the different classes: [0.809836, 0.985356,
1, 0.821622]
Application.logger (INFO) Recall of the different classes: [1, 0.449857,
0.943633, 0.899408]
Application.logger (INFO) F-score of the different classes: [0.894928, 0.617705,
0.970999, 0.858757]

Application.logger (INFO) Kappa index: 0.71515

Application.logger (INFO) Overall accuracy index: 0.85114
```

4.7 Σύγκριση – Αξιολόγηση Αποτελεσμάτων

Στο συγκεκριμένο κεφάλαιο αξιολογούνται τα αποτελέσματα του παραπάνω κεφαλαίου και ελέγχεται ποιος αλγόριθμος ταξινόμησης αντιμετώπισε πιο αποτελεσματικά το πρόβλημα της ταξινόμησης. Ωστόσο, πρέπει να τονιστεί πως χρησιμοποιήθηκαν όλα τα δεδομένα όπως αυτά περιέχονταν στο φάκελο παραδειγμάτων του λογισμικού orfeo toolbox και δεν ακολουθήθηκε ξεχωριστή διαδικασία για κάθε αλγόριθμο ταξινόμησης. Ακολουθεί συγκεντρωτικός πίνακας των ταξινομημένων εικόνων και των στατιστικών αξιολόγησης για τους αλγορίθμους που χρησιμοποιήθηκαν.

Linear SVM	Decision Tree	KNN
		
Kappa index: 0.802087	Kappa index: 0.522389	Kappa index: 0.71515
Overall accuracy: 0.894118	Overall accuracy: 0.765186	Overall accuracy: 0.85114

Πίνακας 4.8: Συγκεντρωτικά αποτελέσματα σειράς ταξινομήσεων παραδειγμάτων

Παρατηρήσεις

Από τον παραπάνω πίνακα καθώς και από τη φωτοερμηνεία των ταξινομήσεων σε σχέση με την αρχική εικόνα γίνεται αντιληπτό πως ο ταξινομητής svm αντιμετώπισε πιο αποτελεσματικά το πρόβλημα της ταξινόμησης. Ακολούθησε ο αλγόριθμος KNN και στο τέλος ο αλγόριθμος decision tree. Γίνεται αντιληπτό πως οι αλγόριθμοι svm, knn που απαιτούν λιγότερες παραμέτρους αντιμετώπισαν πιο αποτελεσματικά το πρόβλημα, ενώ ο αλγόριθμος decision tree που απαιτεί μεγάλο πλήθος δοκιμών στις παραμέτρους δεν είχε ικανοποιητικά αποτελέσματα. Ωστόσο, με πιο εξειδικευμένα πολύγωνα εκπαίδευσης καθώς και με κατάλληλη βελτιστοποίηση στην επιλογή των παραμέτρων είναι πιθανό να βελτιωθούν τα αποτελέσματα του αλγορίθμου decision tree. Πιο συγκεκριμένα, ο αλγόριθμος svm ταξινόμησε με ικανοποιητική ακρίβεια τις τάξεις και διαφοροποίησε επαρκώς τις κατηγορίες των νερών με τους δρόμους, οι οποίες φαίνεται πως δημιούργησαν προβλήματα στις άλλες δύο μεθόδους με μεγαλύτερα προβλήματα στην ταξινόμηση του αλγορίθμου KNN. Ένα πρόβλημα που έγινε αντιληπτό στον αλγόριθμο svm ήταν η αδυναμία του να ξεχωρίσει πλήρως κάποιες εκτάσεις αστικού πρασίνου από τον αστικό χώρο, το οποίο με πιο «σωστά» και στοχευμένα πολύγωνα εκπαίδευσης ίσως επιλυόταν. Τέλος, ο αλγόριθμος decision tree φαίνεται πως εντόπισε αποτελεσματικά όλες τις κατηγορίες αστικού πρασίνου, ωστόσο γενίκευσε αρκετές κατηγορίες προσφέροντας πιο ομαλοποιημένα αλλά λανθασμένα πρότυπα.

Κεφάλαιο 5: Ανάπτυξη Μεθοδολογίας
Μηχανικής Μάθησης για την Ανίχνευση
Μεταβολών

5.1 Συνοπτική Παρουσίαση Μεθοδολογίας

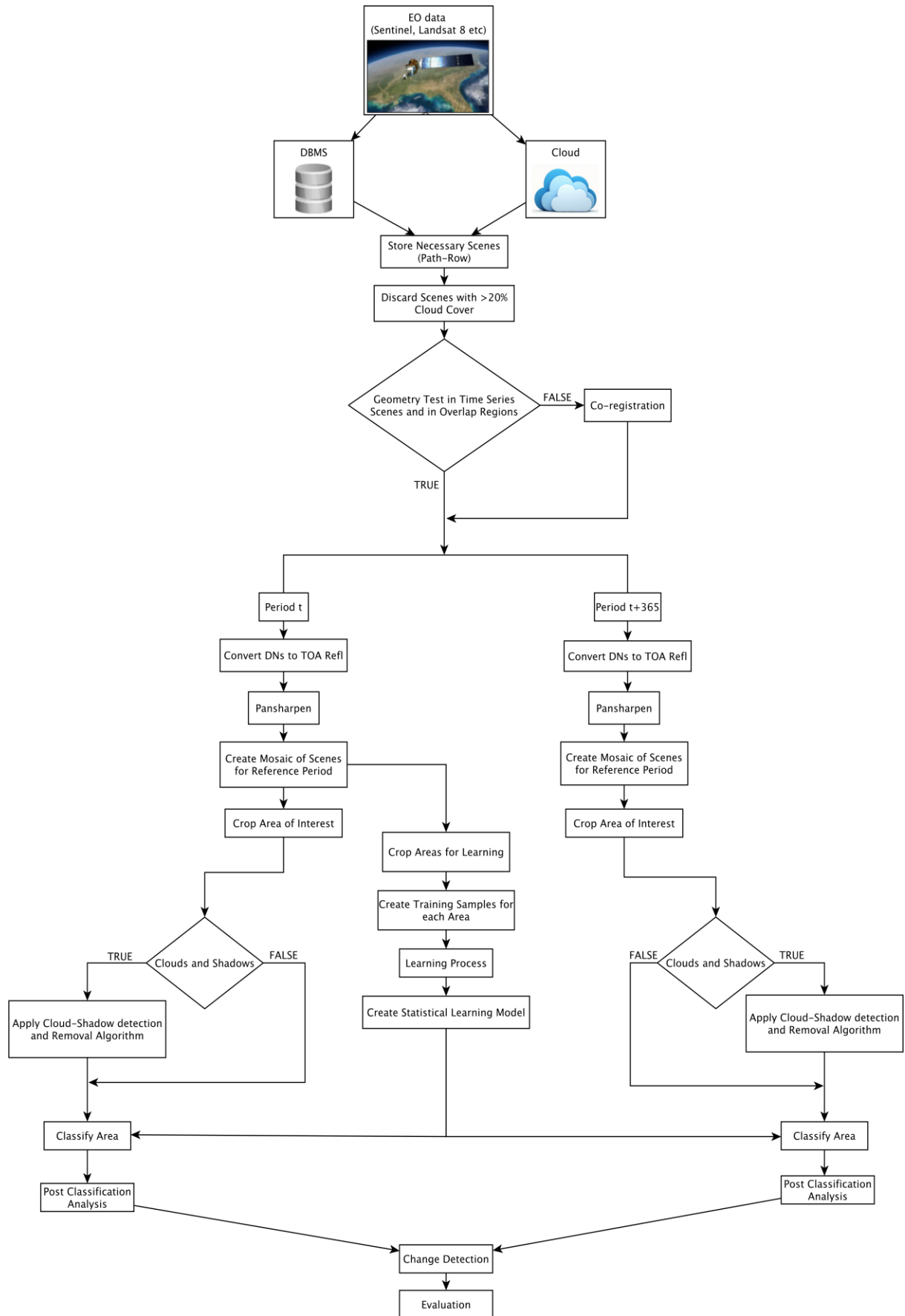
Στα πλαίσια της διπλωματικής εργασίας έγινε προσπάθεια για την κατάστρωση μεθοδολογίας που αντιμετωπίζει αποτελεσματικά την αυτόματη ανίχνευση μεταβολών αξιοποιώντας αλγορίθμους ταξινόμησης με διαδικασίες μηχανικής μάθησης καθώς και μεγάλα διαχρονικά γεωχωρικά δεδομένα. Πιο συγκεκριμένα, η μεθοδολογική προσέγγιση που αναπτύχθηκε αξιοποιεί δεδομένα από το δορυφορικό τηλεπισκοπικό δέκτη Landsat 8 καθώς και διανυσματικά δεδομένα από τον ελληνικό ιστοχώρο Δημοσίων και Ανοιχτών Δεδομένων geodata.gov.gr. Επιπλέον, αξιοποιήθηκε ποιοτική πληροφορία από το [google earth](http://google.com/earth) για να πραγματοποιηθεί η ψηφιοποίηση και κατασκευή νέων πολυγώνων εκπαίδευσης για τη διαδικασία εκμάθησης. Παράλληλα, επισημαίνεται πως βασικό χαρακτηριστικό της συγκεκριμένης μεθοδολογίας είναι η αξιοποίηση και χρήση δεδομένων και λογισμικών που διέπονται από τις αρχές του ανοιχτού λογισμικού και ανοιχτών προτύπων αντίστοιχα.

Αρχικά, επιλέχθηκαν τα κατάλληλα δορυφορικά δεδομένα λαμβάνοντας υπόψη διάφορα κριτήρια που παρουσιάζονται αναλυτικά στη συνέχεια του κεφαλαίου. Στη συνέχεια, πραγματοποιήθηκε ο έλεγχος της γεωμετρίας τόσο μεταξύ των ίδιων σκηνών, σε διάφορες χρονικές περιόδους, όσο και μεταξύ των γειτονικών σκηνών για να εξεταστούν πιθανά προβλήματα στα επικαλυπτόμενα τμήματα. Εφόσον δεν παρατηρήθηκαν προβλήματα στη γεωμετρία πραγματοποιήθηκαν οι απαιτούμενες ραδιομετρικές διορθώσεις, για κάθε επιλεγμένη σκηνή, με σκοπό τη μετατροπή των ψηφιακών τιμών σε τιμές φαινόμενης ανακλαστικότητας (TOA reflectance). Στη συνέχεια, πραγματοποιήθηκε αλγόριθμος pansharpening, για κάθε επιλεγμένη σκηνή, με σκοπό τη δημιουργία υψηλής ανάλυσης δεδομένων, αξιοποιώντας το παγχρωματικό κανάλι (κανάλι 8) του δέκτη OLI.

Ακολούθησε η δημιουργία του μωσαικού από τις επιλεγμένες σκηνές για μια συγκεκριμένη χρονική στιγμή, που θεωρήθηκε ως ημερομηνία αναφοράς. Έχοντας δημιουργήσει το μωσαικό της ευρύτερης περιοχής, όπου και εφαρμόστηκε η μεθοδολογία, δημιουργήθηκε ένα στατιστικό μοντέλο εκμάθησης, το οποίο προέκυψε από τις κατάλληλες υποπεριοχές της επιλεγμένης αυτής περιοχής. Τα κριτήρια επιλογής των παραπάνω υποπεριοχών αναλύονται στα κεφάλαια που ακολουθούν. Για τη δημιουργία των δεδομένων εκπαίδευσης λήφθηκαν υπόψη αφενός τα έτοιμα πολύγωνα από τον ιστοχώρο του geodata.gov.gr και αφετέρου πολύγωνα που προέκυψαν από ψηφιοποίηση στις υποπεριοχές. Δημιουργήθηκαν, λοιπόν, αναλυτικά πολύγωνα εκπαίδευσης για κάθε κατηγορία χρήσης γης με σκοπό την ενίσχυση του στατιστικού μοντέλου εκπαίδευσης.

Στη συνέχεια, επιλέχθηκε τυχαία μια περιοχή του μωσαικού που κατασκευάστηκε, η οποία δεν συμμετείχε στη διαδικασία εκπαίδευσης, έτσι ώστε να εξεταστεί η ακρίβεια και η αποτελεσματικότητα του μοντέλου εκπαίδευσης που δημιουργήθηκε νωρίτερα. Για την ανίχνευση και αντικατάσταση των συννέφων και σκιών αναπτύχθηκε αλγόριθμος προκειμένου να αφαιρεθούν τα σύννεφα από την περιοχή μελέτης και να ενισχυθεί η ακρίβεια των αποτελεσμάτων από την ταξινόμηση. Η αναλυτική παρουσίαση του αλγορίθμου ακολουθεί στις επόμενες ενότητες. Στο επόμενο βήμα, πραγματοποιήθηκε και ελέγχθηκε η ταξινόμηση της περιοχής μελέτης λαμβάνοντας υπόψη το στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε. Τέλος, για την περαιτέρω αξιολόγηση του μοντέλου εκπαίδευσης ταξινομήθηκε η περιοχή μελέτης μετά από ένα χρόνο και έγινε προσπάθεια για να ανιχνευθούν οι μεταβολές στις χρήσεις γης μεταξύ των δύο χρονολογιών.

Παρακάτω ακολουθεί η διαγραμματική παρουσίαση της μεθοδολογίας που ακολουθήθηκε.



Διάγραμμα 5.1: Προτεινόμενη Μεθοδολογία

5.2 Προεργασία

Ένα από τα πιο σημαντικά στάδια στη διαδικασία ανάλυσης δορυφορικών τηλεπισκοπικών δεδομένων είναι η προεργασία και προετοιμασία των δεδομένων για μετέπειτα επεξεργασία. Το στάδιο της προεργασίας γίνεται ακόμα πιο αναγκαίο όταν η ανάλυση απαιτεί τη διαχείριση μεγάλων διαχρονικών δεδομένων καθώς και διαδικασίες που αξιοποιούν αλγορίθμους ταξινόμησης. Στο συγκεκριμένο κεφάλαιο παρουσιάζονται οι απαραίτητες διαδικασίες προεπεξεργασίας δεδομένων για την προτεινόμενη μεθοδολογία.

5.2.1 Επιλογή Δεδομένων

Μεγάλη σημασία για την έκβαση ικανοποιητικών αποτελεσμάτων έχει η επιλογή των κατάλληλων δεδομένων που θα χρησιμοποιηθούν στη διαδικασία ανάλυσης. Αρχικά, είναι απαραίτητο να επιλεγούν σκηνές με μικρά ποσοστά κάλυψης από σύννεφα, κάτι που είναι ιδιαίτερα εύκολο στα δορυφορικά δεδομένα του Landsat 8, καθώς στο αρχείο των μεταδεδομένων αναγράφεται το ποσοστό κάλυψης από σύννεφα ως «Cloud Cover». Στα πλαίσια της συγκεκριμένης μεθοδολογίας επιλέχτηκαν σκηνές με ποσοστά κάλυψης από σύννεφα μικρότερα του 20%, έτσι ώστε αυτές να χρήζουν περαιτέρω επεξεργασίας. Επιπλέον, προκειμένου να αντιμετωπιστούν προβλήματα φαινολογίας είναι να αναγκαίο να χρησιμοποιηθούν δεδομένα από την ίδια περίοδο (καλοκαίρι), καθώς κατά τους καλοκαιρινούς μήνες η ύπαρξη συννέφων είναι πιο περιορισμένη, ενώ παράλληλα έχουν παρατηρηθεί πιο αξιόπιστα αποτελέσματα στους αλγόριθμους ταξινόμησης. *[Sinha, Kumar, & Reid, 2011]*

5.2.2 Έλεγχος Γεωμετρίας

Απαραίτητο στοιχείο του συνόλου της μεθοδολογίας είναι ο έλεγχος της γεωμετρίας τόσο μεταξύ των διαχρονικών όμοιων σκηνών (ίδιο path και row) όσο και των επικαλυπτόμενων τμημάτων μεταξύ των περιμετρικών σκηνών. Ο συγκεκριμένος έλεγχος εξασφαλίζει τη σωστή κατασκευή του μωσαϊκού, ώστε να μην δημιουργηθούν παραμορφώσεις στο επικαλυπτόμενο τμήμα, καθώς και τη σωστή προσαρμογή των πολυγώνων εκπαίδευσης στο σύνολο των διαχρονικών δεδομένων, ώστε να μην αλλοιωθούν τα στατιστικά.

5.2.3 Ραδιομετρικές Διορθώσεις

Όσον αφορά τις απαιτούμενες ραδιομετρικές διορθώσεις, πραγματοποιήθηκε η μετατροπή των ψηφιακών τιμών των εικονοστοιχείων σε τιμές φαινομένης ανακλαστικότητας (top of atmosphere reflectance) για τα κανάλια του δέκτη OLI, όπου σύμφωνα με τη βιβλιογραφία της USGS, η μετατροπή των ψηφιακών τιμών σε τιμές ανακλαστικότητας πραγματοποιείται σε δύο στάδια. Για τα θερμικά κανάλια του δέκτη TIRS πραγματοποιήθηκε η μετατροπή των ψηφιακών τιμών τους σε τιμές

ακτινοβολίας και στη συνέχεια σε τιμές θερμοκρασίας σύμφωνα με τις εξισώσεις της USGS. [http://landsat.usgs.gov/Landsat8_Using_Product.php]

Conversion to TOA Reflectance

Η μετατροπή των ψηφιακών τιμών σε τιμές φαινόμενης ανακλαστικότητας πραγματοποιήθηκε με τις παρακάτω εξισώσεις:

$$\rho\lambda' = M_p Q_{cal} + A_p$$

όπου $\rho\lambda'$ η τιμή ανακλαστικότητας χωρίς τη διόρθωση από τη γωνία του ήλιου, M_p ο πολλαπλασιαστικός αναβαθμονομημένος συντελεστής, του οποίου η τιμή βρίσκεται στα μεταδεδομένα με ονομασία REFLECTANCE_MULT_BAND_x, A_p ο αθροιστικός αναβαθμονομημένος συντελεστής, του οποίου η τιμή βρίσκεται στα μεταδεδομένα με ονομασία REFLECTANCE_ADD_BAND_x και Q_{cal} η ψηφιακή τιμή του εικοστοιχείου όπως αυτή λήφθηκε.

Στη συνέχεια, πραγματοποιήθηκε η διόρθωση από τη γωνία του ήλιου με την εξής εξίσωση:

$$\rho\lambda = \frac{\rho\lambda'}{\cos(\theta_{SZ})} = \frac{\rho\lambda'}{\sin(\theta_{SE})}$$

όπου $\rho\lambda$ η τιμή της φαινόμενης πλανητικής ανακλαστικότητας (TOA planetary reflectance), θ_{SE} η τοπική γωνία ανύψωσης του ηλίου, της οποίας η τιμή βρίσκεται στα μεταδεδομένα με ονομασία SUN_ELEVATION και θ_{SZ} η τοπική ηλιακή ζενίθια γωνία που υπολογίζεται από τη σχέση $\theta_{SZ} = 90^\circ - \theta_{SE}$.

Κατά την εφαρμογή του μετασχηματισμού παρατηρήθηκε πως ορισμένα εικονοστοιχεία είχαν αρνητική τιμή ανακλαστικότητας, η οποία δεν έχει φυσική σημασία, ενώ το μεγαλύτερο μέρος τους ανήκε είτε στο περίγραμμα της εικόνας, είτε στα εσωτερικά της εικόνας σε σημεία που παρατηρήθηκε το φαινόμενο του «θορύβου». Για την αντιμετώπιση του συγκεκριμένου προβλήματος τέθηκαν οι αρνητικές τιμές ανακλαστικότητας ίσες με το μηδέν, έτσι ώστε να μην επηρεαστούν τα στατιστικά της εικόνας. [*Yale guide to Landsat 8 image processing, 2013*]

Conversion to At-Satellite Brightness Temperature

Αρχικά, πραγματοποιήθηκε η μετατροπή των ψηφιακών τιμών σε τιμές φαινόμενης ακτινοβολίας με την παρακάτω εξίσωση:

$$L_\lambda = M_L Q_{cal} + A_L$$

όπου L_λ η τιμή ακτινοβολίας ($\text{Watts}/(\text{m}^2 \cdot \text{srad} \cdot \mu\text{m})$), M_L ο πολλαπλασιαστικός αναβαθμονομημένος συντελεστής, του οποίου η τιμή βρίσκεται στα μεταδεδομένα με ονομασία `RADIANCE_MULT_BAND_x`, A_L ο αθροιστικός αναβαθμονομημένος συντελεστής, του οποίου η τιμή βρίσκεται στα μεταδεδομένα με ονομασία `RADIANCE_ADD_BAND_x` και Q_{cal} η ψηφιακή τιμή του εικοστοιχείου όπως αυτή λήφθηκε.

Στη συνέχεια, πραγματοποιήθηκε η μετατροπή των τιμών ακτινοβολίας σε τιμές θερμοκρασίας σε βαθμούς Kelvin με την εξής εξίσωση:

$$T = \frac{K_2}{\ln\left(\frac{K_1}{L_\lambda} + 1\right)}$$

όπου K_1 , K_2 , οι ειδικές θερμικές σταθερές μετατροπής, των οποίων οι τιμές βρίσκονται στα μεταδεδομένα με ονομασία `K1_CONSTANT_BAND_x` και `K2_CONSTANT_BAND_x` αντίστοιχα.

Στο Παράρτημα II παρουσιάζονται τα προγράμματα που δημιουργήθηκαν σε γλώσσα προγραμματισμού C για την εκτέλεση των μετατροπών με ονομασίες: «Μετατροπή Ψηφιακών Τιμών σε Τιμές Φαινόμενης Ανακλαστικότητας για τον Δέκτη OLI του Δορυφόρου Landsat 8», «Μετατροπή Ψηφιακών Τιμών σε Τιμές Θερμοκρασίας για το Κανάλι 10 του Δορυφόρου Landsat 8» και «Μετατροπή Ψηφιακών Τιμών σε Τιμές Θερμοκρασίας για το Κανάλι 11 του Δορυφόρου Landsat 8»

5.2.4 Pansharpening

Για τη διαδικασία του pansharpening χρησιμοποιήθηκε ο αλγόριθμος του orfeotoolbox, ο οποίος πραγματοποιεί τα παρακάτω βήματα:

[<http://orfeo-toolbox.org/SoftwareGuide/SoftwareGuidech13.html#x39-2070011>]

- Εφαρμογή φίλτρου low-pass στο παγχρωματικό κανάλι (Band 8) για να δημιουργηθεί φασματικό περιεχόμενο όμοιο με των πολυφασματικών δεδομένων
- Ομαλοποίηση των πολυφασματικών δεδομένων με το παγχρωματικό κανάλι που δημιουργήθηκε μετά την εφαρμογή του φίλτρου low-pass
- Πολλαπλασιασμός του αποτελέσματος με το αρχικό παγχρωματικό κανάλι

Η διαδικασία σύντηξης (fusion) πραγματοποιείται με την παρακάτω σχέση:

$$Fused_Image = \frac{XS}{Filtered(PAN)} PAN$$

όπου XS η δορυφορική εικόνα με τα πολυφασματικά κανάλια.

Στο Παράρτημα I βρίσκεται ο κώδικας του αλγορίθμου με ονομασία «Orfeo Toolbox – Bundle To Perfect Sensor»

5.2.5 Κατασκευή Μωσαικού

Στη συνέχεια, ακολουθεί η κατασκευή του μωσαικού με τις διορθωμένες εικόνες που προέκυψαν από τα παραπάνω βήματα. Ωστόσο, δόθηκε ιδιαίτερη προσοχή στην κατάλληλη επιλογή των σκηνών, στην αντιμετώπιση των επικαλυπτόμενων τμημάτων καθώς και στο εύρος των ραδιομετρικών τιμών προς απεικόνιση στο παραγόμενο μωσαικό. Όσον αφορά την επιλογή των εικόνων, χρησιμοποιήθηκαν εκείνες που λήφθηκαν την ίδια εποχή (π.χ. καλοκαίρι) και χρόνο έτσι ώστε να αντιμετωπιστούν τα προβλήματα της φαινολογίας. Για τα επικαλυπτόμενα τμήματα διατηρήθηκαν οι ραδιομετρικές τιμές της καταλληλότερης εικόνας, η οποία επιλέχθηκε με βάση τα εξής κριτήρια:

- αν οι επικαλυπτόμενες περιοχές χαρακτηρίζονταν από χαμηλά επίπεδα συννεφοκάλυψης πριν την εφαρμογή του αλγορίθμου για την αφαίρεση τους
- αν οι επικαλυπτόμενες περιοχές βρίσκονταν μακριά από τα πλαίσια της εικόνας, όπου και παρατηρούνται οι μεγαλύτερες παραμορφώσεις

Τέλος, για το εύρος των ραδιομετρικών τιμών προς απεικόνιση, εφόσον οι δορυφορικές εικόνες έχουν ληφθεί μέσα στην ίδια εποχή και με χρονικό βήμα ± 16 ημερών, πραγματοποιήθηκε η απεικόνιση των κοινών ραδιομετρικών τιμών (σύμφωνα με τα ιστογράμματα των εικόνων προς συγχώνευση) έτσι ώστε να προκύψει οπτικά μια πιο ομαλή δορυφορική εικόνα.

5.3 Ανίχνευση και Αφαίρεση Συννέφων και Σκιών

Προκειμένου να παραχθούν ταξινομημένες εικόνες υψηλής ακρίβειας ήταν απαραίτητο να ανιχνευθούν και να αντικατασταθούν τα σύννεφα και οι σκιές από τις επιλεγμένες σκηνές. Παρόλο που η ανίχνευση των συννέφων δεν είναι δύσκολη, η εξάλειψη και αντικατάσταση αυτών αποτελεί ένα αρκετά πολύπλοκο πρόβλημα, αφού γίνεται προσπάθεια να μην αλλοιωθεί η πραγματικότητα και να διατηρηθούν τα αρχικά χαρακτηριστικά της εικόνας.

5.3.1 Μέθοδοι ανίχνευσης συννέφων και σκιών

Στη συνέχεια, παρουσιάζονται κάποιες μέθοδοι που έχουν αναπτυχθεί στα πλαίσια της ανίχνευσης συννέφων και σκιών και στο κεφάλαιο 5.3.2 παρουσιάζεται ο αλγόριθμος που δημιουργήθηκε για την ανίχνευση και αφαίρεση συννέφων και σκιών σε δορυφορικά τηλεπισκοπικά δεδομένα του Landsat 8.

Ανίχνευση Συννέφων

Όσον αφορά την ανίχνευση συννέφων στα δορυφορικά τηλεπισκοπικά δεδομένα του Landsat 8, έχουν προταθεί διάφορες μέθοδοι. Παρακάτω παρουσιάζονται οι τρεις πιο αποτελεσματικές εξ' αυτών.

- 1) Χρήση του καναλιού BQA του Landsat 8. Σύμφωνα με τη βιβλιογραφία της USGS, συγκεκριμένες ραδιομετρικές τιμές του καναλιού υποδηλώνουν την πιθανότητα ύπαρξης συννέφου. Ακολουθεί πίνακας που παρουσιάζει τις ραδιομετρικές τιμές του καναλιού BQA για την ύπαρξη συννέφων.

Τιμή Εικονοστοιχείου	Ύπαρξη συννέφου
61440	Ναι
59424	Ναι
57344	Ναι
56320	Ναι
53248	Ναι
52256	Ναι
52224	Ναι
49184	Ναι
49152	Ναι
48128	Ίσως
45056	Ίσως
43040	Ίσως
39936	Ίσως
36896	Ίσως
36864	Ίσως
32768	Ίσως

Πίνακας 5.1: Τιμές εικονοστοιχείων καναλιού BQA για ύπαρξη συννέφου
<http://landsat.usgs.gov/L8QualityAssessmentBand.php>

16-bit Landsat 8 QA Band - Read bits from RIGHT to LEFT ← starting with Bit 0																
BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DESCRIPTION	Cloud Confidence		Cirrus Confidence		Snow/Ice Confidence		Vegetation Confidence		Reserved for cloud shadow		Water Confidence		Reserved	Terrain Occlusion	Dropped Frame	Designated Fill

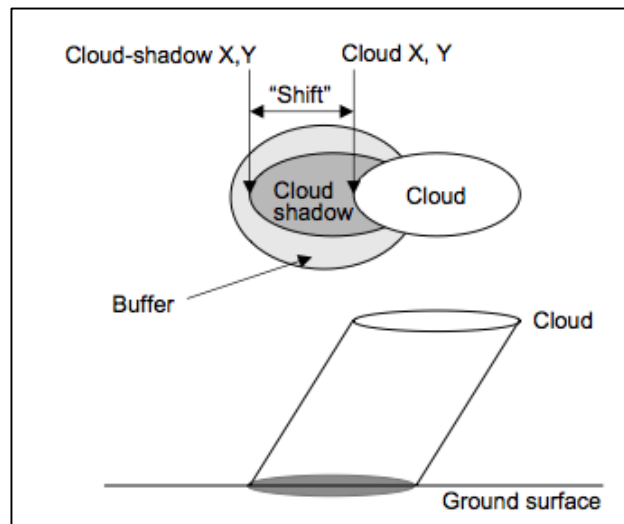
Πίνακας 5.2: Αριθμοί Bit καναλιού BQA για ύπαρξη συννέφου
<http://landsat.usgs.gov/L8QualityAssessmentBand.php>

Με τη χρήση του παραπάνω πίνακα είναι δυνατή η ανίχνευση των περιοχών που καλύπτονται από σύννεφα μέσω της δημιουργίας μιας μάσκας συννέφου ανάλογα με τις παραπάνω τιμές. Ωστόσο, δεν είναι δυνατή η παράλληλη ανίχνευση των αντίστοιχων σκιών τους καθώς δεν προσφέρονται αντίστοιχες ραδιομετρικές τιμές των εικονοστοιχείων για την περίπτωση αυτή.

- 2) Ένας άλλος τρόπος για την ανίχνευση των συννέφων είναι η εφαρμογή εξειδικευμένης ταξινόμησης με διαδικασίες μηχανικής μάθησης. Πιο συγκεκριμένα, είναι δυνατή η δημιουργία ενός στατιστικού μοντέλου αποκλειστικά για τις κατηγορίες των συννέφων και μετέπειτα η δημιουργία μιας μάσκας συννέφου, ανάλογα με τα εικονοστοιχεία που ταξινομήθηκαν στην κατηγορία σύννεφα. Ωστόσο, σημαντικό πρόβλημα της συγκεκριμένης μεθόδου είναι πως προκύπτουν γενικεύσεις στο διαχωρισμό των κατηγοριών, λόγω των αλγορίθμων ταξινόμησης, οπότε η συγκεκριμένη μέθοδος δεν παρουσιάζει ικανοποιητικά αποτελέσματα στις περισσότερες περιπτώσεις.
- 3) Οι δύο παραπάνω μέθοδοι δεν είναι ιδιαίτερα αποτελεσματικοί στην ανίχνευση συννέφων σε επίπεδο εικονοστοιχείου. Από δοκιμές σε διάφορα δεδομένα παρατηρήθηκε πως μεγάλο μέρος των συννέφων δεν ανιχνεύονται ιδίως σε περιοχές που τα σύννεφα δεν είναι ιδιαίτερα πυκνά. Στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας η ανίχνευση των συννέφων πραγματοποιήθηκε με τη χρήση του καναλιού 1 (Coastal Aerosol). Πιο συγκεκριμένα, στο κανάλι 1 οι περιοχές με σύννεφα είναι σε σημαντικό βαθμό πιο φωτεινές από ότι τα χαρακτηριστικά εδάφους. Επιπλέον, με τη δημιουργία του δείκτη $\frac{1}{4} \left(\frac{\text{Coastal Aerosol} - \text{Green}}{\text{Coastal Aerosol}} \right)$ και με την πρόσθεση του στο κανάλι Coastal Aerosol παρατηρείται πως η ανίχνευση των συννέφων γίνεται ακόμα πιο εύκολη. [Hallahan & Prepperneau, 2013]

Ανίχνευση Σκιών

Η ανίχνευση των σκιών αποτελεί μία από τις πιο δύσκολες διαδικασίες καθώς λόγω της μη συμμετρικότητας των συννέφων είναι δύσκολη η ανίχνευση των σκιών σε επίπεδο εικονοστοιχείου. Τις περισσότερες φορές η ανίχνευση των σκιών πραγματοποιείται λαμβάνοντας υπόψη το αζιμούθιο του ήλιου και δημιουργώντας μια ζώνη (buffer) ανάλογα με ένα προσεγγιστικό πλάτος της σκιάς από τα όρια του συννέφου.



Εικόνα 5.1: Ανίχνευση σκιών συννέφου με δημιουργία ζώνης
[Martinuzzi, Gould, & Gonzalez, 2011]

Όπως γίνεται αντιληπτό η συγκεκριμένη μέθοδος δεν ανιχνεύει αποτελεσματικά τις σκιές των συννέφων καθώς απαιτεί τον καθορισμό συγκεκριμένου πλάτους ζώνης από τον ερευνητή και έτσι το αποτέλεσμα αποκλείνει από την πραγματικότητα. Στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας, αναπτύχθηκε μέθοδος που ανιχνεύει τις σκιές, βασιζόμενος σε διαχρονικά τηλεπισκοπικά δεδομένα. Η παρουσίαση της μεθόδου ακολουθεί στην επόμενη ενότητα.

5.3.2 Ανάπτυξη Αλγορίθμου για την Ανίχνευση και Αφαίρεση Συννέφων και Σκιών με Βάση Διαχρονικά Γεωχωρικά Δεδομένα του Landsat 8

Με στόχο την ανίχνευση και αφαίρεση συννέφων και σκιών, αξιοποιώντας διαχρονικά τηλεπισκοπικά δεδομένα από το δορυφόρο Landsat 8, δημιουργήθηκε αλγόριθμος ο οποίος αξιοποιεί τα εξής δεδομένα:

- δορυφορική εικόνα που καλύπτεται από σύννεφα (εικόνα αναφοράς)
- δορυφορικές εικόνες με χρονική διαφορά ± 16 ημερών από την εικόνα αναφοράς ή εναλλακτικά δορυφορικές εικόνες που λήφθηκαν ακριβώς ένα χρόνο πριν ή ένα χρόνο μετά από την εικόνα αναφοράς

Η επιλογή δορυφορικών εικόνων διαφορετικού έτους, αλλά ίδιας εποχής, από την εικόνα αναφοράς (π.χ. καλοκαίρι) είναι πολύ σημαντική, έτσι ώστε να μην δημιουργηθούν προβλήματα φαινολογίας. Έχοντας, λοιπόν, ολοκληρώσει τη συλλογή των δεδομένων, επιλέχθηκαν οι δορυφορικές εικόνες με τη μικρότερη κάλυψη συννέφων, οι οποίες συγχρόνως δεν καλύπτονταν από σύννεφα στις περιοχές που καλυπτόταν η εικόνα αναφοράς.

Για την ανίχνευση των συννέφων δημιουργήθηκε το κανάλι Coastal Aerosol + $\frac{1}{4} \left(\frac{\text{Coastal Aerosol} - \text{Green}}{\text{Coastal Aerosol}} \right)$ τόσο για την εικόνα αναφοράς όσο και για την εικόνα χωρίς

σύννεφα στις περιοχές ενδιαφέροντος. Παρατηρήθηκε πως οι ραδιομετρικές τιμές στις περιοχές με σύννεφα της εικόνας αναφοράς ήταν πολύ μεγαλύτερες (μεγαλύτερες των 0.1 τιμών ανακλαστικότητας) από τις αντίστοιχες ραδιομετρικές τιμές στις περιοχές της εικόνας χωρίς σύννεφα, το οποίο δεν οφείλεται σε τυχαίο γεγονός αν ληφθεί υπόψη πως οι εικόνες που επιλέχθηκαν λήφθηκαν την ίδια περίοδο.

Για τον παραπάνω λόγο, τέθηκε ένα κατώφλι έτσι ώστε να ξεχωρίσουν οι ραδιομετρικές τιμές που υποδηλώνουν την ύπαρξη συννέφου στην εικόνα αναφοράς. Η τιμή του κατωφλίου αυτού ισούται με 0.1, αφού ύστερα από παρατηρήσεις, η τιμή αυτή θεωρήθηκε το ελάχιστο δυνατό όριο διαφοράς μεταξύ της ραδιομετρικής τιμής της εικόνας με σύννεφα και της αντίστοιχης χωρίς σύννεφα. Ωστόσο, πρέπει να τονιστεί πως το συγκεκριμένο κατώφλι δεν είναι μοναδικό και πρέπει να προσαρμοστεί σε ορισμένες περιπτώσεις ανάλογα με την περιοχή μελέτης. Επιπλέον, ανιχνεύθηκε η τιμή εκείνη του καναλιού για την οποία υπήρχαν σίγουρα σύννεφα. Ύστερα από παρατηρήσεις έγινε αντιληπτό πως τα σύννεφα στο κανάλι που δημιουργήθηκε είχαν τιμή ανακλαστικότητας μεγαλύτερη ή ίση του 0.2.

Για την ανίχνευση των σκιών των συννέφων παρατηρήθηκε πως οι ραδιομετρικές τιμές στο κανάλι 6 (SWIR 1) στις δορυφορικές εικόνες με σύννεφα ήταν πολύ μικρότερες (μικρότερες των 0.1 τιμών ανακλαστικότητας) από τις αντίστοιχες ραδιομετρικές τιμές της εικόνας χωρίς σύννεφα. Το παραπάνω συμπέρασμα δεν οφείλεται σε τυχαίο γεγονός καθώς αφενός οι εικόνες που επιλέχθηκαν λήφθηκαν την ίδια περίοδο και αφετέρου η μία από τις δύο εικόνες καλύπτεται από σύννεφα σε συγκεκριμένες περιοχές. Για τον παραπάνω λόγο τέθηκε ένα κατώφλι έτσι ώστε να ξεχωρίσουν οι ραδιομετρικές τιμές που υποδηλώνουν την ύπαρξη σκιάς συννέφου στην εικόνα αναφοράς. Η τιμή του κατωφλίου αυτού ισούται με 0.1, αφού ύστερα από παρατηρήσεις, αυτή θεωρήθηκε το ελάχιστο δυνατό όριο διαφοράς μεταξύ της ραδιομετρικής τιμής της εικόνας χωρίς σύννεφα και της αντίστοιχης με σύννεφα. Ωστόσο, πρέπει να τονιστεί πως το συγκεκριμένο κατώφλι δεν είναι μοναδικό και πρέπει να προσαρμοστεί σε ορισμένες περιπτώσεις ανάλογα με την περιοχή μελέτης. Επιπλέον, ανιχνεύθηκε η τιμή εκείνη του καναλιού για την οποία υπήρχαν σκιές συννέφου στην εικόνα. Ύστερα από παρατηρήσεις έγινε αντιληπτό πως οι σκιές στο κανάλι που 6 είχαν τιμή ανακλαστικότητας μικρότερη του 0.1.

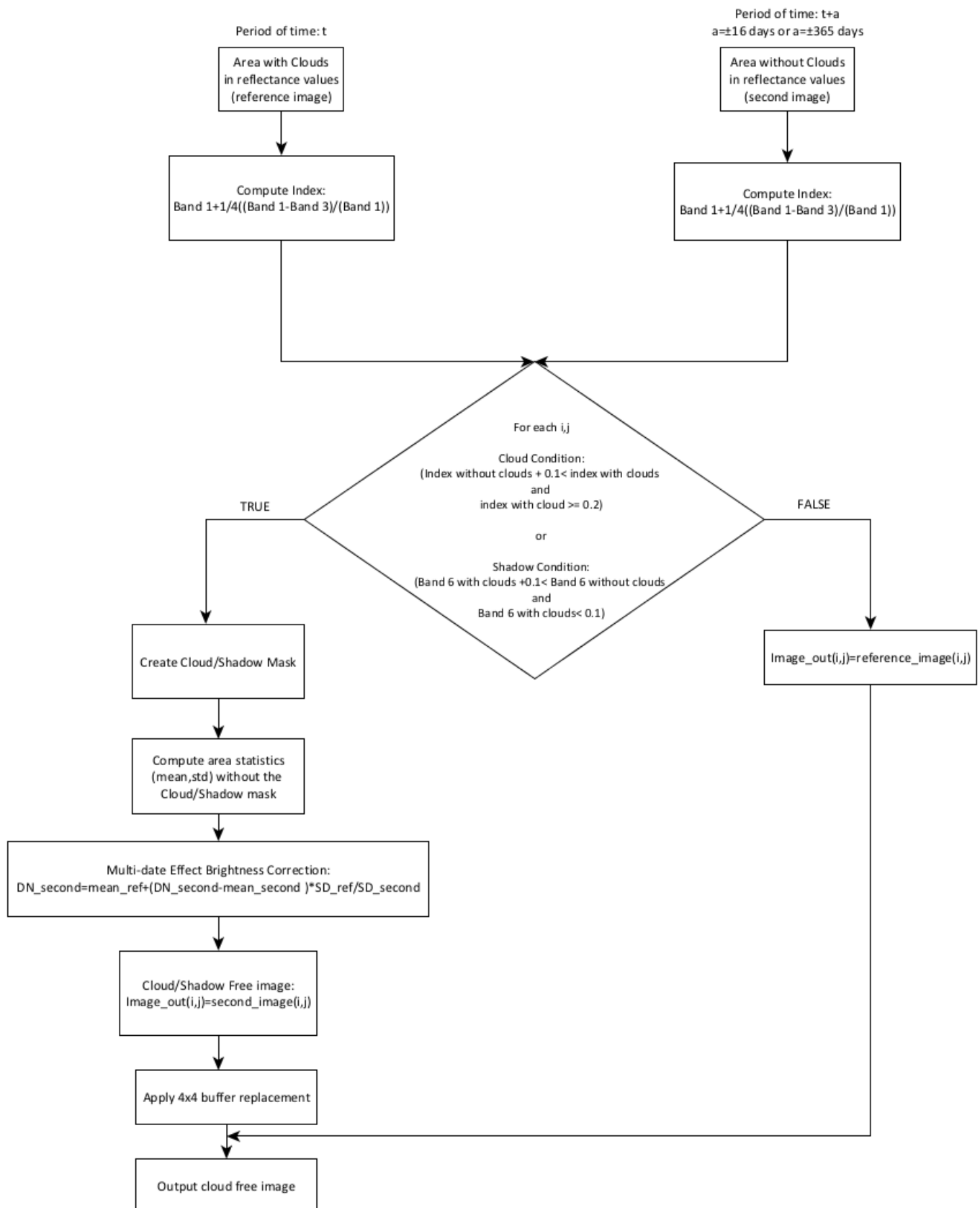
Για την αφαίρεση των συννέφων και των σκιών τους καθώς και για την αναδόμηση της εικόνας αναφοράς ακολουθήθηκε η εξής διαδικασία:

- Δημιουργία εικόνας με τις ραδιομετρικές τιμές της εικόνας αναφοράς
- Υπολογισμός των στατιστικών της εικόνας αναφοράς χωρίς να ληφθούν υπόψη τα εικονοστοιχεία που χαρακτηρίστηκαν ως σύννεφα ή ως σκιές συννέφων
- Διόρθωση των τιμών της δεύτερης εικόνας λαμβάνοντας υπόψη τις διαφορετικές συνθήκες λήψης της εικόνας αναφοράς και της δεύτερης εικόνας. Για το σκοπό αυτό χρησιμοποιήθηκε η στατιστική προσέγγιση του *Caselles (1989)* μέσω της παρακάτω εξίσωσης:

$$DN_{second} = mean_{ref} + (DN_{second} - mean_{second}) * SD_{ref} / SD_{second}$$

- Αντικατάσταση των ραδιομετρικών τιμών της εικόνας αναφοράς με τις αντίστοιχες ραδιομετρικές τιμές της εικόνας χωρίς σύννεφα όταν η διαφορά της ραδιομετρικής τιμής του καναλιού Coastal Aerosol + $\frac{1}{4} \left(\frac{Coastal\ Aerosol - Green}{Coastal\ Aerosol} \right)$ της πρώτης εικόνας είναι μεγαλύτερη του 0.1 από την αντίστοιχη τιμή της δεύτερης εικόνας και η ραδιομετρική τιμή του καναλιού με σύννεφα είναι μεγαλύτερη ή ίση του 0.2
- Αντικατάσταση των ραδιομετρικών τιμών της εικόνας αναφοράς με τις αντίστοιχες ραδιομετρικές τιμές της εικόνας χωρίς σύννεφα όταν η διαφορά της ραδιομετρικής τιμής του καναλιού 6 της δεύτερης εικόνας είναι μεγαλύτερη του 0.1 από την αντίστοιχη τιμή της πρώτης εικόνας και η ραδιομετρική τιμή του καναλιού με σκιά είναι μεγαλύτερη ή ίση του 0.01
- Παράλληλη αντικατάσταση των τεσσάρων περιμετρικών εικονοστοιχείων κάθε εικονοστοιχείου που βρίσκεται στο όριο της σκιάς των συννέφων με τα αντίστοιχα εικονοστοιχεία της εικόνας χωρίς σύννεφα, έτσι ώστε να μην δημιουργηθούν «κενά» στη νέα εικόνα και να υπάρξει ένα είδος ομαλοποίησης του τελικού αποτελέσματος

Στη συνέχεια, ακολουθεί το λογικό διάγραμμα του αλγορίθμου που περιγράφηκε παραπάνω, ενώ ο ίδιος ο αλγόριθμος παρουσιάζεται στο Παράρτημα II με ονομασία «Αλγόριθμος Ανίχνευσης και Αφαίρεσης των Συννέφων και των Σκιών τους σε Δορυφορικές Εικόνες Landsat 8» και αναπτύχθηκε στη γλώσσα προγραμματισμού C.



Διάγραμμα 5.2: Λογικό διάγραμμα αλγορίθμου ανίχνευσης και αφαίρεσης των συννέφων και των σκιών τους σε δορυφορικές εικόνες Landsat 8

5.4 Ταξινόμηση

Όσον αφορά τη διαδικασία της ταξινόμησης δόθηκε ιδιαίτερη προσοχή στην επιλογή των κατάλληλων δεδομένων εκπαίδευσης, των κατάλληλων καναλιών που θα χρησιμοποιηθούν στον αλγόριθμο ταξινόμησης καθώς και στην επιλογή του κατάλληλου αλγορίθμου, ο οποίος θα αντιμετώπιζε αποτελεσματικά την ταξινόμηση των κατηγοριών χρησιμοποιώντας διαδικασίες μηχανικής μάθησης. Πιο συγκεκριμένα, έχοντας ως στόχο την ταξινόμηση περιοχών μεγάλης έκτασης, δεν κρίθηκε αποδοτική η απευθείας ταξινόμηση όλης της έκτασης για τους παρακάτω λόγους:

- δυσκολία στην εύρεση σωστών και αντιπροσωπευτικών πολυγώνων εκπαίδευσης
- δύσκολα διαχειρίσιμα δεδομένα, λόγω της μεγάλης υπολογιστικής ισχύς που απαιτείται
- δυσκολία στην ερμηνεία και αξιολόγηση των αποτελεσμάτων

Λαμβάνοντας υπόψη τα παραπάνω, κρίθηκε αναγκαία η εύρεση εναλλακτικού τρόπου αντιμετώπισης του προβλήματος της ταξινόμησης. Παρατηρήθηκε ότι μέσω της τμηματοποίησης της εικόνας σε υποπεριοχές, τη δημιουργία ενός αναλυτικού στατιστικού μοντέλου εκμάθησης και την εφαρμογή του συγκεκριμένου μοντέλου στο σύνολο της περιοχής, τα αποτελέσματα της ταξινόμησης βελτιώθηκαν σε σημαντικό βαθμό. Για τη δημιουργία του στατιστικού μοντέλου εκμάθησης ακολουθήθηκαν τα εξής βήματα:

- απομόνωση αντιπροσωπευτικών περιοχών που χαρακτηρίζονταν από όλες τις δυνατές κατηγορίες χρήσεων γης
- εφαρμογή αναλυτικής μη επιβλεπόμενης ταξινόμησης σε κάθε υποπεριοχή
- δημιουργία αντιπροσωπευτικών πολυγώνων εκπαίδευσης, λαμβάνοντας υπόψη τα αποτελέσματα της μη επιβλεπόμενης ταξινόμησης, τα έτοιμα διανυσματικά δεδομένα από τον ιστοχώρο του geodata.gov.gr καθώς και τη φωτοερμηνεία των υποπεριοχών.
- έλεγχος της φασματικής υπογραφής καθώς και των ελλείψεων στα διάφορα κανάλια για όλα τα πολύγωνα εκπαίδευσης
- επιλογή του κατάλληλου αλγορίθμου ταξινόμησης για τη δημιουργία του αποδοτικότερου μοντέλου εκμάθησης με βάση τα πολύγωνα εκπαίδευσης που δημιουργήθηκαν καθώς και τους διάφορους συνδυασμούς καναλιών

Όσον αφορά στην επιλογή των κατάλληλων καναλιών, παρατηρήθηκε πως το κανάλι 9 του Landsat 8 (Cirrus) δημιουργούσε προβλήματα στα αποτελέσματα των ταξινομήσεων, ανέξαρτητα από την επιλογή του αλγορίθμου ταξινόμησης. Για το σκοπό αυτό το κανάλι 9 δεν χρησιμοποιήθηκε στη διαδικασία ταξινόμησης και έτσι χρησιμοποιήθηκαν αποκλειστικά τα κανάλια 1-7. Επίσης, ύστερα από δοκιμές έγινε αντιληπτό πως η παράλληλη χρήση κανονικοποιημένων λόγων στη διαδικασία

ταξινόμησης βελτίωσε σε σημαντικό βαθμό τα τελικά αποτελέσματα. Πιο συγκεκριμένα, παρατηρήθηκε πως η ενσωμάτωση των λόγων NDVI και NDWI στα κανάλια για την ταξινόμηση βελτίωσε αισθητά την ακρίβεια της, ενώ παράλληλα παρατηρήθηκε καλύτερος διαχωρισμός μεταξύ των διάφορων τύπων καλλιεργειών και των δασικών εκτάσεων. Όσον αφορά την επιλογή του αλγορίθμου ταξινόμησης, χρησιμοποιήθηκε ο αλγόριθμος SVM, ο οποίος ύστερα από την εφαρμογή διάφορων αλγορίθμων εκμάθησης έδειξε πως αντιμετωπίζει πιο αποτελεσματικά το πρόβλημα της ταξινόμησης πετυχαίνοντας αφενός υψηλότερη ακρίβεια στο στατιστικό μοντέλο και αφετέρου υψηλότερη ακρίβεια στην ταξινόμηση της εικόνας.

5.5 Post Classification

Με την ολοκλήρωση της ταξινόμησης πραγματοποιήθηκαν διαδικασίες μετά-ταξινόμησης έτσι ώστε να εξομαλυνθεί ο τελικός ταξινομημένος χάρτης και να ελεγχθεί η ακρίβεια της ταξινόμησης. Πιο συγκεκριμένα, εξομαλύνθηκε η ταξινομημένη εικόνα μέσω του αλγορίθμου majority voting έτσι ώστε να αντιμετωπιστεί ο «θόρυβος» από τα εικονοστοιχεία που δεν ταξινομήθηκαν σωστά και να παραχθεί ένα πιο ικανοποιητικό αποτέλεσμα. Παράλληλα, δόθηκε προσοχή ώστε να επιλεγεί η κατάλληλη ακτίνα εφαρμογής της διαδικασίας εξομάλυνσης, ώστε να μην γενικευθούν σε μεγάλο βαθμό τα τελικά αποτελέσματα. Στη συνέχεια, πραγματοποιήθηκε ένωση των όμοιων κατηγοριών, καθώς κατά τη διαδικασία εκμάθησης δημιουργήθηκαν πολλαπλά πολύγωνα εκπαίδευσης που αντιπροσώπευαν την ίδια θεματική κατηγορία με σκοπό την ενίσχυση του στατιστικού μοντέλου εκμάθησης. Τέλος, δημιουργήθηκαν πολύγωνα ελέγχου αξιοποιώντας παράλληλα ποιοτική πληροφορία από το google earth προκειμένου να ελεγχθεί η ακρίβεια των αποτελεσμάτων της ταξινόμησης.

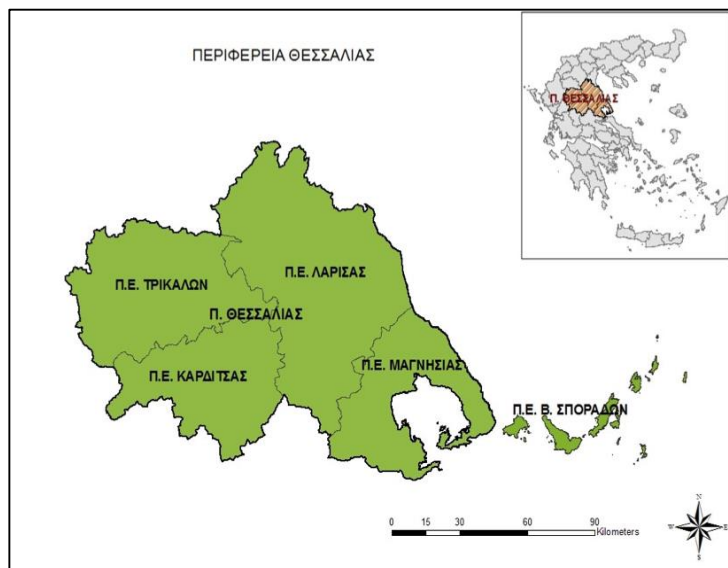
5.6 Ανίχνευση Μεταβολών

Στο τελικό στάδιο χρησιμοποιήθηκε η εικόνα της περιοχής μελέτης μετά από ένα χρόνο ακριβώς, έτσι ώστε να ανιχνευθούν οι πιθανές μεταβολές. Για το σκοπό αυτό χρησιμοποιήθηκε η τεχνική της post classification analysis, όπου σύμφωνα με αυτή ταξινομήθηκε η νέα εικόνα με το ίδιο στατιστικό μοντέλο εκπαίδευσης και στη συνέχεια ελέγχθηκαν οι διαφορές και οι μεταβολές στις κλάσεις μεταξύ των δύο περιόδων. Τέλος, ανιχνεύθηκαν το μέγεθος και τα είδη των μεταβολών, βάσει του αριθμού των εικονοστοιχείων που ταξινομήθηκαν σε διαφορετική κλάση από αυτήν της ταξινόμησης αναφοράς.

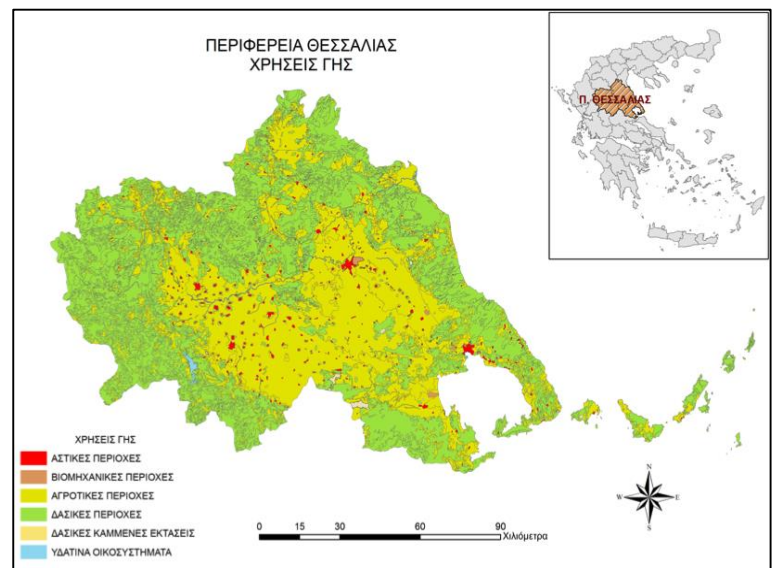
Κεφάλαιο 6: Ποσοτική και Ποιοτική Αξιολόγηση Αποτελεσμάτων

6.1 Επιλογή Περιοχής Μελέτης

Στο συγκεκριμένο κεφάλαιο εφαρμόστηκε η μεθοδολογία που περιγράφηκε στην προηγούμενη ενότητα για την ευρύτερη περιοχή της Θεσσαλίας. Ο λόγος που επιλέχθηκε η περιοχή της Θεσσαλίας, ως περιοχή μελέτης, είναι γιατί συνδυάζει διαφορετικές και ποικίλες χρήσεις γης με έμφαση τόσο στο αστικό κομμάτι όσο και στη βλάστηση και στους υδάτινους πόρους. Πιο συγκεκριμένα, η περιοχή της Θεσσαλίας συγκροτείται τόσο από ορεινά όσο και από πεδινά και παράκτια τμήματα εμφανίζοντας συχνές εναλλαγές χρήσεων γης. Ακολουθούν χάρτες που απεικονίζουν τη γεωγραφική θέση της Περιφέρειας της Θεσσαλίας καθώς και τις σημαντικότερες χρήσεις γης.



Χάρτης 6.1: Γεωγραφική θέση Περιφέρειας Θεσσαλίας
Επιχειρησιακό σχέδιο Περιφέρειας Θεσσαλίας, 2011



Χάρτης 6.2: Χρήσεις γης Περιφέρειας Θεσσαλίας
Επιχειρησιακό σχέδιο Περιφέρειας Θεσσαλίας, 2011

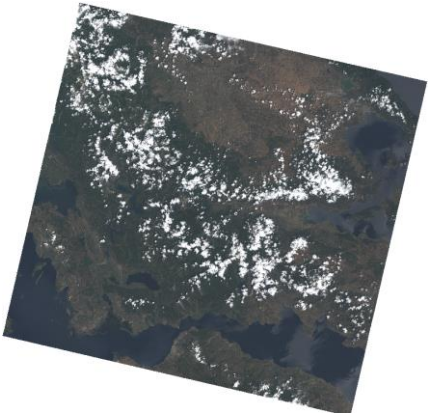
6.2 Επιλογή Δεδομένων

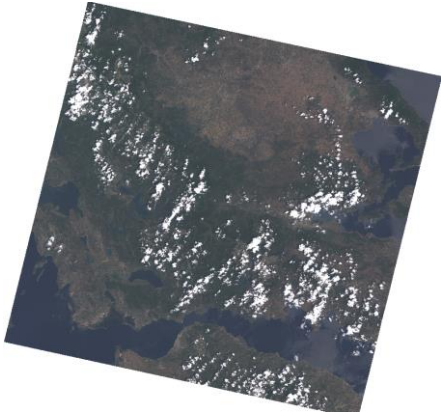
Για τη σωστή επιλογή των δεδομένων, αρχικά, ήταν απαραίτητο να επιλεγεί η ιδανική εποχή έτσι ώστε τα δεδομένα να έχουν όσο το δυνατόν λιγότερα σύννεφα, ενώ παράλληλα, οι συνθήκες λήψης των δορυφορικών εικόνων να είναι κατάλληλες για την εφαρμογή διαδικασιών ταξινόμησης. Για το σκοπό αυτό, επιλέχθηκε ο μήνας Ιούνιος, ο οποίος εντάσσεται στην καλοκαιρινή περίοδο και η πιθανότητα ύπαρξης συννέφων είναι αρκετά περιορισμένη. Επιπλέον, τους καλοκαιρινούς μήνες μεγάλο μέρος των καλλιεργειών βρίσκεται σε κατάσταση προχωρημένης ανάπτυξης γεγονός που διευκολύνει την ανίχνευση τους και τη διερεύνηση των διαφορετικών τους τύπων.

Για την επιλογή των δεδομένων, αρχικά, βρέθηκε το μονοπάτι (path) και η στήλη (row) του δορυφόρου Landsat 8 που πραγματοποιεί λήψη σκηνών, οι οποίες καλύπτουν την περιοχή της Θεσσαλίας. Παρατηρήθηκε, πως οι καταλληλότερες

σκηνές για το σκοπό αυτό είναι οι: 184-032 και 184-033. Παράλληλα, στόχος ήταν η εύρεση των ιδανικών σκηνών από άποψη συννεφοκάλυψης του μήνα Ιουνίου των ετών 2013 και 2014 και τελικά η επιλογή των εικόνων αναφοράς. Για το σκοπό αυτό τέθηκε το ανώτατο όριο συννεφοκάλυψης ίσο με 20% (από τα μεταδεδομένα των εικόνων με ονομασία cloud cover) και έτσι ως εικόνες αναφοράς επιλέχθηκαν οι σκηνές με ημερομηνία 25 Ιουνίου 2013. Οι σκηνές αυτές έχουν μικρά ποσοστά συννέφων, χωρίς όμως να είναι εντελώς απαλλαγμένες από αυτά.

Προκειμένου, λοιπόν, να εφαρμοστεί ο αλγόριθμος ανίχνευσης και αφαίρεσης συννέφων και σκιών, που περιγράφηκε στο προηγούμενο κεφάλαιο, ήταν απαραίτητη η εύρεση περισσότερων εικόνων, οι οποίες αφενός να έχουν χρονική διαφορά ± 16 μέρες ή ± 365 μέρες από την εικόνα αναφοράς και αφετέρου ποσοστό συννεφοκάλυψης μικρότερο από 20%. Λαμβάνοντας υπόψη τους παραπάνω περιορισμούς επιλέχθηκαν για την εφαρμογή οι σκηνές που παρουσιάζονται στον επόμενο πίνακα.

							
Ημερομηνία: 8/6/13	Σκηνή: 184/33	Ημερομηνία: 25/6/13	Σκηνή: 184/32	Ημερομηνία: 25/6/13	Σκηνή: 184/33	Ημερομηνία: 10/7/13	Σκηνή: 184/32
RGB: 432	Cloud Cover: 12.7%	RGB: 432	Cloud Cover : 5.04%	RGB: 432	Cloud Cover : 0.83%	RGB: 432	Cloud Cover : 2.43%

					
Ημερομηνία: 10/7/13	Σκηνή: 184/33	Ημερομηνία: 11/6/14	Σκηνή: 184/32	Ημερομηνία: 11/6/14	Σκηνή: 184/33
RGB: 432	Cloud Cover: 2.11%	RGB: 432	Cloud Cover: 3.10%	RGB: 432	Cloud Cover: 7.57%

Πίνακας 6.1: Επιλεγμένες σκηνές Landsat 8

6.3 Προεργασία Δεδομένων

Στο κεφάλαιο αυτό πραγματοποιήθηκαν όλες οι απαραίτητες διαδικασίες προεργασίας των δεδομένων, σύμφωνα με τη μεθοδολογία που χρησιμοποιήθηκε.

6.3.1 Έλεγχος γεωμετρίας

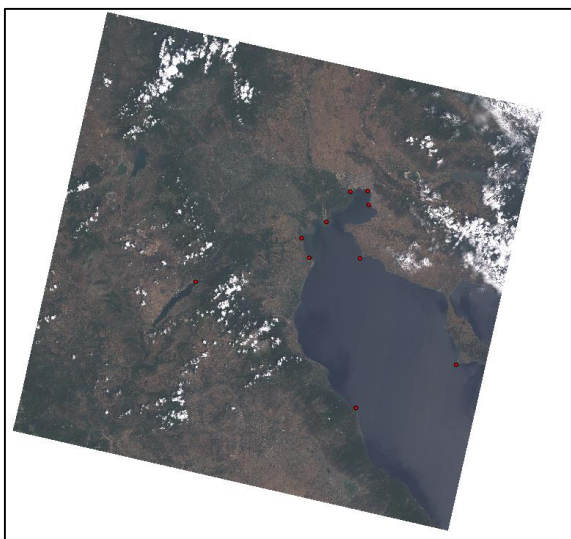
Στη συγκεκριμένη ενότητα πραγματοποιείται έλεγχος της γεωμετρίας των σκηνών 184/32 καθώς και των σκηνών 184/33 στις διάφορες χρονικές περιόδους που λήφθηκαν καθώς και έλεγχος της γεωμετρίας των επικαλυπτόμενων τμημάτων τους.

Για τον έλεγχο της γεωμετρίας ακολουθήθηκε η εξής μεθοδολογία:

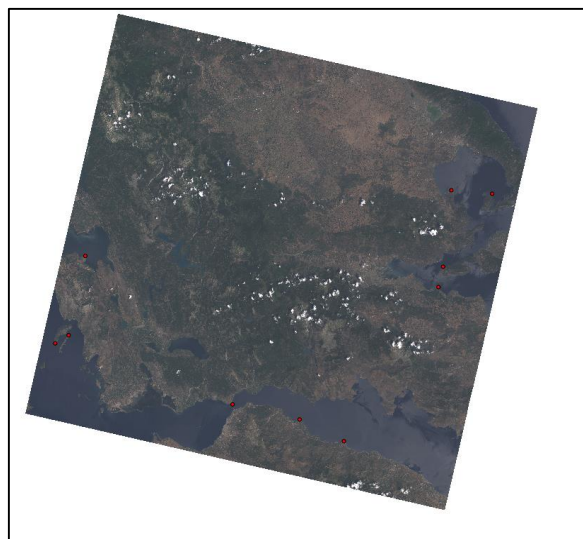
- ψηφιοποίηση 10 χαρακτηριστικών σημείων στην εικόνα αναφοράς (25/6/2013). Τα σημεία που επιλέχθηκαν ήταν ομοιόμορφα κατανομημένα σε όλη την έκταση της εικόνας, ενώ παράλληλα δεν βρίσκονταν στις άκρες της.
- εύρεση συντεταγμένων των παραπάνω 10 σημείων
- «προβολή» των σημείων στις άλλες εικόνες και έλεγχος μέσου σφάλματος

Από την εφαρμογή της παραπάνω μεθοδολογίας παρατηρήθηκαν σφάλματα μικρότερα του 1/3 του εικονοστοιχείου κατά τον άξονα Χ και Υ. Μάλιστα, τα σφάλματα διατηρήθηκαν ίδια όταν τα σημεία προβλήθηκαν και στις άλλες εικόνες, κάτι που αποδεικνύει πως πρόκειται για σφάλματα που προέκυψαν κατά τη διαδικασία της ψηφιοποίησης και δεν μπορούσαν να αποφευχθούν λόγω της διακριτικής ικανότητας του χρήστη. Ακολουθούν τα σημεία που χρησιμοποιήθηκαν σε κάθε περίπτωση καθώς και τα αναλυτικά σφάλματα ανά σημείο. Η παραπάνω διαδικασία πραγματοποιήθηκε στο λογισμικό ανοιχτού κώδικα QGIS.

Έλεγχος σκηνών 184/32, 184/33



Εικόνα 6.1: Σημεία ελέγχου για τις σκηνές 184/32



Εικόνα 6.2: Σημεία ελέγχου για τις σκηνές 184/33

on/off	id	srcX	srcY	dstX	dstY	dX[pixels]	dY[pixels]	residual[pixels]
☒	0	660292.35	4470942.35	660293.50	4470941.37	-0.25	0.10	0.27
☒	1	701234.55	4425593.40	701267.87	4425608.61	0.09	-0.04	0.10
☒	2	635318.89	4479577.11	635330.60	4479563.85	0.31	0.32	0.45
☒	3	638649.33	4471172.76	638648.26	4471172.14	-0.24	-0.05	0.24
☒	4	593594.90	4461189.06	593600.13	4461178.33	0.02	0.03	0.03
☒	5	590152.69	4460969.51	590156.98	4460963.45	-0.00	-0.15	0.15
☒	6	657138.71	4497578.35	657137.95	4497580.39	0.04	-0.10	0.11
☒	7	663585.56	4499657.43	663584.92	4499661.35	0.04	-0.13	0.13
☒	8	656161.46	4499402.86	656158.27	4499402.54	-0.02	-0.03	0.04
☒	9	663958.48	4493759.39	663958.71	4493758.60	-0.00	0.05	0.05

Transform: Polynomial 1 Mean error: 0.238403

Πίνακας 6.2: Έλεγχος ακρίβειας σημείων ελέγχου για σκηνές 184/32

on/off	id	srcX	srcY	dstX	dstY	dX[pixels]	dY[pixels]	residual[pixels]
☒	0	682223.81	4336677.71	682213.13	4336672.72	-0.09	0.06	0.11
☒	1	567382.16	4242669.34	567354.21	4242680.34	-0.12	-0.30	0.33
☒	2	494799.49	4273530.19	494799.38	4273538.76	0.29	-0.22	0.37
☒	3	488850.59	4269898.11	488842.40	4269895.13	0.02	0.18	0.18
☒	4	597004.14	4235938.56	597007.05	4235940.84	0.19	-0.02	0.19
☒	5	616623.41	4226225.30	616616.67	4226219.17	-0.18	0.26	0.31
☒	6	660510.71	4304112.78	660501.70	4304122.08	-0.10	-0.36	0.38
☒	7	664170.19	4338242.42	664167.35	4338240.22	0.19	-0.02	0.20
☒	8	502177.98	4308928.78	502157.57	4308924.83	-0.30	0.16	0.34
☒	9	658470.26	4295017.47	658467.14	4295008.10	0.08	0.27	0.28

Transform: Polynomial 1 Mean error: 0.336369

Πίνακας 6.3: Έλεγχος ακρίβειας σημείων ελέγχου για σκηνές 184/33

Από τους παραπάνω δύο ελέγχους καθίσταται σαφές πως οι διάφορες σκηνές του δορυφόρου Landsat 8 είναι άρτια συσχετισμένες και δεν απαιτείται η εφαρμογή γεωμετρικού μετασχηματισμού για εφαρμογές που απαιτούν τη χρήση διαχρονικών δεδομένων. Στη συνέχεια, πραγματοποιείται έλεγχος της γεωμετρίας για το επικαλυπτόμενο τμήμα των δύο παραπάνω σκηνών.

Έλεγχος επικαλυπτόμενων τμημάτων

Στους προηγούμενους ελέγχους των σκηνών χρησιμοποιήθηκαν σημεία ελέγχου, τα οποία δεν άνηκαν στο επικαλυπτόμενο τμήμα τους, έτσι ώστε να ελεγχθεί ξεχωριστά η γεωμετρική ακρίβεια των επικαλυπτόμενων περιοχών. Ο έλεγχος των συγκεκριμένων περιοχών είναι πολύ σημαντικός καθώς αποτελεί ένδειξη για το κατά πόσο η δημιουργία του μωσαικού είναι εφικτή χωρίς να είναι απαραίτητη η συσχέτιση των σκηνών. Για το σκοπό αυτό ελέγχθηκε αποκλειστικά η συσχέτιση των δύο σκηνών σε μία χρονική στιγμή και όχι σε διαχρονικά δεδομένα καθώς οι παραπάνω δύο έλεγχοι που προηγήθηκαν εξασφάλιζαν τη συγκεκριμένη συνθήκη.



Εικόνα 6.3: Σημεία ελέγχου για το επικαλυπτόμενο τμήμα

on/off	id	srcX	srcY	dstX	dstY	dX[pixels]	dY[pixels]	residual[pixels]
☒	0	648179.44	4379643.21	648179.36	4379635.84	0.03	0.11	0.11
☒	1	647130.79	4378904.87	647132.74	4378907.11	0.12	-0.19	0.22
☒	2	656100.52	4368198.97	656082.88	4368203.62	-0.23	-0.19	0.30
☒	3	623917.73	4390015.61	623924.73	4390014.09	0.02	0.02	0.03
☒	4	627749.42	4390402.61	627752.99	4390400.46	-0.12	-0.01	0.12
☒	5	660016.03	4371032.83	660004.07	4371025.61	-0.14	0.11	0.18
☒	6	672692.61	4378976.21	672695.66	4378958.47	0.07	0.18	0.19
☒	7	677607.73	4376283.24	677603.56	4376275.87	-0.10	-0.18	0.21
☒	8	656883.45	4369609.66	656881.30	4369599.94	0.24	0.25	0.35
☒	9	655154.86	4375035.73	655154.24	4375034.22	0.13	-0.09	0.16

Transform: Polynomial 1 Mean error: 0.227753

Πίνακας 6.4: Έλεγχος ακρίβειας σημείων ελέγχου για επικαλυπτόμενο τμήμα

6.3.2 Εφαρμογή ραδιομετρικών διορθώσεων

Έχοντας ολοκληρώσει τον έλεγχο της γεωμετρίας των διαχρονικών δεδομένων του Landsat 8, πραγματοποιήθηκε η μετατροπή των ψηφιακών τιμών σε τιμές φαινόμενης ανακλαστικότητας χρησιμοποιώντας το πρόγραμμα που περιγράφηκε στην ενότητα 5.2.3. Η εφαρμογή των ραδιομετρικών διορθώσεων εφαρμόστηκε σε όλες τις σκηνές που επιλέχθηκαν για τη διαδικασία ανάλυσης και για τα κανάλια 1-8. Ο χρόνος εφαρμογής των ραδιομετρικών διορθώσεων για τα 7 πρώτα κανάλια του Landsat 8, σε τετραπύρηνο υπολογιστή, i7 2,6 Ghz, με μνήμη 10 GB, ήταν 18 δευτερόλεπτα.

Το πρόγραμμα που δημιουργήθηκε δέχεται ως δεδομένα εισόδου την πολυφασματική εικόνα σε μορφή ERS, τον αριθμό των σειρών και των στηλών της, τον αριθμό των καναλιών καθώς και τη γωνία ανύψωσης ήλιου (Sun_Elevation) από τα μεταδεδομένα της. Σαν έξοδο έχει την πολυφασματική εικόνα με τιμές φαινόμενης ανακλαστικότητας σε μορφή ERS. Για τη δημιουργία της εικόνας αυτής καθώς και για τη μετατροπή της από μορφή Gtiff σε ERS χρησιμοποιήθηκε η βιβλιοθήκη gdal μέσω γραμμών εντολών στο τερματικό. Μετά την εφαρμογή του προγράμματος προσαρμόστηκε το αρχείο επικεφαλίδας (header) στην ραδιομετρικά διορθωμένη εικόνα. Ακολουθεί η διαδικασία για την εφαρμογή του προγράμματος σε περιβάλλον Unix.

Δημιουργία πολυφασματικής εικόνας σε φορμάτ ERS:

```
gdalmerge.py -n 0 -a_nodata 0 -separate -of ERS -o "Mult_Band.ers" "Band 1.tif" "Band 2.tif" "Band 3.tif" "Band 4.tif" "Band 5.tif" "Band 6.tif" "Band 7.tif"
```

Μετατροπή ψηφιακών τιμών σε τιμές φαινόμενης ανακλαστικότητας:

```
./DN2TOA_REFL Mult_Band 7371 7581 66.23572837 Mult_Band_Toa_Refl
```

Για την προσαρμογή του αρχείου επικεφαλίδας στην ραδιομετρικά διορθωμένη εικόνα επεξεργάστηκε η επικεφαλίδα της αρχικής εικόνας με κατάληξη .ers με πρόγραμμα επεξεργασίας κειμένου, αλλάζοντας το κελί Name με το αντίστοιχο όνομα της ραδιομετρικά διορθωμένης εικόνας καθώς και το κελί CellType με το νέο τύπο που ήταν IEEE4ByteReal, δηλαδή τύπου πραγματικών αριθμών (float). Η ίδια διαδικασία ακολουθήθηκε και για το κανάλι 8 που δεν περιηγήθηκε στην πολυφασματική εικόνα λόγω των διαφορετικών χαρακτηριστικών του.

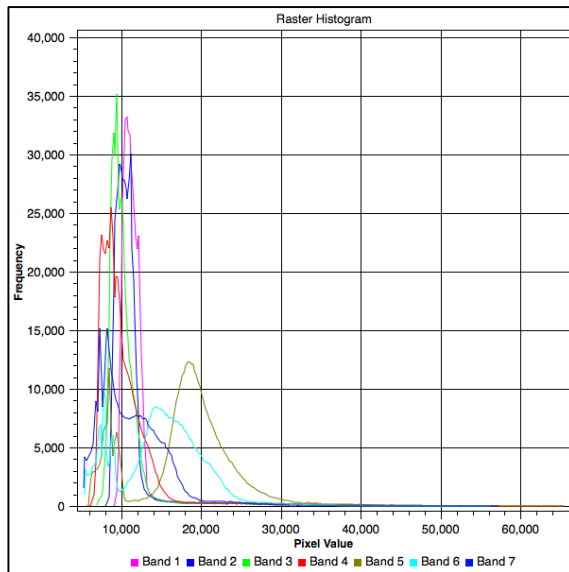
Επεξήγηση πηγαίου κώδικα

Στη γραμμή 40 και 41 του κώδικα στο Παράρτημα II με ονομασία «Μετατροπή Ψηφιακών Τιμών σε Τιμές Φαινόμενης Ανακλαστικότητας για τον Δέκτη OLI του Δορυφόρου Landsat 8 σε γλώσσα C» πραγματοποιείται η δήλωση των μεταβλητών για την εικόνα εισόδου και την εικόνα εξόδου. Πιο συγκεκριμένα, για την εικόνα εισόδου επιλέχτηκε μεταβλητή τύπου unsigned short int, η οποία δεσμεύει χώρο 2 bytes, καθώς η εικόνα εισόδου είναι 16 bit unsigned int. Για την εικόνα εξόδου, χρησιμοποιείται μεταβλητή τύπου πραγματικού αριθμού float, καθώς οι τελικές τιμές φαινόμενης ανακλαστικότητας εκφράζονται σε πραγματικούς αριθμούς. Στις γραμμές 50-53 δηλώνονται οι μεταβλητές που απαιτούνται ως είσοδο από τον χρήστη και βρίσκονται στα μεταδεδομένα της εικόνας.

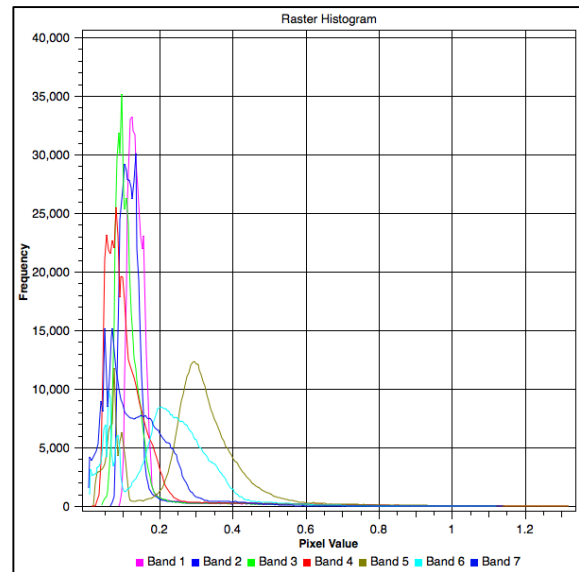
Στη γραμμή 62 πραγματοποιείται αναδρομική ανάγνωση της εικόνας εισόδου σε συνεχή μνήμη δεσμεύοντας χώρο που ισούται με τον αριθμό των γραμμών επί των στηλών επί των καναλιών της εικόνας. Όμοια, στη γραμμή 79 δεσμεύεται μνήμη για την εικόνα εξόδου δεσμεύοντας πάλι συνεχή μνήμη, έτσι ώστε να είναι πιο εύκολο το «γράψιμο» της εικόνας από την μνήμη στον αποθηκευτικό χώρο του υπολογιστή.

Στις γραμμές 85-87 πραγματοποιείται η προσπέλαση της εικόνας σε μορφή BIL, δηλαδή προσπέλαση ανά σειρά, ανά κανάλι και τέλος ανά στήλη. Στη συνέχεια, στις γραμμές 89-93 πραγματοποιείται η μετατροπή των ψηφιακών τιμών σε τιμές φαινόμενης ανακλαστικότητας και μηδενίζονται οι αρνητικές τιμές που προέκυψαν από την αντίστοιχη αριθμητική πράξη. Τέλος, στις γραμμές 101-105 πραγματοποιείται το «γράψιμο» της εικόνας στο αρχείο που επέλεξε ο χρήστης και αποδεσμεύεται ο χώρος της μνήμης που χρησιμοποιήθηκε για την ανάγνωση και τη δημιουργία των εικόνων.

Ακολουθεί παράδειγμα από το ιστόγραμμα της αρχικής εικόνας καθώς και της ραδιομετρικά διορθωμένης εικόνας για την ημερομηνία 25 Ιουνίου 2013, path: 184 και row: 32.



Διάγραμμα 6.1: Αρχικό ιστόγραμμα εικόνας



Διάγραμμα 6.2: Ιστόγραμμα εικόνας με τιμές ανακλαστικότητας

Στη συνέχεια, παρουσιάζονται τα στατιστικά της αρχικής και της τελικής εικόνας σε μορφή πίνακα:

Χαρακτηριστικό/Κανάλι	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση
1	8464	64783	11919.51	3726.712
2	7547	65535	11191.846	3923.746
3	6359	65535	10678.725	3932.016
4	5513	65535	10420.954	4439.835
5	5304	65535	17957.360	6339.283
6	4697	65535	14947.309	5261.213
7	4929	64726	11209.160	4001.657
8	0	65535	7312.7	5947.45

Πίνακας 6.5: Στατιστικά εικόνας αναφοράς πριν την εφαρμογή των ραδιομετρικών διορθώσεων

Χαρακτηριστικό/Κανάλι	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση
1	0.076	1.306	0.151	0.081
2	0.056	1.323	0.135	0.086
3	0.030	1.323	0.124	0.086
4	0.011	1.323	0.118	0.097
5	0.007	1.323	0.283	0.139

6	0.001	1.323	0.217	0.115
7	0.001	1.305	0.136	0.087
8	0	1.323	0.084	0.094

Πίνακας 6.6: Στατιστικά εικόνας αναφοράς μετά την εφαρμογή των ραδιομετρικών διορθώσεων

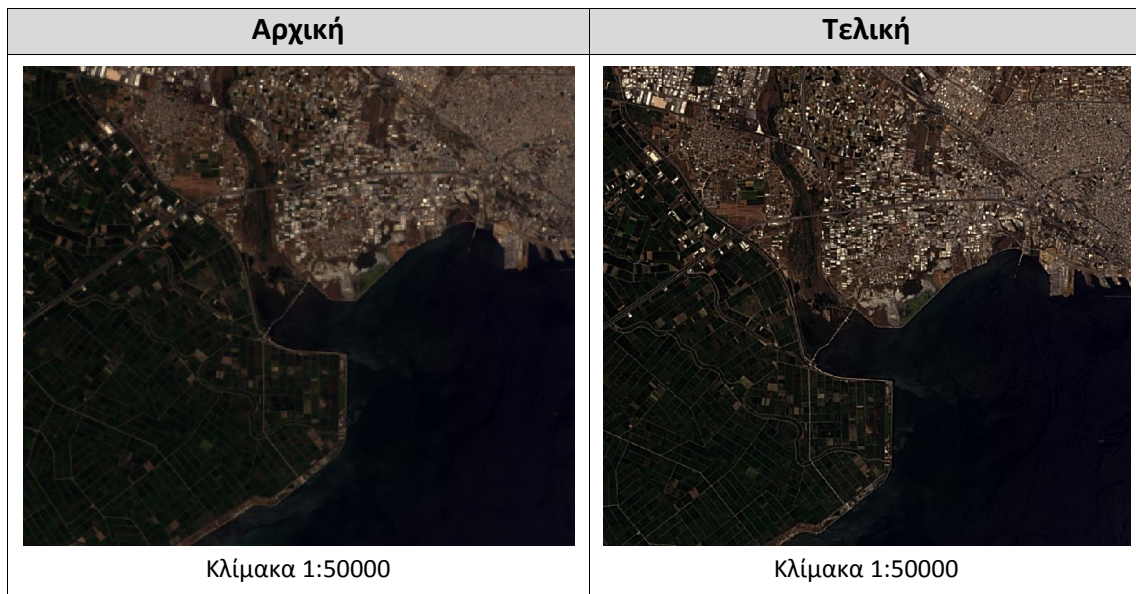
6.3.3 Εφαρμογή pansharpening

Στη συνέχεια, εφαρμόστηκε ο αλγόριθμος BundleToPerfectSensor του orfeotoolbox που πραγματοποιεί τη διαδικασία του pansharpening όπως αναλύθηκε στο κεφάλαιο 5.2.4. Για την εφαρμογή του αλγορίθμου χρησιμοποιήθηκε η ραδιομετρικά διορθωμένη πολυφασματική εικόνα που δημιουργήθηκε καθώς και το αντίστοιχο κανάλι 8, που ήταν το πανχρωματικό. Ο χρόνος εφαρμογής του αλγορίθμου, σε τετραπύρηνο υπολογιστή, i7 2,6 Ghz, με μνήμη 10 GB, ήταν 4 λεπτά.

Ακολουθεί παράδειγμα από την εφαρμογή του αλγορίθμου για την ημερομηνία 25 Ιουνίου 2013, path: 184 και row: 32.

Διαδικασία Pansharpening:

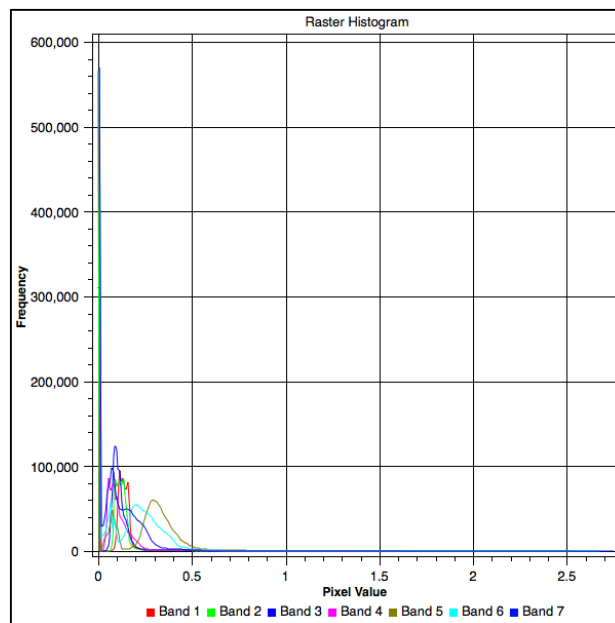
```
otbcli_BundleToPerfectSensor -inp Band_8_Refl.ers -inxs Mult_Band_Refl.ers -out Mult_Pan_Refl.tif
```





Πίνακας 6.7: Παράδειγμα εφαρμογής αλγορίθμου pansharpening του orfeotoolbox

Στη συνέχεια, ακολουθεί το ιστόγραμμα της μετασχηματισμένης εικόνας καθώς και τα αντίστοιχα στατιστικά της.



Διάγραμμα 6.3: Ιστόγραμμα εικόνας μετά την εφαρμογή του pansharpening

Χαρακτηριστικό/Κανάλι	Ελάχιστη Τιμή	Μέγιστη Τιμή	Μέση Τιμή	Τυπική Απόκλιση
1	0	3.869	0.105	0.099
2	0	3.513	0.094	0.097
3	0	3.596	0.086	0.094
4	0	4.395	0.083	0.1
5	0	5.202	0.196	0.175
6	0	4.847	0.151	0.142
7	0	5.055	0.095	0.099

Πίνακας 6.8: Στατιστικά εικόνας αναφοράς μετά την εφαρμογή του pansharpening

6.4 Δημιουργία Μωσαικού

Στην ενότητα αυτή δημιουργήθηκε μωσαικό για την ευρύτερη περιοχή της Θεσσαλίας χρησιμοποιώντας τα δεδομένα από την ημερομηνία αναφοράς (25 Ιουνίου 2013). Για το σκοπό αυτό χρησιμοποιήθηκε η εντολή `gdalbuildvrt` της βιβλιοθήκης `gdal`. Αρχικά, συγχωνεύτηκαν οι δύο σκηνές (184/32 και 184/33) και στη συνέχεια κόπηκε το μωσαικό στην περιοχή της Θεσσαλίας. Δόθηκε προσοχή στην προσαρμογή των δύο σκηνών στο ίδιο εύρος ψηφιακών τιμών καθώς και στην κατάλληλη αντιμετώπιση του επικαλυπτόμενου τμήματος. Λόγω του γεγονότος πως το συγκεκριμένο μωσαικό χρησιμοποιήθηκε για διαδικασίες ταξινόμησης, στο επικαλυπτόμενο τμήμα επιλέχθηκε η διατήρηση των ψηφιακών τιμών της καλύτερης σκηνής. Πιο συγκεκριμένα, επιλέχθηκε η σκηνή 184/32, η οποία κάλυπτε μεγαλύτερο τμήμα των επικαλυπτόμενων περιοχών και στα άκρα της παρατηρήθηκαν μικρότερες παραμορφώσεις. Στη συνέχεια, παρουσιάζεται η διαδικασία για την κατασκευή του μωσαικού, το κόσψιμο αυτού στην περιοχή της Θεσσαλίας καθώς και το τελικό μωσαικό.

Ένωση σκηνών:

```
gdalbuildvrt Scenes_Mosaic.vrt -srcnodata 0 -vrtnodata 0 Mult_Pan_Refl_184_32.tif  
Mult_Pan_Refl_184_33.tif
```

Κόψιμο μωσαικού:

```
gdal_translate -projwin 546637.500 4419862.5 676897.500 4289722.5 -a_nodata 0 -of  
Gtiff Scenes_Mosaic.vrt Thes_Mosaic.tif
```



Εικόνα 6.4: Μωσαικό Θεσσαλίας για την ημερομηνία: 25 Ιουνίου 2013, 9684x9676 εικον/χρεια, μέγεθος: 16 GB

6.5 Δημιουργία Μοντέλου Εκπαίδευσης

Έχοντας ολοκληρώσει την προεργασία των δεδομένων δημιουργήθηκε ένα στατιστικό μοντέλο εκπαίδευσης, το οποίο έχει τη δυνατότητα να ταξινομήσει την ευρύτερη περιοχή της Θεσσαλίας χωρίς να είναι απαραίτητη η δημιουργία εξειδικευμένων πολυγώνων εκπαίδευσης για κάθε περιοχή. Για το σκοπό αυτό επιλέχθηκαν τέσσερις υποπεριοχές που κάλυπταν όλες τις δυνατές κατηγορίες χρήσεων γης. Πιο συγκεκριμένα, η πρώτη υποπεριοχή συνδύαζε τον αστικό ιστό με τους διάφορους τύπους καλλιεργειών, η δεύτερη υποπεριοχή τους υδατικούς πόρους με την αραιή βλάστηση, η τρίτη υποπεριοχή τους ορεινούς όγκους με την πυκνή βλάστηση και η τέταρτη υποπεριοχή συνδύαζε τους επίγειους υδατικούς πόρους με όλες τις ανθρωπογενείς παρεμβάσεις. Οι υποπεριοχές που επιλέχθηκαν χαρακτηρίζονταν από καθόλου ή αραιή συννεφοκάλυψη, έτσι ώστε αν δεν ήταν δυνατή η αφαίρεση των συννέφων και των σκιών να πραγματοποιούνταν η ανίχνευσή τους μέσω της ταξινόμησης. Ακολουθούν οι τέσσερις υποπεριοχές που επιλέχθηκαν:



Εικόνα 6.5: Υποπεριοχή 1 – Εύρύτερη περιοχή λίμνης Κάρλας



Εικόνα 6.6: Υποπεριοχή 2 – Εύρύτερη περιοχή πόλης Λάρισας



Εικόνα 6.7: Υποπεριοχή 3 – Εύρύτερη περιοχή λίμνης Βεγορίτιδας

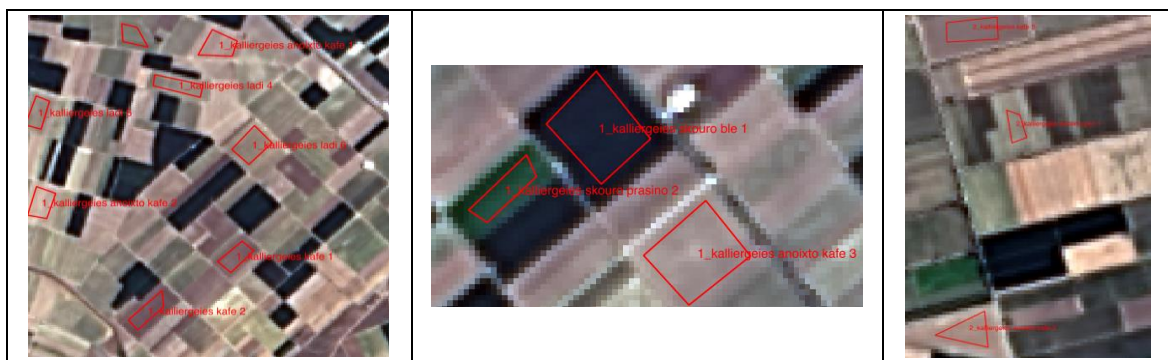


Εικόνα 6.8: Υποπεριοχή 4 – Εύρύτερη περιοχή λίμνης Τριχωνίδας

Όπως παρουσιάστηκε στο κεφάλαιο 6.4, για τη διαδικασία της ταξινόμησης επιλέχθηκαν τα κανάλια 1-7 του Landsat 8 και δημιουργήθηκαν επιπλέον οι κανονικοποιημένοι λόγοι NDVI και NDWI, από την εφαρμογή του orfeo toolbox. Στο Παράρτημα Ι με ονομασία «Orfeo Toolbox – Radiometric Indices» βρίσκεται ο κώδικας για τον υπολογισμό των κανονικοποιημένων λόγων. Για τη δημιουργία του στατιστικού μοντέλου εκμάθησης ακολουθήθηκε η εξής διαδικασία:

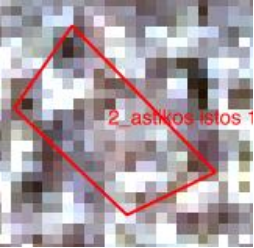
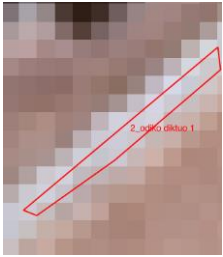
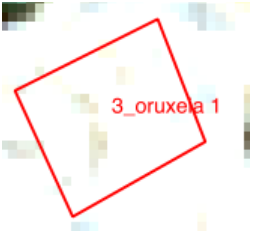
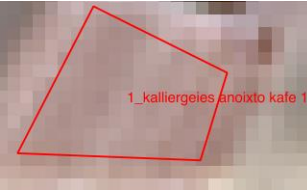
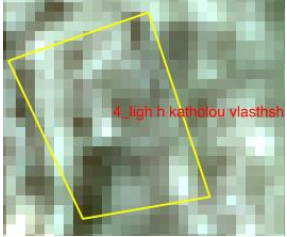
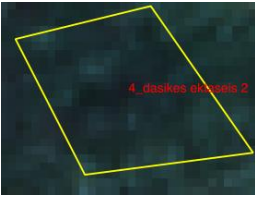
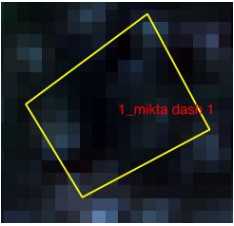
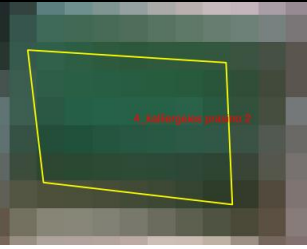
- μη επιβλεπόμενη ταξινόμηση (kmeans) 60 κλάσεων σε κάθε υποπεριοχή
- χαρακτηρισμός της κάθε κλάσης για κάθε υποπεριοχή
- δημιουργία πολυγώνων για κάθε υποπεριοχή με βάση τα αποτελέσματα της μη επιβλεπόμενης ταξινόμησης, τη φωτοερμηνεία από το google earth καθώς και τα πολύγωνα από το geodata.gov.gr
- έλεγχος του ιστογράμματος για κάθε πολύγωνο εκπαίδευσης (ώστε να μην είναι «πολυκόρυφο»)
- έλεγχος των φασματικών υπογραφών στα κανάλια που επιλέχθηκαν
- ανάλυση ευαισθησίας στα διάφορα πολύγωνα εκπαίδευσης, ώστε να χρησιμοποιηθούν τα ιδανικά

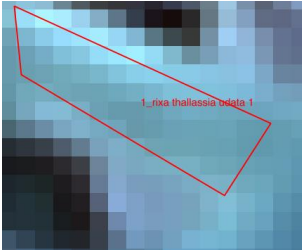
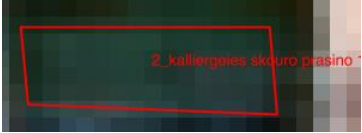
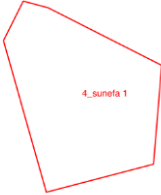
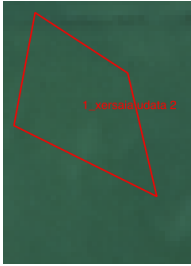
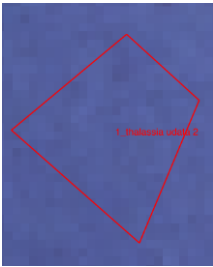
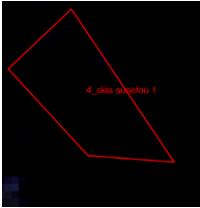
Για την καλύτερη εκπαίδευση του μοντέλου λήφθηκαν πολλαπλά δείγματα για κάθε κατηγορία και από κάθε υποπεριοχή, έτσι ώστε να ενισχυθεί η αξιοπιστία του στατιστικού μοντέλου. Επιπλέον, διαχωρίστηκαν οι διάφοροι τύποι καλλιεργειών δημιουργώντας ξεχωριστά πολύγωνα εκπαίδευσης για κάθε είδος. Η διάκριση των διάφορων τύπων καλλιεργειών πραγματοποιήθηκε τόσο με φωτοερμηνεία όσο και με εποπτεία των περιοχών από το υψηλής ανάλυσης δορυφορικό υπόβαθρο του google earth. Αξίζει να σημειωθεί πως η διάκριση των καλλιεργειών πραγματοποιήθηκε για την ημερομηνία που επιλέχθηκε ως ημερομηνία αναφοράς, ενώ σε κάθε άλλη περίοδο πιθανώς να απαιτούνταν διαφορετική διάκριση. Ακολουθούν κάποιες κατηγοριοποιήσεις καλλιεργειών που πραγματοποιήθηκαν.



Πίνακας 6.9: Παραδείγματα κατηγοριοποίησης τύπων καλλιεργειών

Στον παρακάτω πίνακα παρουσιάζονται οι κατηγορίες για τις οποίες δημιουργήθηκαν πολύγωνα εκπαίδευσης με τα αντίστοιχα παραδείγματα για κάθε κατηγορία.

		
Αστικός Ιστός	Πλατύφυλλα Δάση	Οδικό Δίκτυο
		
Χωμάτινο Οδικό Δίκτυο	Καλλιέργειες Τύπου 1 (ανοιχτό γκρι)	Γυμνό Έδαφος
		
Μεταλλεία/Ορυχεία	Καλλιέργειες Τύπου 3 (ανοιχτό καφέ)	Λίγη ή Καθόλου Βλάστηση
		
Θαμνώδεις Εκτάσεις	Καλλιέργειες Τύπου 5 (σκούρο καφέ)	Δασικές Εκτάσεις
		
Μεικτά Δάση	Καλλιέργειες Τύπου 7 (πράσινο)	Κωνοφόρα Δάση

		
Καλλιέργειες Τύπου 4 (καφέ)	Καλλιέργειες Τύπου 9 (σκούρο μπλε)	Σκληρόφυλλη Βλάστηση
		
Καλλιέργειες Τύπου 6 (λαδί)	Ρηχά Θαλάσσια Ύδατα	Καλλιέργειες Τύπου 2 (γκρι)
		
Καλλιέργειες Τύπου 8 (σκούρο πράσινο)	Σύννεφα	Χερσαία Ύδατα
		
Θαλάσσια Ύδατα	Σκιά Συννέφου	Αστικές Υποδομές

Πίνακας 6.10: Θεματικές κατηγορίες για το μοντέλο εκπαίδευσης

Με το πέρας της εφαρμογής πολλαπλών δοκιμών δημιουργήθηκαν συνολικά 217 πολύγωνα εκπαίδευσης, εκ των οποίων τα 63 ψηφιοποιήθηκαν στην πρώτη υποπεριοχή, τα 38 στη δεύτερη υποπεριοχή, τα 64 στην τρίτη περιοχή και τα υπόλοιπα 52 στην τέταρτη υποπεριοχή. Με βάση τα συγκεκριμένα πολύγωνα δημιουργήθηκε ένα στατιστικό μοντέλο εκπαίδευσης μέσω του αλγορίθμου SVM. Ακολουθεί η διαδικασία για τη δημιουργία του μοντέλου με χρήση των εντολών του orfeotoolbox.

Αρχικά, υπολογίστηκαν τα στατιστικά των τεσσάρων υποπεριοχών με την εξής εντολή:

```
otbcli_ComputeImagesStatistics -il Region_1.tif Region_2.tif Region_3.tif Region_4.tif -out Stats_All.xml
```

Στη συνέχεια, πραγματοποιήθηκε η εκπαίδευση του μοντέλου από τα δεδομένα εκπαίδευσης των τεσσάρων υποπεριοχών χρησιμοποιώντας γραμμικό μοντέλο μηχανών υποστήριξης διανυσμάτων:

```
otbcli_TrainImagesClassifier -io.il Region_1.tif Region_2.tif Region_3.tif Region_4.tif -io.vd Train_1.shp Train_2.shp Train_3.shp Train_4.shp -io.imstat Stats_All.xml --classifier libsvm --classifier.libsvm.k linear -sample.vfn Class -io.out Model_All.svm
```

Το μοντέλο που δημιουργήθηκε χαρακτηριζόταν από δείκτη kappa ίσο με 0.874038 κάτι που απέδειξε πως το στατιστικό μοντέλο εκπαίδευσης που δημιουργήθηκε ήταν σε σημαντικό βαθμό ακριβές για την εφαρμογή του σε διαδικασία ταξινόμησης. Τα αναλυτικά στατιστικά ανά κλάση βρίσκονται στο Παράρτημα Ι με ονομασία «Στατιστικά Μοντέλου Εκπαίδευσης για την Περιοχή της Θεσσαλίας»

6.6 Επιλογή Περιοχής προς Ταξινόμηση

Έχοντας ολοκληρώσει την προεργασία των δεδομένων για όλες τις σκηνές που επιλέχθηκαν καθώς και τη δημιουργία του στατιστικού μοντέλου εκπαίδευσης, επιλέχτηκε μια περιοχή της ευρύτερης περιοχής της Θεσσαλίας με σκοπό την ταξινόμησή της με το μοντέλο που δημιουργήθηκε. Η περιοχή αυτή επιλέχθηκε με βάση τα εξής κριτήρια:

- να ανήκει στο μωσαικό που δημιουργήθηκε από την ημερομηνία αναφοράς (25 Ιουνίου 2013)
- να μην είχε συμπεριληφθεί στη διαδικασία εκπαίδευσης του μοντέλου έτσι ώστε να αξιολογηθεί η ακρίβεια του μοντέλου
- να περιέχει όλες τις κατηγορίες εκπαίδευσης που χρησιμοποιήθηκαν στο μοντέλο εκπαίδευσης
- να καλύπτεται σε κάποιο κομμάτι της από σύννεφα, έτσι ώστε να αξιολογηθεί ο αλγόριθμος ανίχνευσης και αφαίρεσης συννέφων και σκιών που δημιουργήθηκε

Λαμβάνοντας υπόψη τα παραπάνω κριτήρια επιλέχτηκε η ευρύτερη περιοχή της Καρδίτσας. Ακολουθεί η εικόνα της περιοχής αυτής σε έγχρωμο σύνθετο 4-3-2 για την ημερομηνία 25 Ιουνίου 2013.



Εικόνα 6.9: Περιοχή για την εφαρμογή του μοντέλου εκπαίδευσης – Ευρύτερη περιοχή του οικισμού της Καρδίτσας ημερομηνία: 25/6/13, 3359x2606 εικον/χεια , μέγεθος: 1,5 GB

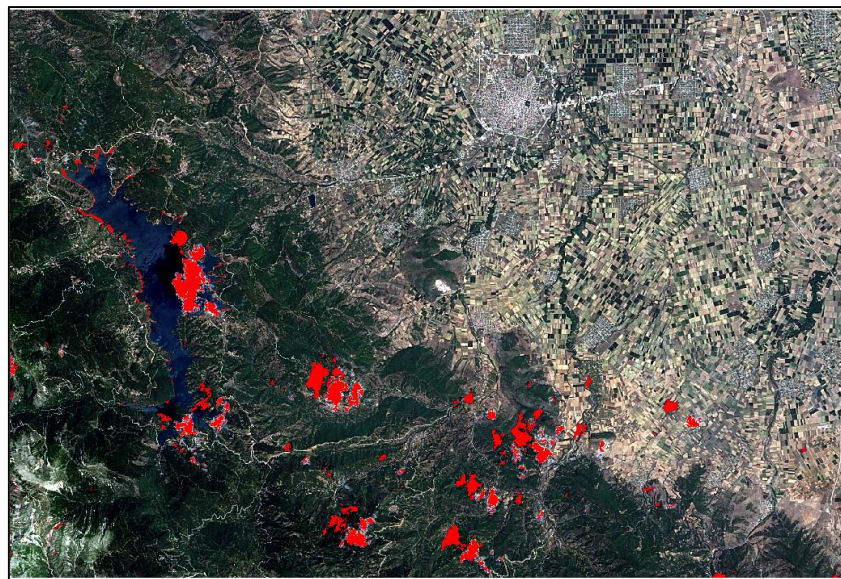
Στη συνέχεια, βρέθηκε η κατάλληλη σκηνή από το σύνολο των δεδομένων που παρουσιάστηκαν στο κεφάλαιο 6.6.1, έτσι ώστε να αντικατασταθούν τα σύννεφα και οι σκιές από τη συγκεκριμένη σκηνή. Για το σκοπό αυτό, επιλέχτηκε η σκηνή 184/33 της ημερομηνίας 10 Ιουλίου 2013, δηλαδή η ακριβώς επόμενη λήψη της συγκεκριμένης περιοχής (+16 μέρες) σε σχέση με την εικόνα αναφοράς. Ακολουθεί η εικόνα της περιοχής σε έγχρωμο σύνθετο 4-3-2 για την ημερομηνία 10 Ιουλίου 2013.



Εικόνα 6.10: Περιοχή για την αντικατάσταση των συννέφων της εικόνας αναφοράς – Ευρύτερη περιοχή του οικισμού της Καρδίτσας 10/7/13

6.7 Εφαρμογή του Αλγορίθμου Ανίχνευσης και Αφαίρεσης Συννέφων και Σκιών

Σαν πρώτο βήμα δημιουργήθηκε μία μάσκα συννέφων και σκιών σύμφωνα με τις συνθήκες που παρουσιάστηκαν στο κεφάλαιο 6.3.2, έτσι ώστε να αξιολογηθεί η εφαρμογή του αλγορίθμου. Ακολουθεί το αποτέλεσμα από την εφαρμογή του αλγορίθμου.



Εικόνα 6.11: Μάσκα συννέφων και σκιών

Ο αλγόριθμος εντόπισε ικανοποιητικά όλα τα σύννεφα και στο μεγαλύτερο βαθμό τους σε επίπεδο εικονοστοιχείου. Ωστόσο, σε κάποια σύννεφα δεν ανιχνεύτηκαν ακριβώς τα όρια τους χωρίς όμως το συγκεκριμένο να αποτελεί πρόβλημα, καθώς κατά τη διαδικασία της αντικατάστασης λήφθηκαν υπόψη και τα τέσσερα περιμετρικά εικονοστοιχεία επιλύοντας, έτσι, το συγκεκριμένο πρόβλημα. Όσον αφορά στις σκιές, εντοπίστηκαν όλες εκτός από τη σκιά που είχε σχηματιστεί μέσα στη λίμνη. Πρόκειται για ένα αναμενόμενο πρόβλημα που προέκυψε από το γεγονός πως οι σκιές στο κανάλι 6 δεν έχουν την ίδια τιμή ανακλαστικότητας στο έδαφος και στο νερό. Παρόλο που η μη ανίχνευση της συγκεκριμένης σκιάς δεν δημιουργούσε σημαντικό πρόβλημα, πραγματοποιήθηκε η εξάλειψή της ψηφιοποιώντας τα όριά της και εντάσσοντάς την στη μάσκα συννέφων και σκιών εκ των υστέρων. Τέλος, εμφανίστηκαν λανθασμένες ενδείξεις (false alarms) σε κάποια σημεία περιμετρικά της λίμνης για τον ίδιο ακριβώς λόγο με παραπάνω. Ομοίως, η αφαίρεσή τους από τη μάσκα πραγματοποιήθηκε με την ψηφιοποίησή τους.

Προκειμένου να αξιολογηθεί η εφαρμογή του αλγορίθμου με ποσοτικά μεγέθη, ψηφιοποιήθηκαν τα σύννεφα και οι σκιές στην εικόνα ανάφοράς και πραγματοποιήθηκε σύγκριση των στατιστικών από τη μάσκα που δημιουργήθηκε από τον αλγόριθμο σε σχέση με τα ψηφιοποιημένα. Ακολουθεί πίνακας με την αξιολόγηση του αλγορίθμου.

	Δείκτες	Αριθμός Εικονοστοιχείων	Δείκτης Αξιολόγησης	Τιμή
Μάσκα από αλγόριθμο ανίχνευσης συννέφων και σκιών	TP	56104	Precision	0.932
	TN	3727250	Recall	0.801
	FP	5203	F-score	0.861
	FN	13909	Accuracy	0.758
Μάσκα από ψηφιοποίηση	TP	74013		

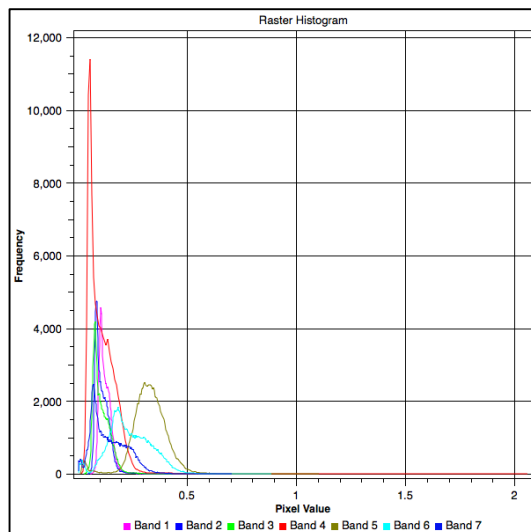
Πίνακας 6.11: Αξιολόγηση αλγορίθμου ανίχνευσης συννέφων και σκιών

Η ακρίβεια του αλγορίθμου ήταν αρκετά ικανοποιητική, λαμβάνοντας υπόψη πως κατά τη διαδικασία της αντικατάστασης εξετάζονται και τα τέσσερα περιμετρικά εικονοστοιχεία των ακραίων εικονοστοιχείων.

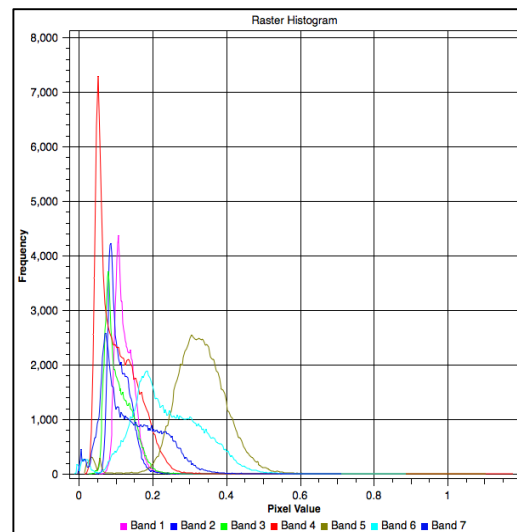
Στο επόμενο βήμα του αλγορίθμου πραγματοποιείται η αντικατάσταση των εικονοστοιχείων που βρίσκονται στα εσωτερικά της μάσκας σύμφωνα με τη μεθοδολογία που αναπτύχθηκε στο κεφάλαιο 6.3.2. Ακολουθεί η παρουσίαση του τελικού αποτελέσματος καθώς και το ιστόγραμμα της εικόνας πριν και μετά την εφαρμογή του αλγορίθμου.



Εικόνα 6.12: Τελική εικόνα μετά την εφαρμογή του αλγορίθμου ανίχνευσης και αφαίρεσης συννέφων και σκιών



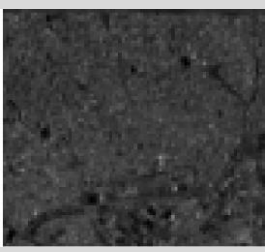
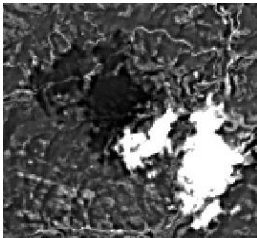
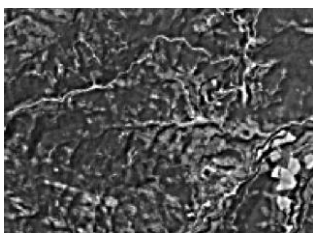
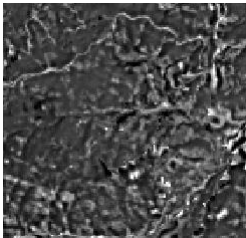
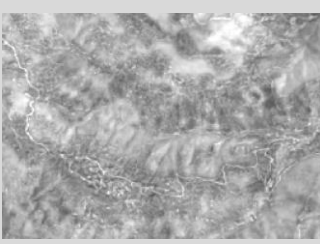
Διάγραμμα 6.4: Αρχικό ιστόγραμμα περιοχής μελέτης



Διάγραμμα 6.5: Ιστόγραμμα μετά την αφαίρεση των συννέφων

Από τη μελέτη των ιστογραμμάτων γίνεται αντιληπτό πως μετά την εφαρμογή του αλγορίθμου αντικατάστασης των συννέφων και των σκιών υπήρξε ένα άπλωμα των τιμών ανακλαστικότητας καθώς και μεταβολή των ελαχίστων και μέγιστων τιμών για κάθε κανάλι. Πρόκειται για ένα αναμενόμενο αποτέλεσμα, καθώς μετά από την αφαίρεση των συννέφων εξαλείφθηκαν τα πολύ φωτεινά εικονοστοιχεία, λόγω συννέφων, καθώς και τα πολύ σκοτεινά εικονοστοιχεία, λόγω σκιών. Επιπλέον, εντάχθηκαν στα στατιστικά της εικόνας νέα εικονοστοιχεία (από την εικόνα που χρησιμοποιήθηκε για την αφαίρεση των συννέφων και των σκιών) οπότε μια μικρή μεταβολή του ιστογράμματος ήταν αναμενόμενη. Ωστόσο, το μεγαλύτερο ποσοστό της εικόνας (περίπου το 98%) παρέμεινε ανεπηρέαστο, καθώς δεν επιλέχτηκαν εικόνες με πολύ μεγάλη συννεφοκάλυψη, έτσι ώστε να διατηρηθούν σε σημαντικό βαθμό τα αρχικά στατιστικά της εικόνας και να μην υπάρξουν μεγάλες μεταβολές που θα αλλοίωναν αισθητά την πραγματικότητα. Τέλος, έγινε εμφανής η ανάγκη για επιλογή δεδομένων που λήφθηκαν σε κοντινό χρονικό διάστημα σε σχέση με την εικόνα αναφοράς, έτσι ώστε να υπάρξουν ικανοποιητικά αποτελέσματα. Έτσι, αξιοποιήθηκε εικόνα με χρονική διαφορά 16 ημερών από την εικόνα αναφοράς, γεγονός που ευθύνεται τόσο για την μεγάλη ακρίβεια του αλγορίθμου όσο και για την μικρή διαφοροποίηση του ιστογράμματος.

Στη συνέχεια, ακολουθεί πίνακας που παρουσιάζει τα αποτελέσματα του αλγορίθμου σε συγκεκριμένες υποπεριοχές της περιοχής μελέτης.

Κανάλι με σύννεφα	Κανάλι χωρίς σύννεφα	Τελικό αποτέλεσμα
		
		
		

Πίνακας 6.12: Παραδείγματα εφαρμογής του αλγορίθμου ανίχνευσης-αφαίρεσης συννέφων/σκιών

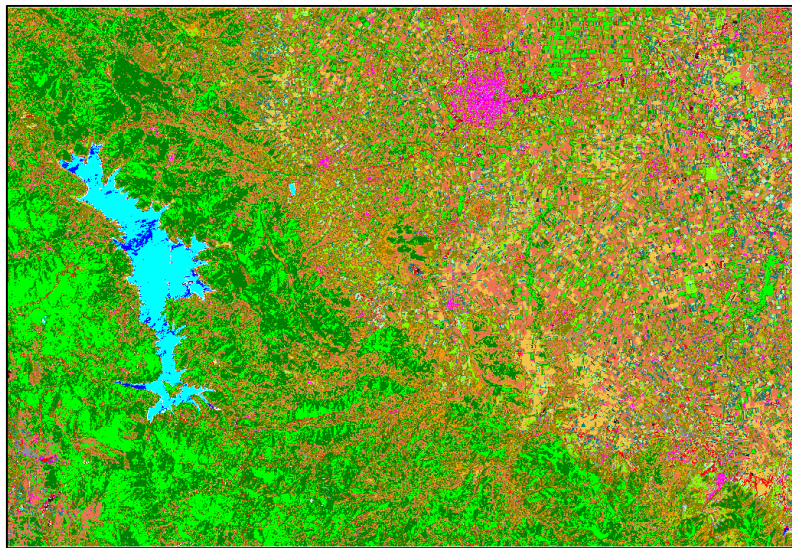
Επεξήγηση πηγαίου κώδικα (Παράρτημα II)

- Στις γραμμές 51-52 πραγματοποιείται η δήλωση των μεταβλητών που χρησιμοποιούνται. Πιο συγκεκριμένα, δηλώνονται οι μεταβλητές τόσο για τα κανάλια που έχουν σύννεφα όσο και για τα αντίστοιχα που χρησιμοποιούνται για την αφαίρεση των συννέφων. Λόγω του γεγονότος πως το πρόγραμμα δέχεται ως δεδομένα εισόδου εικόνες με τιμές ανακλαστικότητας, οι εικόνες δηλώνονται ως μεταβλητές πραγματικών αριθμών. Η διαφοροποίηση στις μεταβλητές των καναλιών με σύννεφα από τα αντίστοιχα κανάλια που χρησιμοποιούνται για την αφαίρεση τους, πραγματοποιείται με την αντίστοιχη ονομασία των μεταβλητών.
- Το πρόγραμμα δέχεται ως δεδομένα εισόδου τα εξής: το κανάλι για την αφαίρεση συννέφων της εικόνας με σύννεφα, το κανάλι 1 της εικόνας με σύννεφα, το κανάλι 3 της εικόνας με σύννεφα, το κανάλι 6 της εικόνας με σύννεφα, το κανάλι που χρησιμοποιείται για την αφαίρεση των συννέφων της εικόνας χωρίς σύννεφα, το κανάλι 1 της εικόνας χωρίς σύννεφα, το κανάλι 3 της εικόνας χωρίς σύννεφα, κανάλι 6 της εικόνας χωρίς σύννεφα καθώς και τις σειρές και τις στηλές των εικόνων. Το διάβασμα των δεδομένων εισόδου πραγματοποιείται στις γραμμές 61-228.
- Στη συνέχεια, δεσμεύεται χώρος μνήμης για το κανάλι που δημιουργείται μέσω της πράξης: $\text{Band 1} + \frac{1}{4} \left(\frac{\text{Band 1} - \text{Band 3}}{\text{Band 1}} \right)$, τόσο για τα δεδομένα με σύννεφα όσο και για τα αντίστοιχα χωρίς σύννεφα. Επιπλέον, δεσμεύεται χώρος μνήμης για την εικόνα εξόδου, όπου το κάθε κανάλι είναι απαλλαγμένο από σύννεφα και σκιές. Τα παραπάνω πραγματοποιούνται στις γραμμές 232-250.
- Στις γραμμές 271-282 δημιουργείται το κανάλι μέσω της πράξης: $\text{Band 1} + \frac{1}{4} \left(\frac{\text{Band 1} - \text{Band 3}}{\text{Band 1}} \right)$ για τις εικόνες με σύννεφα, ενώ αντίστοιχα στις γραμμές 284-296 πραγματοποιείται το βήμα αυτό για τις εικόνες χωρίς σύννεφα.
- Στη συνέχεια του προγράμματος, στις γραμμές 309-323, δημιουργείται μια μάσκα συννέφων-σκιών, με βάση τις συνθήκες που περιγράφηκαν στο κεφάλαιο 6.3.2. Η συγκεκριμένη μάσκα αξιοποιείται για την αφαίρεση των εικονοστοιχείων που έχουν χαρακτηριστεί ως σύννεφα ή σκιές. Έπειτα, ξαναυπολογίζονται τα στατιστικά του καναλιού χωρίς την επίδραση των συννέφων και των σκιών στις γραμμές 325-364. Εν συνεχεία, στις γραμμές 366-395 υπολογίζονται τα στατιστικά για το κανάλι χωρίς σύννεφα και τέλος στις γραμμές 397-404 εφαρμόζεται η διόρθωση των τιμών φωτεινότητας, λόγω διαφορετικής ημέρας λήψης των δεδομένων του καναλιού χωρίς σύννεφα, λαμβάνοντας υπόψη τα νέα στατιστικά που υπολογίστηκαν.

- Στις γραμμές 406-442 αφαιρούνται οι περιοχές με σύννεφα και αντικαθίστανται από τις αντίστοιχες περιοχές της εικόνας χωρίς σύννεφα. Για το σκοπό αυτό, πραγματοποιείται αντικατάσταση κάθε εικονοστοιχείου που έχει χαρακτηριστεί ως σύννεφο ή σκιά, με βάση τη μάσκα που δημιουργήθηκε, και επιπλέον αντικαθίστανται τα τέσσερα περιμετρικά εικονοστοιχεία στις άκρες της μάσκας για να ομαλοποιηθεί το τελικό αποτέλεσμα.
- Τέλος, στη γραμμή 444 πραγματοποιείται το «γράψιμο» της τελικής εικόνας στο αρχείο που έχει οριστεί από το χρήστη και στις γραμμές 447-457 αποδεσμεύεται η μνήμη που έχει χρησιμοποιηθεί για την ανάγνωση των δεδομένων εισόδου.

6.8 Εφαρμογή του Μοντέλου Εκπαίδευσης στην Περιοχή Μελέτης

Με την αφαίρεση των συννέφων και των σκιών ταξινομήθηκε η περιοχή μελέτης χρησιμοποιώντας το στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε από τις τέσσερις υποπεριοχές. Ακολουθεί η εικόνα που ταξινομήθηκε με τη μέθοδο των μηχανών υποστήριξης διανυσμάτων (SVM).



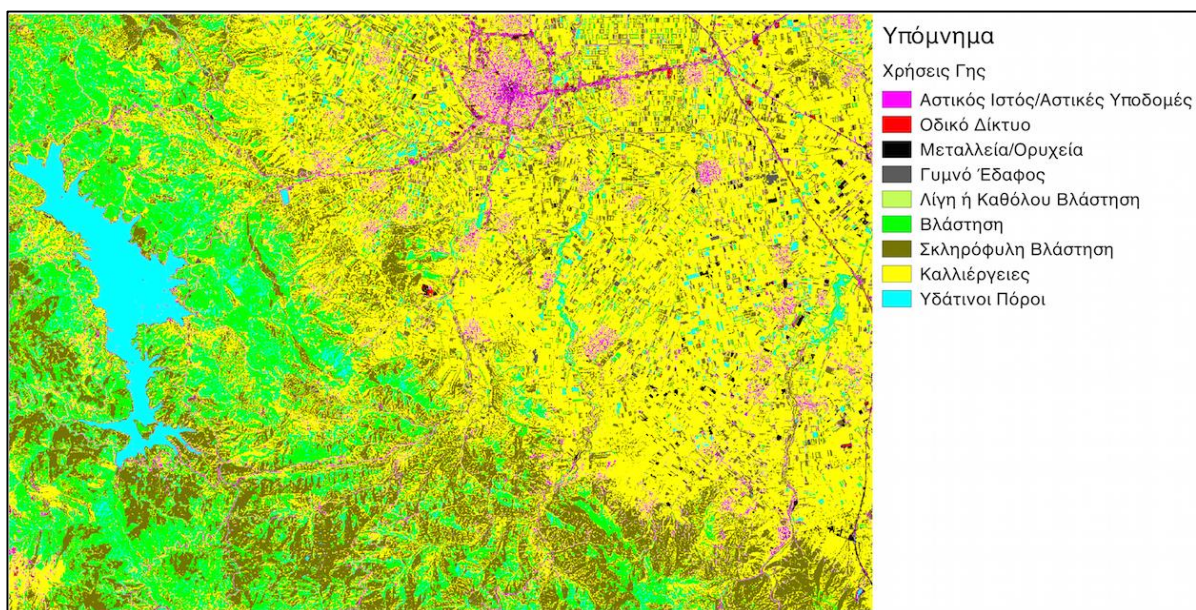
Εικόνα 6.13: Ταξινόμηση περιοχής μελέτης με τον αλγόριθμο SVM

Για την ταξινόμηση της συγκεκριμένης εικόνας χρησιμοποιήθηκαν 25 διαφορετικές κλάσεις από 217 διαφορετικά πολύγωνα εκπαίδευσης, οπότε το τελικό αποτέλεσμα ήταν αρκετά δύσκολο να αξιολογηθεί. Για το σκοπό αυτό συγχωνεύτηκαν οι όμοιες κλάσεις, καθώς το τόσο μεγάλο πλήθος πολυγώνων είχε ως σκοπό αποκλειστικά την ενίσχυση της ακρίβειας του μοντέλου εκπαίδευσης, και όχι την απεικόνιση του τελικού αποτελέσματος. Η συγχώνευση των κατηγοριών οδήγησε στις παρακάτω εννιά θεματικές κατηγορίες:

1. Αστικός ιστός/ Αστικές Υποδομές
2. Οδικό Δίκτυο
3. Μεταλλεία/ Ορυχεία

4. Γυμνό Έδαφος
5. Λίγη ή καθόλου βλάστηση
6. Βλάστηση
7. Σκληρόφυλλη Βλάστηση
8. Καλλιέργειες
9. Υδάτινοι Πόροι

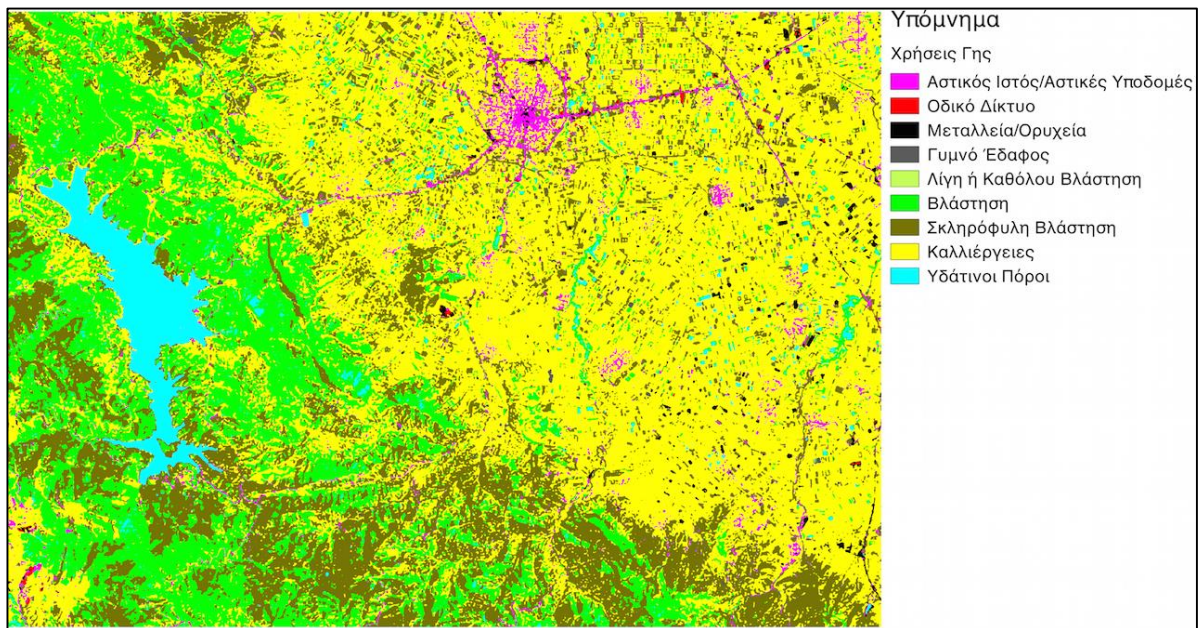
Με τη συγχώνευση των κλάσεων δημιουργήθηκαν πιο γενικές και αντιπροσωπευτικές κατηγορίες κάνοντας πιο εύκολη τη διαδικασία υπολογισμού των εμβαδών καθώς και την αξιολόγηση της ταξινόμησης. Ακολουθεί η ταξινομημένη εικόνα όπως προέκυψε μετά από την συγχώνευση των κλάσεων.



Εικόνα 6.14: Ταξινομημένη εικόνα μετά τη συγχώνευση των όμοιων κλάσεων

Επιπλέον, παρατηρήθηκε πως με την εφαρμογή του αλγορίθμου ταξινόμησης κάποια εικονοστοιχεία ταξινομήθηκαν σε λανθασμένες κατηγορίες, δημιουργώντας θόρυβο στο σύνολο της εικόνας. Για το σκοπό αυτό καθώς και για να δημιουργηθούν πιο ομοιογενείς περιοχές εφαρμόστηκε ένα φίλτρο majority voting με βάση τα γειτονικά εικονοστοιχεία. Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ο αλγόριθμος NeighborhoodMajorityVotingImageFilter του orfeotoolbox που χρησιμοποιεί ένα φίλτρο πλειοψηφίας σε μια κυκλική γειτονιά για κάθε εικονοστοιχείο της ταξινομημένης εικόνας ως προς την πιο αντιπροσωπευτική τιμή. Προκειμένου να μην αλλοιωθούν τα αποτελέσματα και να μην γενικευθούν σε σημαντικό βαθμό οι κατηγορίες επιλέχθηκε η ακτίνα του κύκλου για την εφαρμογή του φίλτρου να είναι ίση με τη μονάδα. Ακολουθεί η ταξινομημένη εικόνα μετά την εφαρμογή του φίλτρου.

```
otbcli_ClassificationMapRegularization -io.in Labeled.tif -io.out Labeled_reg.tif -
ip.radius 1
```



Εικόνα 6.15: Ταξινομημένη εικόνα μετά την εφαρμογή του φίλτρου MajorityVoting

Στη συνέχεια, ακολουθούν τα στατιστικά της ταξινόμησης καθώς και οι εμβαδομετρήσεις των κατηγοριών:

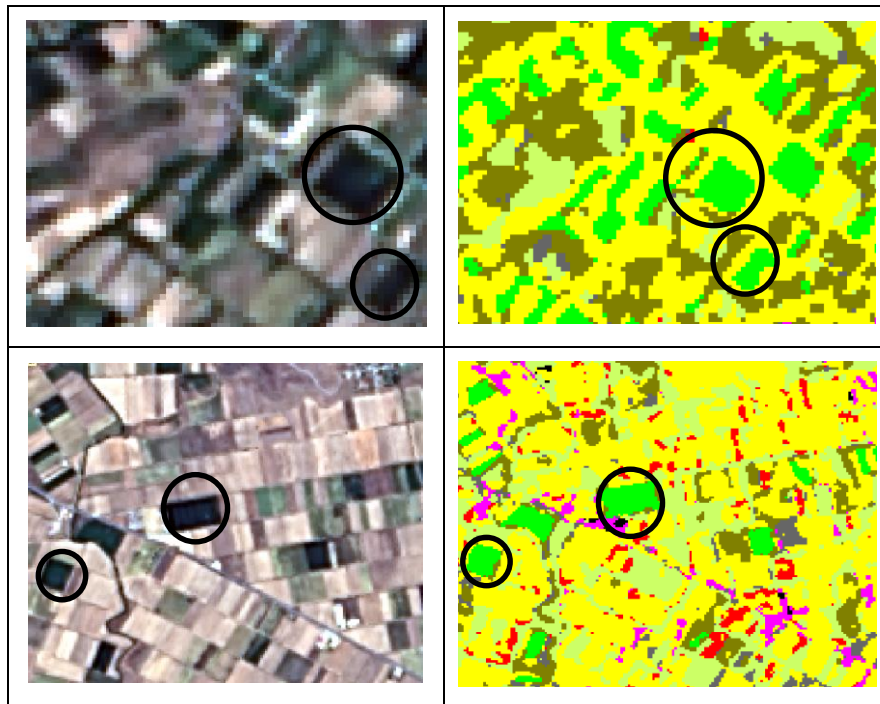
Class	PixelSum	Percentage %	Area [metre ²]
Αστικός Ιστός/ Αστικές Υποδομές	81158	2.142	18260550
Οδικό Δίκτυο	6590	0.174	1482750
Μεταλλεία/Ορυχεία	28115	0.742	6325875
Γυμνό Έδαφος	13319	0.351	2996775
Λίγη ή Καθόλου Βλάστηση	26988	0.172	6072300
Βλάστηση	848105	22.386	190823625
Σκληρόφυλη Βλάστηση	804261	21.229	180958725
Καλλιέργειες	1881813	49.671	423407925
Υδάτινοι Πόροι	98205	2.592	22096125
TOTAL	3788554	100	852424650

Πίνακας 6.13: Στατιστικά και εμβαδομέτρηση από την ταξινόμηση

6.9 Αξιολόγηση Ταξινόμησης

Τα αποτελέσματα της ταξινόμησης αξιολογήθηκαν τόσο ποιοτικά (με διαδικασία φωτοερμηνείας) όσο και ποσοτικά, δημιουργώντας πολύγωνα ελέγχου και υπολογίζοντας τον πίνακα σύγχυσης. Με τη φωτοερμηνεία των αποτελεσμάτων έγινε αντιληπτό πως παρόλο του μεγάλου πλήθους δεδομένων εκπαίδευσης για τους διάφορους τύπους καλλιεργειών, υπήρξαν καλλιέργειες που ταξινομήθηκαν στην κατηγορία της βλάστησης. Πιο συγκεκριμένα, τέτοιου είδους λάθη παρατηρήθηκαν κυρίως στις καλλιέργειες με σκούρο πράσινες αποχρώσεις καθώς

και γενικότερα στις καλλιέργειες που βρίσκονταν σε προχωρημένη ανάπτυξη. Παρακάτω ακολουθούν παραδείγματα από την λανθασμένη ταξινόμηση των συγκεκριμένων περιοχών.



Πίνακας 6.14: Λανθασμένες ενδείξεις στην κατηγορία των καλλιεργειών

Για την ποσοτική αξιολόγηση των αποτελεσμάτων της ταξινόμησης ψηφιοποιήθηκαν πολύγωνα στην περιοχή μελέτης για κάθε κατηγορία που εμφανίστηκε στην ταξινομημένη εικόνα. Πιο συγκεκριμένα, για δεδομένα ελέγχου χρησιμοποιήθηκαν 73 πολύγωνα, δηλαδή το 30% των πολυγώνων εκπαίδευσης που δημιούργησαν το στατιστικό μοντέλο εκπαίδευσης. Ακολουθούν τα στατιστικά της κάθε κλάσης καθώς και ο πίνακας σύγχυσης όπως αυτά προέκυψαν:

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[1]	232	0	6	0	2	0	2	59	0
[2]	11	4	0	1	0	0	0	0	0
[3]	5	0	1	0	5	0	0	0	0
[4]	0	0	0	0	8	0	2	0	0
[5]	4	0	0	0	32	11	6	38	0
[6]	0	0	0	0	0	66	0	0	0
[7]	0	0	0	0	0	33	0	0	0
[8]	0	0	0	0	0	26	0	27	0
[9]	0	0	0	0	0	0	0	0	763

Application.logger (INFO) Precision of class [1] vs all: 0.990991
 Application.logger (INFO) Recall of class [1] vs all: 0.506912
 Application.logger (INFO) F-score of class [1] vs all: 0.670732


```

Application.logger (INFO) Precision of class [2] vs all: 0.333333
Application.logger (INFO) Recall of class [2] vs all: 0.5
Application.logger (INFO) F-score of class [2] vs all: 0.4

Application.logger (INFO) Precision of class [3] vs all: 0.8
Application.logger (INFO) Recall of class [3] vs all: 0.8
Application.logger (INFO) F-score of class [3] vs all: 0.8

Application.logger (INFO) Precision of class [4] vs all: 0.948276
Application.logger (INFO) Recall of class [4] vs all: 0.982143
Application.logger (INFO) F-score of class [4] vs all: 0.964912

Application.logger (INFO) Precision of class [5] vs all: 0.0769231
Application.logger (INFO) Recall of class [5] vs all: 1
Application.logger (INFO) F-score of class [5] vs all: 0.142857

Application.logger (INFO) Precision of class [6] vs all: 0.521739
Application.logger (INFO) Recall of class [6] vs all: 1
Application.logger (INFO) F-score of class [6] vs all: 0.685714

Application.logger (INFO) Precision of class [7] vs all: 0.715328
Application.logger (INFO) Recall of class [7] vs all: 0.659933
Application.logger (INFO) F-score of class [7] vs all: 0.686515

Application.logger (INFO) Precision of class [8] vs all: 0.358621
Application.logger (INFO) Recall of class [8] vs all: 0.604651
Application.logger (INFO) F-score of class [8] vs all: 0.450216

Application.logger (INFO) Precision of class [9] vs all: 1
Application.logger (INFO) Recall of class [9] vs all: 1
Application.logger (INFO) F-score of class [9] vs all: 1

Application.logger (INFO) Precision of the different classes: [0.990991,
0.333333,
0.8, 0.948276, 0.0769231, 0.521739, 0.715328, 0.358621, 1]

Application.logger (INFO) Recall of the different classes: [0.506912,
0.5, 0.8, 0.982143, 1, 1, 0.659933, 0.604651, 1]

Application.logger (INFO) F-score of the different classes: [0.670732,
0.4, 0.8, 0.964912, 0.142857, 0.685714, 0.686515, 0.450216, 1]

Application.logger (INFO) Kappa index: 0.734046

Application.logger (INFO) Overall accuracy index: 0.810775

```

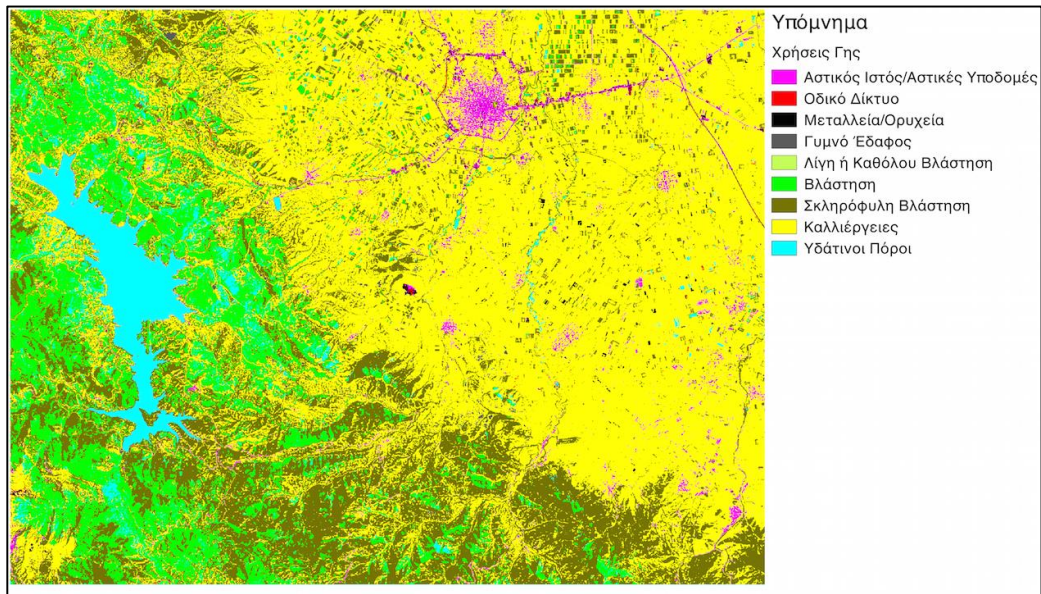
6.10 Εφαρμογή Ανίχνευσης Μεταβολών

Έχοντας ολοκληρώσει την ταξινόμηση της εικόνας αναφοράς, με ημερομηνία 25 Ιουνίου 2013, ανιχνεύθηκαν οι μεταβολές της συγκεκριμένης περιοχής μελέτης μετά από διάστημα ενός έτους. Για το σκοπό αυτό, χρησιμοποιήθηκε το ίδιο στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε για το έτος 2013, έτσι ώστε να ελεγχθεί το κατά πόσο μπορεί να χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων σε διαχρονικά δεδομένα. Ακολουθεί η εικόνα της περιοχής μελέτης για την ημερομηνία 27 Ιουνίου 2014.



Εικόνα 6.16: Περιοχή μελέτης, ημερομηνία: 27/6/14

Για την ανίχνευση των μεταβολών ακολουθήθηκε η τεχνική της post classification analysis, η οποία παρουσιάστηκε στο κεφάλαιο 4.2.3. Πιο συγκεκριμένα, εφόσον πραγματοποιήθηκε η ίδια ακριβώς προεργασία δεδομένων, που έλαβε χώρα και στα δεδομένα του 2013, εφαμόστηκε το στατιστικό μοντέλο εκπαίδευσης που δημιουργήθηκε για να ταξινομηθεί η νέα εικόνα. Ακολουθεί το αποτέλεσμα της ταξινόμησης μετά από τη συγχώνευση των κλάσεων καθώς και την εφαρμογή του φίλτρου Majority Voting για την αντιμετώπιση του θορύβου.



Εικόνα 6.17: Ταξινόμηση περιοχής μελέτης, ημερομηνία: 27/6/14

Στη συνέχεια ακολουθούν τα στατιστικά της ταξινόμησης καθώς και οι εμβαδομετρήσεις των κατηγοριών:

Class	PixelSum	Percentage %	Area [metre ²]
Αστικός Ιστός/ Αστικές Υποδομές	67425	1.779	15170625
Οδικό Δίκτυο	16670	0.44	3750750
Μεταλλεία/Ορυχεία	23674	0.625	5326650
Γυμνό Έδαφος	17134	0.452	3855150
Λίγη ή Καθόλου Βλάστηση	26988	0.172	6072300
Βλάστηση	546196	14.417	122894100
Σκληρόφυλλη Βλάστηση	855187	22.5729	192417075
Καλλιέργειες	2107529	55.0629	474194025
Υδάτινοι Πόροι	100321	2.648	22572225
TOTAL	3788554	100	852424650

Πίνακας 6.15: Στατιστικά και εμβαδομέτρηση από τη νέα ταξινόμηση

Από τα στατιστικά της ταξινόμησης καθώς και από τη φωτοερμηνεία της ταξινομημένης εικόνας έγινε αντιληπτό πως το στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε από τα δεδομένα του 2013 αντιμετώπισε ικανοποιητικά την ταξινόμηση της εικόνας του 2014. Ωστόσο, παρατηρήθηκαν σε κάποιες κατηγορίες αρκετά μεγάλες διαφοροποιήσεις στα εμβαδά και ιδιαίτερα στις κατηγορίες της βλάστησης και των καλλιεργειών. Το συγκεκριμένο οφείλεται αφενός στις πιθανές μεταβολές των χρήσεων γης και αφετέρου στο γεγονός πως χρησιμοποιήθηκε το ίδιο στατιστικό μοντέλο εκμάθησης, χωρίς να δημιουργηθεί νέο για το έτος 2014, για την περαιτέρω αξιολόγησή του. Για την αξιολόγηση της ταξινόμησης δημιουργήθηκαν 73 πολύγωνα στη νέα εικόνα, δηλαδή το 30% των πολυγώνων εκπαίδευσης που δημιούργησαν το στατιστικό μοντέλο εκπαίδευσης.

Ακολουθούν τα στατιστικά της κάθε κλάσης καθώς και ο πίνακας σύγχυσης όπως προέκυψαν:

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[1]	187	4	22	0	0	0	6	80	0
[2]	2	10	0	0	0	0	0	0	0
[3]	3	3	31	0	0	0	0	0	0
[4]	0	0	0	56	0	0	0	0	0
[5]	0	1	0	0	0	0	22	187	0
[6]	0	0	0	0	0	231	0	0	0
[7]	10	0	0	0	0	2	143	60	0
[8]	8	0	0	0	0	0	24	574	0
[9]	0	0	0	0	0	0	0	0	431

Application.logger (INFO) Precision of class [1] vs all: 0.890476
 Application.logger (INFO) Recall of class [1] vs all: 0.625418
 Application.logger (INFO) F-score of class [1] vs all: 0.734774

Application.logger (INFO) Precision of class [2] vs all: 0.555556
 Application.logger (INFO) Recall of class [2] vs all: 0.833333
 Application.logger (INFO) F-score of class [2] vs all: 0.666667

Application.logger (INFO) Precision of class [3] vs all: 0.584906
 Application.logger (INFO) Recall of class [3] vs all: 0.837838
 Application.logger (INFO) F-score of class [3] vs all: 0.688889

Application.logger (INFO) Precision of class [4] vs all: 1
 Application.logger (INFO) Recall of class [4] vs all: 1
 Application.logger (INFO) F-score of class [4] vs all: 1

Application.logger (INFO) Precision of class [5] vs all: 0
 Application.logger (INFO) Recall of class [5] vs all: 0
 Application.logger (INFO) F-score of class [5] vs all: 0

Application.logger (INFO) Precision of class [6] vs all: 0.991416
 Application.logger (INFO) Recall of class [6] vs all: 0.987179
 Application.logger (INFO) F-score of class [6] vs all: 0.989293

Application.logger (INFO) Precision of class [7] vs all: 0.733333
 Application.logger (INFO) Recall of class [7] vs all: 0.481481
 Application.logger (INFO) F-score of class [7] vs all: 0.581301

Application.logger (INFO) Precision of class [8] vs all: 0.588718
 Application.logger (INFO) Recall of class [8] vs all: 0.927302
 Application.logger (INFO) F-score of class [8] vs all: 0.720201

Application.logger (INFO) Precision of class [9] vs all: 1
 Application.logger (INFO) Recall of class [9] vs all: 1
 Application.logger (INFO) F-score of class [9] vs all: 1

Application.logger (INFO) Precision of the different classes: [0.890476, 0.555556, 0.584906, 1, 0, 0.991416, 0.733333, 0.588718, 1]
 Application.logger (INFO) Recall of the different classes: [0.625418,

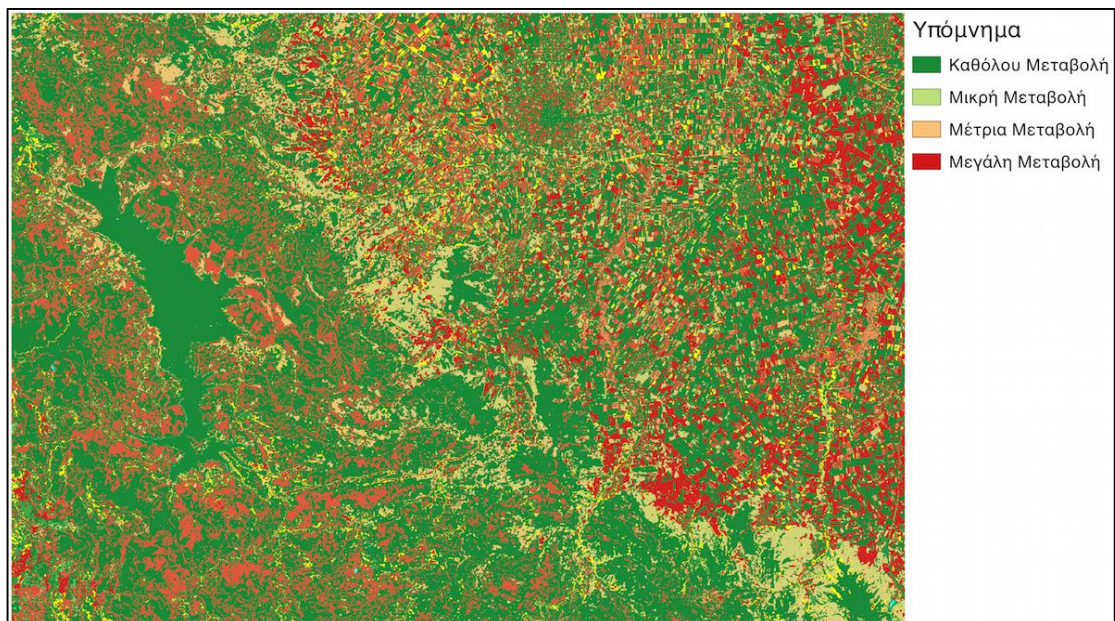
0.833333, 0.837838, 1, 0, 0.987179, 0.481481, 0.927302, 1]

Application.logger (INFO) F-score of the different classes: [0.734774, 0.666667, 0.688889, 1, 0, 0.989293, 0.581301, 0.720201, 1]

Application.logger (INFO) Kappa index: 0.69798

Application.logger (INFO) Overall accuracy index: 0.758401

Τέλος, ακολούθησε η post classification analysis μέσω της εφαρμογής change detection του λογισμικού Saga Gis και υπολογίστηκαν τα ποσοστά των αλλαγών στις χρήσεις γης καθώς και τα είδη των μεταβολών. Ως εικόνα αναφοράς ορίστηκε η εικόνα του 2013 και ως νέα εικόνα η αντίστοιχη του 2014. Ακολουθεί εικόνα που απεικονίζει τη μεταβολή των εικονοστοιχείων από το έτος 2013 στο έτος 2014 καθώς και πίνακας που παρουσιάζει τα ποσοστά των χρήσεων γης που χαρακτηρίστηκαν από μέτρια ή μεγάλη μεταβολή.



Εικόνα 6.18: Ανίχνευση μεταβολών στην περιοχή μελέτης μεταξύ των ετών 2013-2014

Είδος Μεταβολής	Εικόνα 2013	Εικόνα 2014	Αριθμός εικονοστοιχείων που μεταβλήθηκαν
Μέτρια Μεταβολή	Αστικός Ιστός/Υποδομές	Οδικό Δίκτυο	9653
	Αστικός Ιστός/Υποδομές	Μεταλλεία/Ορυχεία	1664
	Οδικό Δίκτυο	Αστικός Ιστός/Υποδομές	2744
	Μεταλλεία/Ορυχεία	Λίγη ή Καθόλου Βλάστηση	5594
	Μεταλλεία/Ορυχεία	Καλλιέργειες	13313
	Λίγη ή Καθόλου Βλάστηση	Γυμνό Έδαφος	8977
	Βλάστηση	Σκληρόφυλη Βλάστηση	29162
Μεγάλη Μεταβολή	Γυμνό Έδαφος	Λίγη ή Καθόλου Βλάστηση	79828
	Γυμνό Έδαφος	Σκληρόφυλη Βλάστηση	88941

	Λίγη ή Καθόλου Βλάστηση	Σκληρόφυλη Βλάστηση	112424
	Βλάστηση	Καλλιέργειες	22811
	Σκληρόφυλη Βλάστηση	Λίγη ή Καθόλου Βλάστηση	45419
	Σκληρόφυλη Βλάστηση	Βλάστηση	297028
	Καλλιέργειες	Λίγη ή Καθόλου Βλάστηση	380847

Πίνακας 6.16: Μέτρια και μεγάλη μεταβολή στις χρήσεις γης για τα έτη 2013-2014

Παρατηρήσεις:

Από τον παραπάνω πίνακα γίνεται αντιληπτό πως οι μεγαλύτερες μεταβολές παρουσιάστηκαν μεταξύ των κατηγοριών γυμνό έδαφος, λίγη ή καθόλου βλάστηση και σκληρόφυλη βλάστηση. Τα παραπάνω αποτελέσματα ανταποκρίνονται σε σημαντικό βαθμό στην πραγματικότητα καθώς από την εποπτεία της περιοχής παρατηρήθηκαν μεγάλες μεταβολές στην κατηγορία του γυμνού εδάφους, στην οποία σε διάφορα σημεία παρατηρήθηκε ύπαρξη αραιούς βλάστησης με την πάροδο του ενός χρόνου. Ωστόσο, ανιχνεύθηκαν κάποιες λανθασμένες ενδείξεις μεταβολών στην κατηγορία των καλλιεργειών, όπου η εκτέλεση του αλγορίθμου επέδειξε την αλλαγή της χρήσεως γης σε λίγη ή καθόλου βλάστηση. Το παραπάνω αποτέλεσμα δεν ανταποκρίνεται στην πραγματικότητα και οφείλεται στην αδυναμία του στατιστικού μοντέλου να ταξινομήσει με μεγάλη ακρίβεια την εικόνα του 2014.

Συνοψίζοντας, η ακρίβεια της διαδικασίας εξαρτάται άμεσα από την ακρίβεια των επιμέρους ταξινομήσεων που πραγματοποιήθηκαν στις δύο εικόνες. Για το λόγο αυτό ανιχνεύθηκαν λανθασμένες ενδείξεις μεταβολών σε κάποιες υποπεριοχές. Για τη βελτίωση των αποτελεσμάτων θα μπορούσε να πραγματοποιηθεί παράλληλη εκπαίδευση του μοντέλου εκπαίδευσης από τα δεδομένα των εικόνων του 2014, το οποίο όμως δεν αποτελούσε αντικείμενο της συγκεκριμένης διπλωματικής εργασίας, η οποία έχει ως στόχο την αξιολόγηση ενός στατιστικού μοντέλου εκμάθησης για μια χρονική περίοδο όπου θεωρείται πως δεν υπάρχει γνώση εδάφους.

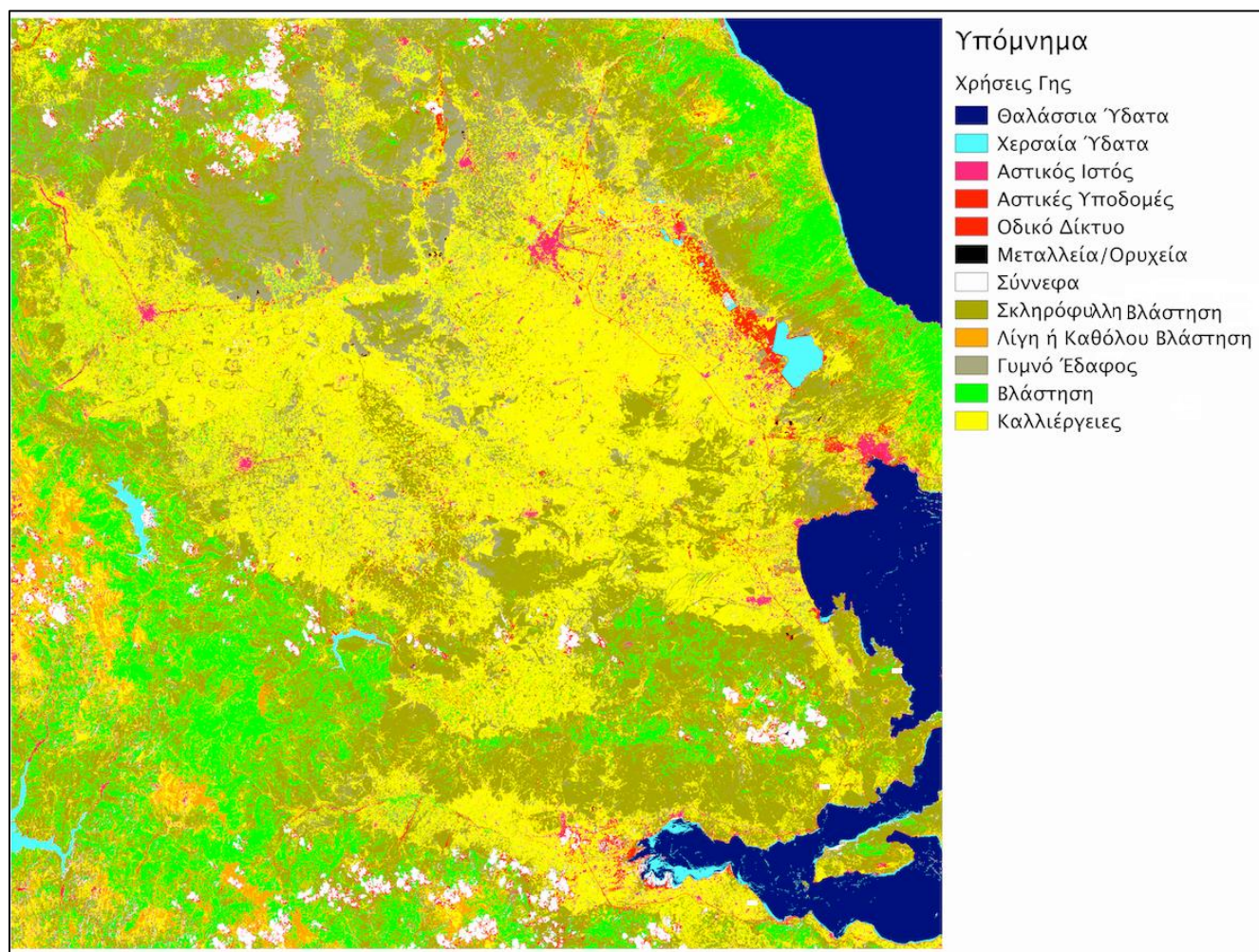
6.11 Πειραματικά Αποτελέσματα στην Ευρύτερη Περιοχή της Θεσσαλίας

Στο συγκεκριμένο κεφάλαιο πραγματοποιείται η ταξινόμηση του μωσαικού της ευρύτερης περιοχής της Θεσσαλίας για την ημερομηνία 25/6/13, το οποίο δημιουργήθηκε με τη διαδικασία που περιγράφηκε στο υποκεφάλαιο 6.4. Για το σκοπό αυτό, αξιοποιήθηκε το ίδιο στατιστικό μοντέλο εκπαίδευσης που δημιουργήθηκε νωρίτερα, έτσι ώστε να αξιολογηθεί η αποτελεσματικότητά του και σε μεγαλύτερες περιοχές. Παράλληλα, στη συγκεκριμένη εφαρμογή δεν χρησιμοποιήθηκε ο αλγόριθμος ανίχνευσης και αφαίρεσης συννέφων και σκιών

που δημιουργήθηκε, έτσι ώστε να αξιολογηθεί κατά πόσο το στατιστικό μοντέλο ήταν σε θέση να ανιχνεύσει και να κατηγοριοποιήσει τις κατηγορίες αυτές. Για την ταξινόμηση του μωσαικού επιλέχθηκε, όπως και πριν, η συγχώνευση των όμοιων τάξεων έτσι ώστε να είναι πιο εύκολη η αξιολόγηση της ταξινόμησης. Πιο συγκεκριμένα, δημιουργήθηκαν οι εξής κατηγορίες:

1. Θαλάσσια Ύδατα
2. Χερσαία Ύδατα
3. Αστικός Ιστός
4. Αστικές Υποδομές
5. Οδικό Δίκτυο
6. Μεταλλεία/Ορυχεία
7. Γυμνό Έδαφος
8. Σύννεφα
9. Σκιές Συννέφων
10. Σκληρόφυλλη Βλάστηση
11. Λίγη ή Καθόλου Βλάστηση
12. Βλάστηση
13. Καλλιέργειες

Ακολουθεί η ταξινομημένη εικόνα της ευρύτερης περιοχής της Θεσσαλίας, όπως αυτή προέκυψε.



Εικόνα 6.19: Ταξινόμηση της ευρύτερης περιοχής της Θεσσαλίας, 06/2013

Στη συνέχεια, πραγματοποιήθηκε η αξιολόγηση της ταξινόμησης με την ψηφιοποίηση πολυγώνων ελέγχου. Λόγω του γεγονότος πως η περιοχή μελέτης ήταν μεγάλη σε έκταση δεν αρκούσαν μόνο 70 πολύγωνα ελέγχου, όπως έγινε στην προηγούμενη εφαρμογή και έτσι χρησιμοποιήθηκαν 135 πολύγωνα ελέγχου. Με την αξιολόγηση της ταξινόμησης προέκυψαν τα εξής στατιστικά για κάθε κλάση:

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]
[1]	13274	330	0	0	0	0	0	0	0	0	0	0	0
[2]	0	1011	0	0	0	0	0	0	0	0	0	0	0
[3]	0	0	617	23	64	0	0	0	2	0	0	0	46
[4]	0	0	0	73	2	4	0	0	0	0	0	0	0
[5]	0	0	1	0	164	0	0	0	0	0	0	0	16
[6]	0	0	0	4	0	74	0	0	0	0	0	0	0
[7]	0	0	0	0	0	0	294	0	0	0	0	0	0
[8]	0	0	0	0	0	0	0	223	0	0	0	0	0
[9]	0	0	0	0	0	0	0	0	525	0	0	10	0
[10]	0	0	0	0	0	0	0	0	0	36	0	0	2

[11]	0	0	0	0	0	0	0	0	3	0	257	0	0
[12]	0	0	0	0	0	0	0	0	4	16	0	548	0
[13]	0	122	795	6	491	0	0	5	1142	44	241	216	5216

Application.logger (INFO) Precision of class [1] vs all: 1
Application.logger (INFO) Recall of class [1] vs all: 0.975742
Application.logger (INFO) F-score of class [1] vs all: 0.987722

Application.logger (INFO) Precision of class [2] vs all: 0.691046
Application.logger (INFO) Recall of class [2] vs all: 1
Application.logger (INFO) F-score of class [2] vs all: 0.8173

Application.logger (INFO) Precision of class [3] vs all: 0.43666
Application.logger (INFO) Recall of class [3] vs all: 0.820479
Application.logger (INFO) F-score of class [3] vs all: 0.569977

Application.logger (INFO) Precision of class [4] vs all: 0.688679
Application.logger (INFO) Recall of class [4] vs all: 0.924051
Application.logger (INFO) F-score of class [4] vs all: 0.789189

Application.logger (INFO) Precision of class [5] vs all: 0.227462
Application.logger (INFO) Recall of class [5] vs all: 0.906077
Application.logger (INFO) F-score of class [5] vs all: 0.363636

Application.logger (INFO) Precision of class [6] vs all: 0.948718
Application.logger (INFO) Recall of class [6] vs all: 0.948718
Application.logger (INFO) F-score of class [6] vs all: 0.948718

Application.logger (INFO) Precision of class [7] vs all: 1
Application.logger (INFO) Recall of class [7] vs all: 1
Application.logger (INFO) F-score of class [7] vs all: 1

Application.logger (INFO) Precision of class [8] vs all: 0.97807
Application.logger (INFO) Recall of class [8] vs all: 1
Application.logger (INFO) F-score of class [8] vs all: 0.988914

Application.logger (INFO) Precision of class [9] vs all: 0.313246
Application.logger (INFO) Recall of class [9] vs all: 0.981308
Application.logger (INFO) F-score of class [9] vs all: 0.474898

Application.logger (INFO) Precision of class [10] vs all: 0.375
Application.logger (INFO) Recall of class [10] vs all: 0.947368
Application.logger (INFO) F-score of class [10] vs all: 0.537313

Application.logger (INFO) Precision of class [11] vs all: 0.516064
Application.logger (INFO) Recall of class [11] vs all: 0.988462
Application.logger (INFO) F-score of class [11] vs all: 0.6781

Application.logger (INFO) Precision of class [12] vs all: 0.70801
Application.logger (INFO) Recall of class [12] vs all: 0.964789
Application.logger (INFO) F-score of class [12] vs all: 0.816692

Application.logger (INFO) Precision of class [13] vs all: 0.987879
Application.logger (INFO) Recall of class [13] vs all: 0.630104
Application.logger (INFO) F-score of class [13] vs all: 0.769435

Application.logger (INFO) Precision of the different classes: [1, 0.691046, 0.43666, 0.688679, 0.227462, 0.948718, 1, 0.97807, 0.313246, 0.375, 0.516064, 0.70801, 0.987879]

Application.logger (INFO) Recall of the different classes: [0.975742, 1, 0.820479, 0.924051, 0.906077, 0.948718, 1, 1, 0.981308, 0.947368, 0.988462, 0.964789, 0.630104]

Application.logger (INFO) F-score of the different classes: [0.987722, 0.8173, 0.569977, 0.789189, 0.363636, 0.948718, 1, 0.988914, 0.474898, 0.537313, 0.6781, 0.816692, 0.769435]

Application.logger (INFO) Kappa index: 0.78982

Application.logger (INFO) Overall accuracy index: 0.861434

Παρατηρήσεις

Από την αξιολόγηση της ταξινόμησης μέσω των στατιστικών για κάθε κλάση, του πίνακα σύγχυσης καθώς και της φωτοερμηνείας της περιοχής μελέτης έγινε αντιληπτό πως το στατιστικό μοντέλο εκπαίδευσης αντιμετώπισε ικανοποιητικά την ταξινόμηση της περιοχής. Πιο συγκεκριμένα, οι κατηγορίες των συννέφων ανιχνεύθηκαν και ταξινομήθηκαν ικανοποιητικά, ενώ αντίθετα οι σκιές τους δεν χαρακτηρίζονται από υψηλή ακρίβεια. Σε σύγκριση με τον αλγόριθμο που αναπτύχθηκε στο κεφάλαιο 6.3, η ταξινόμηση δεν αντιμετώπισε το ίδιο αποτελεσματικά την ανίχνευση των σκιών και μεγάλο μέρος τους ταξινομήθηκε στην κατηγορία των αστικών υποδομών. Επιπλέον, προβλήματα παρουσιάστηκαν και στις καλλιέργειες που είχαν αποχρώσεις του σκούρου γκρι, όπου και σε αυτήν την περίπτωση μεγάλο πλήθος εικονοστοιχείων ταξινομήθηκε στην κατηγορία των αστικών υποδομών. Ωστόσο, το στατιστικό μοντέλο εκπαίδευσης ταξινόμησε με μεγάλη ακρίβεια τις κατηγορίες της βλάστησης, της σκληρόφυλλης βλάστησης καθώς και των υδατικών πόρων. Επιπλέον, ο διαχωρισμός του οδικού δικτύου ήταν αρκετά αποτελεσματικός αν ληφθεί υπόψη πως τα δεδομένα των δορυφόρων Landsat, λόγω της ανάλυσής τους, δεν ενδείκνυνται για την αντιμετώπιση προβλημάτων τέτοιας κλίμακας. Τέλος, τα αποτελέσματα από την εφαρμογή του αλγορίθμου ταξινόμησης SVM με την αξιοποίηση διαδικασιών μηχανικής μάθησης έδωσαν ικανοποιητικά αποτελέσματα στην ταξινόμηση μεγάλων περιοχών αξιοποιώντας πληροφορία και γνώση από υποπεριοχές. Παρόλα αυτά, η περαιτέρω ενίσχυση του στατιστικού μοντέλου με αντιπροσωπευτικά πολύγωνα από όλες τις χρονικές περιόδους που επιλέχθηκαν να χρησιμοποιηθούν, θα ενίσχυε πιθανώς τόσο την ακρίβεια του μοντέλου εκπαίδευσης όσο και τα τελικά αποτελέσματα της ταξινομημένης εικόνας.

6.12 Ενσωμάτωση Αποτελεσμάτων στη Γεωχωρική Βάση και σε Web Gis

Σαν τελικό στάδιο δημιουργήθηκε μία δοκιμαστική έκδοση (demo) Web Gis για να παρουσιαστούν τα αποτελέσματα της μεθοδολογίας. Για το σκοπό αυτό, χρησιμοποιήθηκαν οι διαδικτυακές γλώσσες html και javascript καθώς και η

διεπαφή openlayers για τη δημιουργία των κατάλληλων επιπέδων. Η κατασκευή του Web Gis πραγματοποιήθηκε σε τοπικό επίπεδο (localhost) και για την απεικόνιση των επιπέδων επιλέχθηκε το φορμάτ TMS (Tile Map Service). Επιπλέον, ως υπόβαθρο χρησιμοποιήθηκε το αντίστοιχο του OpenStreetMap. Ο πηγαίος κώδικας που αναπτύχθηκε βρίσκεται στο Παράρτημα II με ονομασία «Εξαγωγή Αποτελεσμάτων σε Web Gis με Χρήση Ανοιχτών Προτύπων»

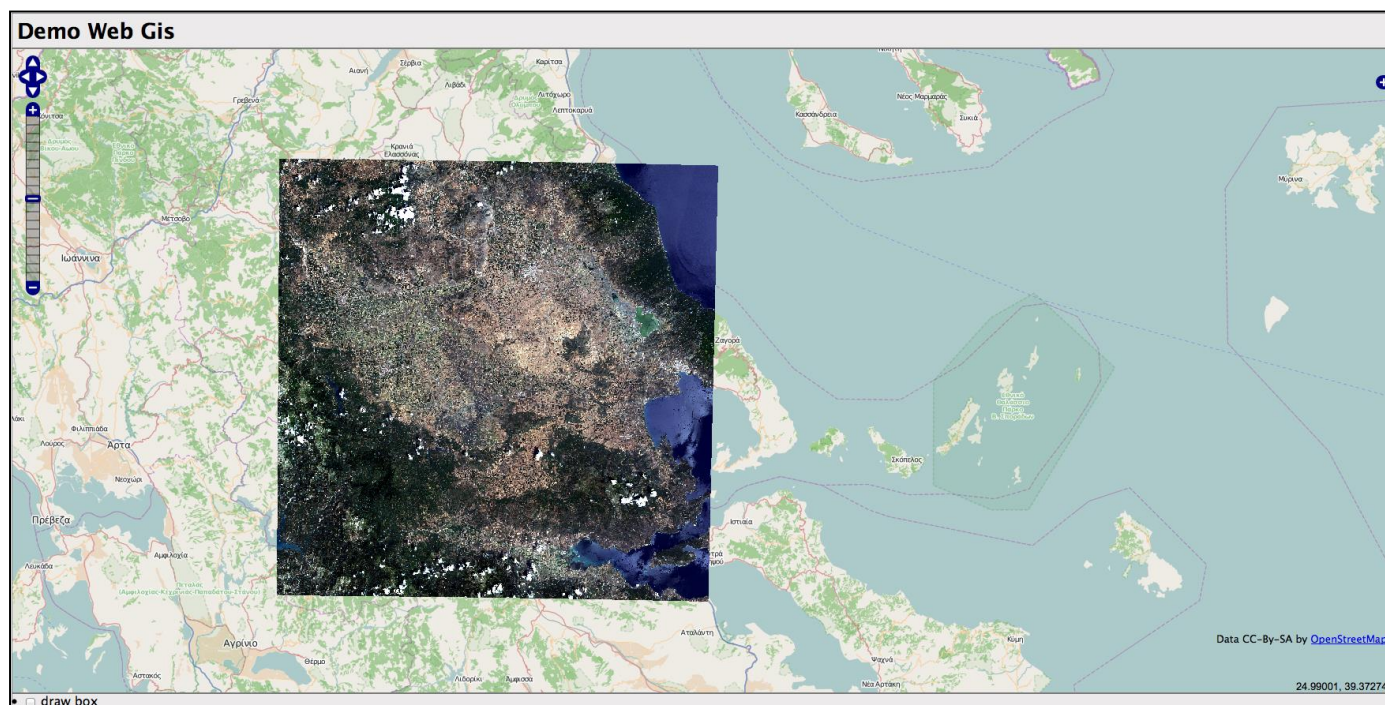
Το φορμάτ TMS είναι μία προδιαγραφή για την αποθήκευση και ανάκτηση χαρτογραφικών δεδομένων, που αναπτύχθηκε από την Geospatial Foundation Open Source. Μέσω του συγκεκριμένου φορμάτ «γεφυρώνεται» το κενό ανάμεσα στο πρότυπο που χρησιμοποιείται από το OpenStreetMap (OSM) και το αντίστοιχο πρότυπο WMS, παρέχοντας τη δημιουργία «πλακιδίων» (tiles) σε εναλλακτικό σύστημα χωρικής συσχέτισης. Το συγκεκριμένο φορμάτ χρησιμοποιείται ευρέως σε διαδικτυακές εφαρμογές, λόγω του γεγονότος πως η ανάκτησή του είναι άμεση και δεν δημιουργούνται προβλήματα επικοινωνίας μεταξύ διακομιστή και χρήστη.

Η κατασκευή του Web Gis περιέχει τα εξής χαρακτηριστικά:

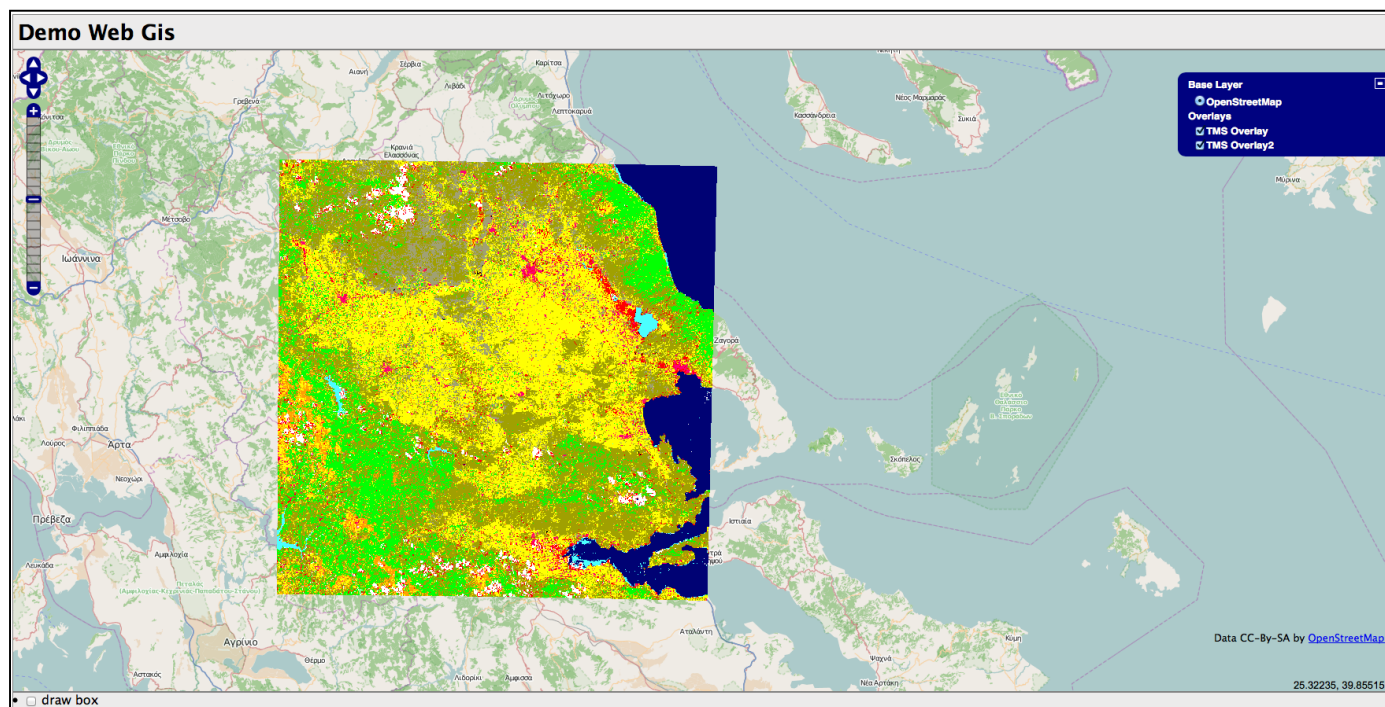
- Υπόβαθρο Open Street Map (OSM)
- Επίπεδο για το μωσαϊκό της ευρύτερης περιοχής της Θεσσαλίας σε φορμάτ TMS
- Επίπεδο για την ταξινομημένη εικόνα που προέκυψε με το πέρας της μεθοδολογίας, σε φορμάτ TMS
- Δυνατότητα για οριοθέτηση περιοχής μελέτης, μέσω ψηφιοποίησης πολυγώνου, και μετέπειτα παρουσίαση της συγκεκριμένης περιοχής απαλλαγμένη από σύννεφα καθώς και ταξινομημένη

Στο Web Gis που αναπτύχθηκε δεν πραγματοποιούνται σε πραγματικό χρόνο (on the fly) οι αλγόριθμοι, αλλά σχεδιάστηκε με τη λογική πως ο χρήστης θα οριοθετεί ή θα επιλέγει την περιοχή μελέτης του, το αίτημά του θα στέλνεται στο διακομιστή και στο τέλος ο χρήστης θα ενημερώνεται για τα αποτελέσματα με κατάλληλο μήνυμα (π.χ. email) για την ολοκλήρωση των διαδικασιών (on demand). Ο λόγος που δεν σχεδιάστηκε για να παρέχει προϊόντα σε πραγματικό χρόνο είναι πως οι αλγόριθμοι που πραγματοποιούνται δεν είναι πλήρως αυτοματοποιημένοι και απαιτείται η παρέμβαση του μελετητή για να βελτιστοποιούνται τα τελικά αποτελέσματα.

Ακολουθούν αποσπάσματα οθόνης από τη δοκιμαστική έκδοση του Web Gis.

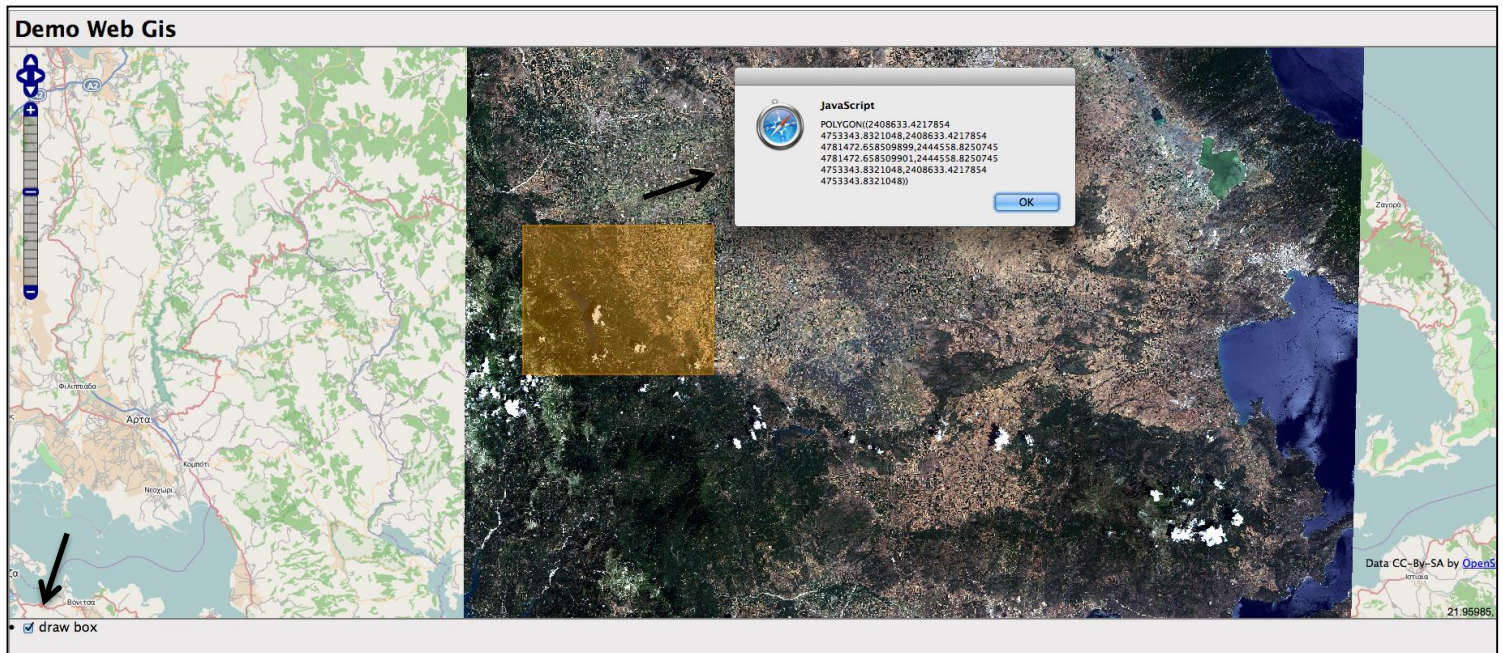


Εικόνα 6.20: TMS μωσαϊκού Θεσσαλίας



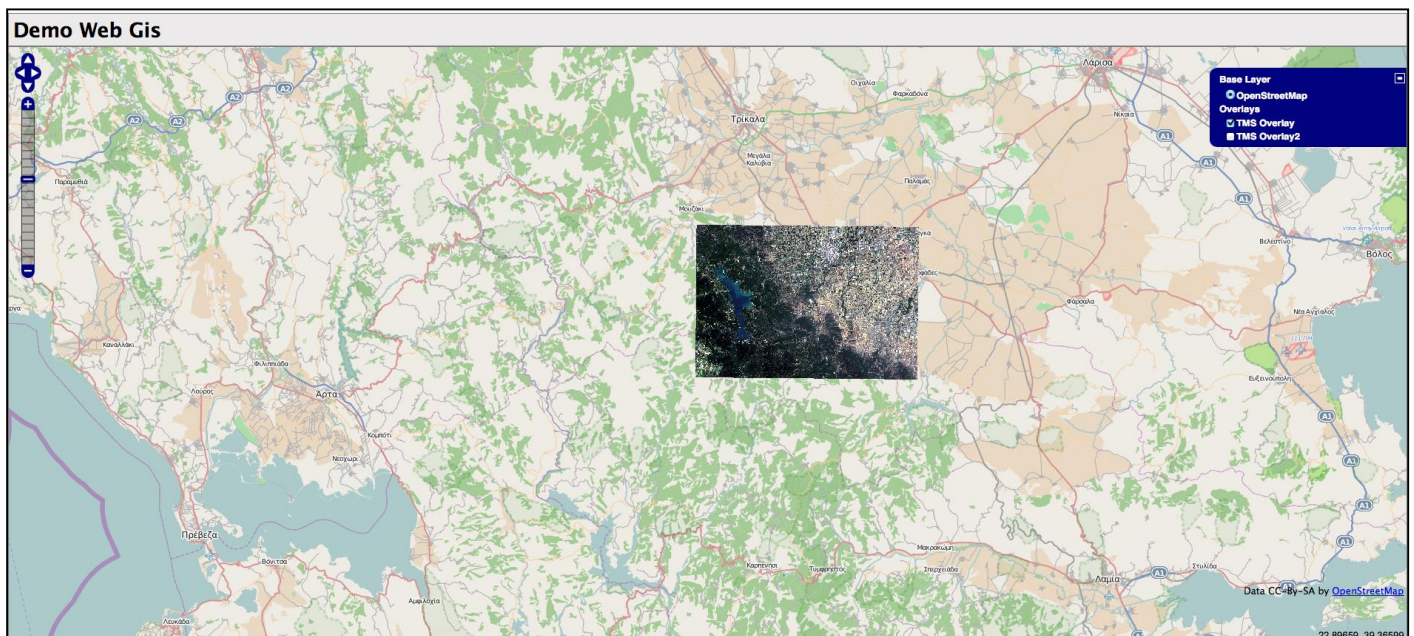
Εικόνα 6.21: TMS ταξινόμηση μωσαϊκού Θεσσαλίας

Στην εικόνα 6.20 φαίνεται το μωσαϊκό της ευρύτερης περιοχής της Θεσσαλίας, ενώ στην εικόνα 6.21 φαίνεται η πρόσθεση της ταξινομημένης εικόνας στον κατάλογο επιπέδων.

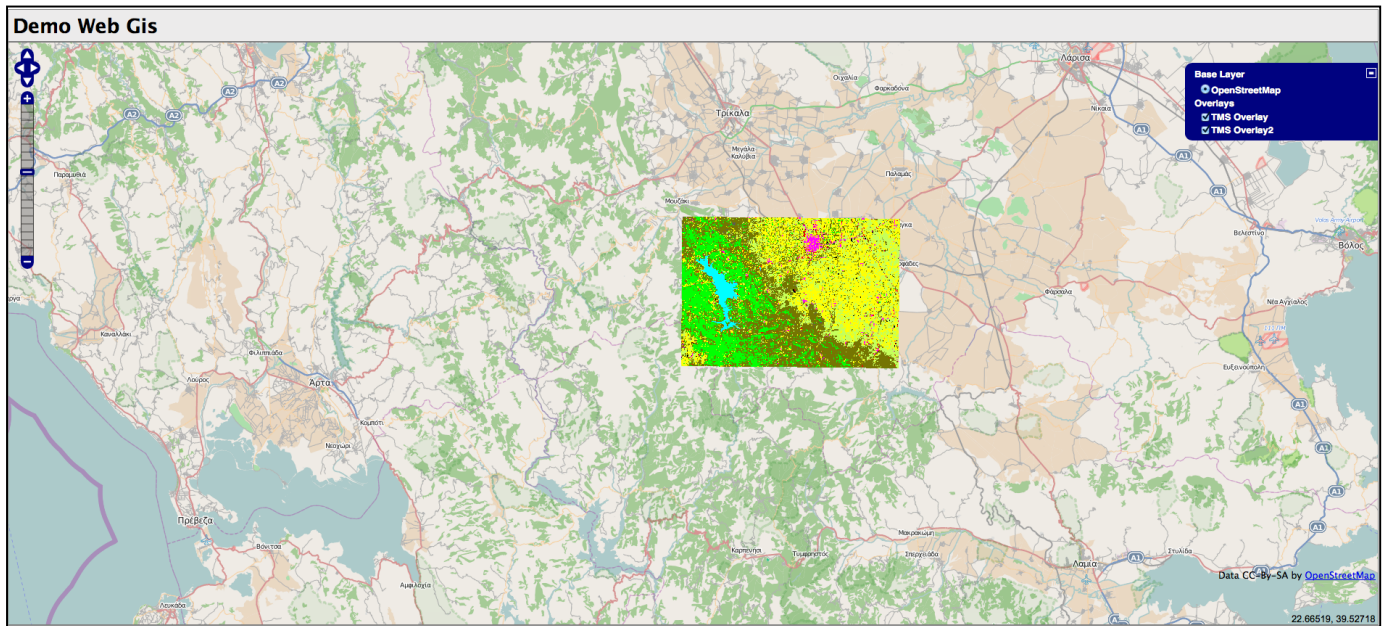


Εικόνα 6.22: Οριοθέτηση περιοχής μελέτης

Στην παραπάνω εικόνα φαίνεται το «εργαλείο» που χρησιμοποιεί ο χρήστης για την οριοθέτηση της περιοχής μελέτης, έτσι ώστε να εφαρμοστεί η μεθοδολογία που παρουσιάστηκε. Επιπλέον, στην οθόνη του χρήστη εμφανίζεται παράθυρο που παρουσιάζει τις συντεταγμένες του πολυγώνου που αυτός όρισε.



Εικόνα 6.23: Περιοχή μελέτης απαλλαγμένη από σύννεφα



Εικόνα 6.24: Ταξινόμηση περιοχής μελέτης

Στην εικόνα 6.23 παρουσιάζεται η περιοχή μελέτης που επιλέχθηκε από το χρήστη, ύστερα από την εφαρμογή των απαραίτητων διαδικασιών προεργασίας καθώς και του αλγορίθμου ανίχνευσης και αφαίρεσης συννέφων και σκιών. Τέλος, στην εικόνα 6.24 παρουσιάζεται η ταξινομημένη εικόνα της συγκεκριμένης περιοχής, η οποία δημιουργήθηκε βάσει της μεθοδολογίας που παρουσιάστηκε.

Επεξήγηση πηγαίου κώδικα

- Στις γραμμές 1-16 ορίζονται τα πλαίσια και οι επικεφαλίδες για τη δημιουργία του διαδικτυακού χώρου. Επίσης, στη γραμμή 17 «φορτώνεται» η διεπαφή προγράμματος εφαρμογής (API) του openlayers από τον αντίστοιχο ιστοχώρο.
- Στις γραμμές 19-35 ορίζονται τα αρχικά όρια απεικόνισης του χάρτη καθώς και τα ελάχιστα και μέγιστα όρια σμίκρυνσης και μεγέθυνσης που μπορεί να πραγματοποιήσει ο χρήστης. Επιπλέον, ορίζεται το σύστημα αναφοράς που χρησιμοποιείται για το φορμάτ TMS και πραγματοποιείται η πρώτη συνάρτηση αρχικοποίησης του διαδικτυακού χώρου.
- Στις γραμμές 36-80 δημιουργείται το επίπεδο για το υπόβαθρο του Open Street Map και αρχικοποιούνται οι παράμετροι για τη δημιουργία των επιπέδων TMS. Πιο συγκεκριμένα, καθορίζεται το φορμάτ των «πλακιδίων», το όνομά τους καθώς και η διεύθυνση για την ανάκτησή τους. Επιπλέον, δημιουργούνται και στοιχίζονται τα απαραίτητα εργαλεία που θα εμφανίζονται στο διαδικτυακό χώρο και πραγματοποιούνται οι απαραίτητοι περιορισμοί.

- Στις γραμμές 81-122 πραγματοποιείται η ανάγνωση και ανάκτηση των πλακιδίων σε μορφή TMS, ανάλογα με το επίπεδο μεγέθυνσης που επιλέγει ο χρήστης. Στη συγκεκριμένη εφαρμογή, δημιουργήθηκαν «πλακίδια» για επίπεδα μεγέθυνσης 1-15, χρησιμοποιώντας το πρόσθετο Qtiles του Qgis.
- Στις γραμμές 123-149 πραγματοποιείται η κατάλληλη οριοθέτηση των πλακιδίων που δημιουργήθηκαν και στοιχίζονται ανάλογα με το περίγραμμα του υποβάθρου και την αντίστοιχη γεωαναφορά που περιέχουν. Επιπλέον, ελέγχονται και περιορίζονται τα όρια απεικόνισης έτσι ώστε να μην δημιουργούνται προβλήματα μεταξύ αυτών και των επιπέδων των εργαλείων.
- Στις γραμμές 150-175 δημιουργείται το εργαλείο για τη ψηφιοποίηση πολυγώνων και καθορίζεται το μήνυμα που εμφανίζεται στο χρήστη όταν ολοκληρώσει τον καθορισμό του πολυγώνου. Τέλος, στις γραμμές 176-191 πραγματοποιείται η τελική οριοθέτηση των επικεφαλίδων (headers) και του κύριου μέρους του πηγαίου κώδικα (body).

Κεφάλαιο 7: Συμπεράσματα και Προοπτικές

7.1 Συμπεράσματα

Με την εφαρμογή και την αξιολόγηση της μεθοδολογικής προσέγγισης που ακουούθηκε, στα διάφορα στάδια της διαδικασίας ανάλυσης, εξάχθηκαν τα παρακάτω συμπεράσματα:

- Έγινε αντιληπτή η αναγκαιότητα για την εφαρμογή ραδιομετρικών διορθώσεων στις περιπτώσεις που απαιτούνταν παράλληλη χρήση διαχρονικών δεδομένων. Παρόλο που χρησιμοποιήθηκε ο ίδιος δορυφορικός δέκτης και αξιοποιήθηκαν δεδομένα από την ίδια περίοδο (καλοκαίρι), η εφαρμογή των ραδιομετρικών διορθώσεων βελτίωσε αισθητά τα αποτελέσματα των ταξινομήσεων καθώς περιορίστηκαν τα σφάλματα που οφείλονταν στη γωνία του ήλιου και γενικότερα στις διαφορετικές συνθήκες λήψης. Επιπλέον, ο μετασχηματισμός των δεδομένων μέσω αλγορίθμων pansharpening βελτίωσε ραδιομετρικά τις εικόνες προσφέροντας υψηλότερα ποσοστά ακρίβειας κατά τη διαδικασία της ταξινόμησης.
- Το στατιστικό μοντέλο εκμάθησης που δημιουργήθηκε από τις τέσσερις υποπεριοχές έδωσε ικανοποιητικά αποτελέσματα στην ταξινόμηση της τυχαίας περιοχής για το έτος 2013. Ωστόσο, παρατηρήθηκαν αδυναμίες στην ταξινόμηση της ίδιας περιοχής μετά από ένα έτος και ιδιαίτερα στις κατηγορίες των καλλιεργειών και του γυμνού εδάφους. Παρόλα αυτά, η ακρίβεια της ταξινόμησης για το έτος 2014 ήταν αρκετά καλή, αν ληφθεί υπόψη πως το μοντέλο εκπαίδευσης δημιουργήθηκε από δεδομένα αποκλειστικά του 2013 και εφαρμόστηκε σε εικόνα του 2014 χωρίς να υπάρξει περαιτέρω εκπαίδευση του στατιστικού μοντέλου.
- Όσον αφορά την εφαρμογή του αλγορίθμου ανίχνευσης και αφαίρεσης συννέφων και σκιών, παρατηρήθηκε πως αντιμετώπισε αποτελεσματικά το πρόβλημα της ύπαρξης συννέφων. Ωστόσο, παρατηρήθηκε αδυναμία του αλγορίθμου να ανιχνεύσει τις σκιές των συννέφων που βρίσκονταν σε υδάτινους πόρους (π.χ. λίμνη), λόγω των όμοιων ραδιομετρικών τιμών που εμφανίζουν στο κανάλι 6. Επιπλέον, παρουσιάστηκαν κάποιες λανθασμένες ενδείξεις ύπαρξης συννέφων περιμετρικά των λιμνών, το οποίο οφείλεται, επίσης, στον παραπάνω λόγο. Τέλος, η επιλογή για την παράλληλη αντικατάσταση των περιμετρικών εικονοστοιχείων στα όρια της σκιάς, σε «παράθυρο» 4x4, βελτίωσε αφενός «οπτικά» το τελικό αποτέλεσμα και αφετέρου οδήγησε σε μια πιο ομαλοποιημένη και ραδιομετρικά ορθή εικόνα.
- Επιπλέον, παρατηρήθηκε πως η αξιοποίηση αλγορίθμων ταξινόμησης, που χρησιμοποιούν διαδικασίες μηχανικής μάθησης, σε εφαρμογές ανίχνευσης μεταβολών δίνει ικανοποιητικά αποτελέσματα. Συγκεκριμένα, μέσω διαδικασιών

εξόρυξης δεδομένων επιτυγχάνεται η αναλυτική δημιουργία μοντέλου εκπαίδευσης, το οποίο μπορεί να χρησιμοποιηθεί κατά τη διαδικασία της *post classification analysis* για την ανίχνευση των μεταβολών μεταξύ δύο διαφορετικών χρονικών περιόδων με υψηλότερα ποσοστά ακρίβειας απ' ό,τι επιτυγχάνουν οι συμβατικές μέθοδοι ανίχνευσης μεταβολών.

- Τέλος, θετικά ήταν τα αποτελέσματα από την εφαρμογή του στατιστικού μοντέλου εκπαίδευσης στην ευρύτερη περιοχή της Θεσσαλίας, όπου αν εξαιρεθούν τα προβλήματα που παρουσιάστηκαν στις σκιές των συννέφων και σε συγκεκριμένους τύπους καλλιεργείων, η διαδικασία θεωρείται σε μεγάλο βαθμό επιτυχημένη.

Συνοψίζοντας, η μεθοδολογική προσέγγιση που ακολουθήθηκε, καθώς και τα αντίστοιχα προγράμματα που δημιουργήθηκαν, αντιμετώπισαν ικανοποιητικά τη διαχείριση και την ανάλυση μεγάλων γεωχωρικών δεδομένων με σκοπό την αξιοποίησή τους σε εφαρμογές Τηλεπισκόπησης. Στο επόμενο κεφάλαιο παρουσιάζονται προοπτικές εξέλιξης όσον αφορά αφενός στη μεθοδολογική προσέγγιση και αφετέρου στους αντίστοιχους αλγορίθμους που αναπτύχθηκαν.

7.2 Προοπτικές Εξέλιξης

Με την αξιοποίηση της συγκεκριμένης μεθοδολογίας καθίσταται δυνατή η δημιουργία υψηλής ανάλυσης και απαλλαγμένου από σύννεφα μωσαικού για το σύνολο της Ελλάδας, το οποίο μπορεί να χρησιμοποιηθεί τόσο ως υπόβαθρο για εφαρμογές γεωπληροφορικής όσο και για διαδικασίες ανάλυσης. Επιπλέον, με την παράλληλη χρήση δορυφορικών δεδομένων Modis και Landsat είναι δυνατή η κατασκευή ημερήσιων και απαλλαγμένων από σύννεφα μωσαικών εικόνων αξιοποιώντας χρονοσειρές δεδομένων. Με το συγκεκριμένο τρόπο, θα υπάρξει αύξηση των διαθέσιμων προς εκμετάλλευση δεδομένων, τα οποία θα συμβάλλουν στην ενίσχυση του στατιστικού μοντέλου εκπαίδευσης. Επιπλέον, προτείνεται η δημιουργία «τράπεζας» διανυσματικών δεδομένων για το σύνολο της χώρας και μετέπειτα η δημιουργία ενός στατιστικού μοντέλου εκμάθησης, το οποίο θα είναι ικανό να ταξινομήσει εικόνες που καλύπτουν το σύνολο της χώρας. Με τον τρόπο αυτό θα είναι δυνατή η αυτόματη παραγωγή ταξινομημένων εικόνων, χωρίς να απαιτείται η παρέμβαση του μελετητή, ανά 16 μέρες (χρονικό βήμα λήψης δεδομένων του Landsat) και η αξιοποίησή τους σε διάφορες εφαρμογές.

Όσον αφορά τον αλγόριθμο ανίχνευσης και αφαίρεσης συννέφων και σκιών προτείνεται η περαιτέρω έρευνα και βελτίωση των συνθηκών και των κατωφλίων που έχουν τεθεί, έτσι ώστε να αυξηθεί η ακρίβεια της ανίχνευσης. Επιπλέον, προτείνεται η περαιτέρω ανάπτυξή του έτσι ώστε να είναι δυνατή η παράλληλη εκμετάλλευση χρονοσειρών δεδομένων όπου σε συνδυασμό με την εφαρμογή

κατάλληλων κριτηρίων να βρίσκονται οι ιδανικές εικόνες προς αξιοποίηση. Με το συγκεκριμένο τρόπο, θα αυτοματοποιηθεί πλήρως ο αλγόριθμος και δεν θα απαιτείται η επέμβαση του μελετητή, η οποία μέχρι στιγμής είναι απαραίτητη για την εύρεση των κατάλληλων εικόνων που θα χρησιμοποιηθούν για την αντικατάσταση των συννέφων και σκιών.

Τέλος, μέσω της αξιοποίησης των αλγορίθμων καθώς και της μεθοδολογικής προσέγγισης που παρουσιάστηκε είναι δυνατή η δημιουργία ενός ολοκληρωμένου web gis σε τεχνολογία cloud, το οποίο θα έχει ως στόχο τη διαρκή παροχή εικόνων απαλλαγμένων από σύννεφα για την περιοχή ενδιαφέροντος του χρήστη. Επίσης, όμοια με το πρόγραμμα WELD που περιγράφηκε στην ενότητα 3.4.1, είναι δυνατή η κατασκευή μωσαικών σε εβδομαδιαία, μηνιαία και χρονιαία κλίμακα παρέχοντας δεδομένα για όλες τις δυνατές χρονικές περιόδους ανάλογα με την εκάστοτε εφαρμογή. Ακόμη, είναι δυνατή η παροχή, σε πραγματικό χρόνο (on the fly), ταξινομημένων εικόνων καθώς και εικόνων μεταβολών για την αντίστοιχη περιοχή μελέτης που ορίζει ο χρήστης (on demand) χρησιμοποιώντας το στατιστικό μοντέλο εκμάθησης που θα έχει προκύψει από το σύνολο της Ελλάδας. Με αυτόν τον τρόπο, θα καθίσταται εφικτή η αυτόματη ταξινόμηση οποιασδήποτε περιοχής που ορίζει ο χρήστης, παρέχοντας άμεσα προϊόντα υψηλής ακρίβειας, χωρίς να απαιτείται η αλληλεπίδραση του.

BIBΛΙΟΓΡΑΦΙΑ

Ξένη Βιβλιογραφία

IBM. (2011). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data.*

Altman, N. (1992). *An introduction to kernel and nearest-neighbor non parametric regression.*

American Geophysical Union. (2002). *Near-line Archive Data Mining at the Goddard Distributed Active Archive Center.*

Bhatt, A. (2012). *A review of change detection techniques.* Indian Institute of Technology.

Boser, Guyon, & Vapnik. (1992). *A training algorithm for optimal margin classifiers.*

Campbell, J. B., & Wynne, R. H. (2011). *Introduction to Remote Sensing Fifth Edition.*

Carlin, S., & Curran, K. (2012). *Cloud Computing Technologies. International Journal of Cloud Computing and Services Science .*

Christopher, D. M., & et al. (2008). *LectureIntroduction to information retrieval.*

Cunningham, P., & Delany, S. J. (2007). *K-nearest neighbor classifiers.*

Dasgupta, P. A. (2013). *Big data: The future is in analytics.* Ανάκτηση από <http://geospatialworld.net/Paper/Cover-Stories/ArticleView.aspx?aid=30512>

Decker, A. (2013). *Change detection process and techniques.*

Dietterich, G, T., Bakiri, & Ghulum. (1995). *Solving multiclass learning problems via error-correcting output codes .*

DODS: Distributed Oceanographic Data System. . Ανάκτηση 2014, από http://dods.ipsl.jussieu.fr/prism/wp4a/WP4a_prism_knmi.pdf

Doulaverakis, Ch., Tzotsos, A., Tsampoulatidis, I., Gerontidis, N., Argyridis, A., Karantzalos, K., Kompatsiaris, I., Argialas. D., 2014. GNORASI: A Modular Knowledge-based Platform for Object-based Image Analysis, In GEOgraphic Object Based Image Analysis (GEOBIA), Thessaloniki, 21-23 May 2014.

Duan, & Keerthi. (2005). *Lecture notes in computer science: Multiple Classifier Systems.*

Earth Imaging Journal. (2013). *Managing Big Data: NASA Tackles Complex Data Challenges.* Ανάκτηση από <http://eijournal.com/2013/managing-big-data>

ESA. (2013). *Big Data from Space-Event.*

ESRI . (2011). *Geospatial Visualization and Big Data.* Ανάκτηση από www.esri.com/products/technology-topics/big-data/geospatial-uses

Everitt, Landau, Leese, & Stahl. (2011). *Miscellaneous clustering methods, in cluster analysis, 5th edition.*

Gartner Inc. (2011). *Gartner Says Solving ‘Big Data’ Challenge Involves More Than Just Managing Volumes of Data .* Gartner Press Release.

Hallahan, N., & Prepperneau, C. (2013). *Cloud detection and removal techniques for Landsat 8 Imagery.*

- Hart, P. (1968).** *The condensed nearest neighbor rule.* IEEE.
- Herold, M. (2010).** *Change detection and time series analysis.* Wageningen University.
- Hsu, Chih-Wei, & Lin. (2002).** *A comparison of methods for multiclass support vector machines.*
- IDC. (2011).** *IDC's Digital Universe Study, sponsored by EMC, June 2011: Big Data Analytics.* Ανάκτηση 2011, από <http://www.idc.com/prodserv/FourPillars/bigData/index.jsp>
- Indiana University Pervasive Technology Institute. (2011).** Ανάκτηση από <http://grids.ucs.indiana.edu/ptliupages/publications/Where%20does%20all%20the%20data%20come%20from%20v7.pdf>
- Kaisler, Armour, Espinosa, & Money. (2013).** *Big Data: Issues and Challenges Moving Forward.*
- Kandare, K. (2010).** *The time series change detection methods of remote sensing.* isprs .
- Karantzalos, K. (2014).** Recent Advances on 2D and 3D Change Detection in Urban Environments from Remote Sensing Data, In: Helbich, M., Arsanjani, J. & Leitner, M. (Eds.). *Computational Approaches for Urban Environments*, Springer: New York (in press).
- Karmas A., Karantzalos K., Athanasiou S., (2014).** Online Analysis of Remote Sensing Data for Agricultural Applications, OSGeo's European Conference on Free and Open Source Software for Geospatial, FOSS4G-Europe, July 15 - 17, 2014.
- Leptoukh, G., Savtchenko, A., Kempler, S., Gopalan, A., & Berrick, S. (2011).** *MODIS Value-added Services at the GES DISC .*
- Martinuzzi, S., Gould, W. A., & Gonzalez, O. M. (2011).** *Creating cloud-free Landsat ETM+ Data Sets in Tropical Landscapes: Cloud and Cloud-shadow removal.* MODIS Online Visualization and Analysis System (MOVAS). Ανάκτηση 2014, από http://gcmd.nasa.gov/records/GES_DAAC_MOVAS.html
- MODIS Terra and Aqua Daily Level-3 Data Atmosphere Daily Global 1X1 Degree Products.* Ανάκτηση 2014, από http://gdata1.sci.gsfc.nasa.gov/daac-bin/G3/gui.cgi?instance_id=MODIS_DAILY_L3
- MODIS Azure.* Ανάκτηση 2014, από <http://research.microsoft.com/en-us/projects/modisazure/default.aspx>
- Nocedal, Jorge, Wright, & Stephen. (2006).** *Numerical Optimization, 2nd edition.* Springer .
- Paul C. Zikopoulos, P. C., & Al, e. (2013).** *Harness the Power of Big Data.* McGraw Hill.
- Platt, John, Cristianini, & Shawe-Taylor. (2000).** *Large margin DAGs for multiclass classification.*
- Powers, & David, M. W. (2011).** *Evaluation: from precision, recall and f-factor to ROC, informedness, markedness & correlation.*

- Prakash, S. (2012).** *Storing and Querying Big Data in Hadoop (HDFS).*
- Rokach, L. (2008).** *Data mining with decision trees: Theory and Application.*
- Roy, D., Ju, J., Kommareddy, I., & al, e. (2011).** *Web Enabled Landsat Data (WELD) Products.*
- S. Lhermitte, J. Verbesselt, W.W. Verstraeten, P. Coppin,** A comparison of time series similarity measures for classification and change detection of ecosystem dynamics, *Remote Sensing of Environment*, Volume 115, Issue 12, 15 December 2011, Pages 3129-3152, ISSN 0034-4257, <http://dx.doi.org/10.1016/j.rse.2011.06.020>.
- Singh, V. K. (2012).** *Big Data: Demystified.*
- Sinha, P., Kumar, L., & Reid, N. (2011).** *Seasonal land use / land cover mapping: accuracy comparison of various band combinations, . ISPRS.*
- Stroustrup Bjarne.** *C++, Εκδόσεις Κλειδάριθμος, 1997*
- Suthaharan, S. (2013).** *Big Data Classification: Problems and Challenges in Network Intrusion Prediction with Machine Learning .* Department of Computer Science . University of North Carolina at Greensboro .
- The 3 V's of Big Data and their Technologies, www.geektime.com.* Ανάκτηση 2014, από <http://www.geektime.com/2013/10/24/the-3-vs-of-big-data-and-their-technologies/>
- Thirumuruganathan, S. (2010).** *A detailed introduction to K-Nearest Neighbor Algorithm.*
- Tzotsos A., Karantzalos K., Argialas D. (2014).** Multiscale Segmentation and Classification of Remote Sensing Imagery with Advanced Edge and Scale-Space Features, in *Scale Issues in Remote Sensing* (ed Q. Weng), John Wiley & Sons, Inc., Hoboken, New Jersey. doi: 10.1002/9781118801628.ch09.
- Tzotsos A., Karantzalos K., Argialas D.** 2011. Object Based Image Analysis through Nonlinear Scale Space Filtering, *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 66, Issue 1, pp. 2-16.
- Udelhoven, T. (2012).** *Big data in Environmental Remote Sensing Challenges and Chances.* University of Trier, Environmental and Geoinformatics Department.
- University of California, Berkeley. (2012).** *Big Data Analytics: Making Data Work.* Ανάκτηση από http://fisheritcenter.haas.berkeley.edu/Big_Data/index.html
- Vapnik, V., & Cortes, C. (1995).** *Support-vector networks, Machine Learning.*
- Wong, P. C., Shen, H. W., Johnson, C. R., Chen, C., & Ross, R. B. (2012).** *The Top 10 Challenges in Extreme-Scale Visual Analytics.* IEEE: Computer Graphics and Applications.
- Zhe Zhu, Curtis E. Woodcock,** Continuous change detection and classification of land cover using all available Landsat data, *Remote Sensing of Environment*, Volume 144, 25 March 2014, Pages 152-171, ISSN 0034-4257, <http://dx.doi.org/10.1016/j.rse.2014.01.011>.

Ελληνική Βιβλιογραφία

Αργιαλάς, Δ. Π. (1998). *Ψηφιακή Τηλεπισκόπηση*. Αθήνα: Εθνικό Μετσόβιο Πολυτεχνείο.

Αργιαλάς, Δ. Π., Τζώτσος, Α., & Ιωσηφίδης, Χ. (2010). *Σύγχρονες τεχνολογίες τηλεπισκόπησης για την ανίχνευση, καταγραφή, αποτίμηση πυρκαγιών και προστασία των πληγέντων περιοχών*.

Αργιαλάς, Δημήτρης Π. (1999). *Φωτοερμηνεία-Τηλεπισκόπηση*. Αθήνα: Εθνικό Μετσόβιο Πολυτεχνείο.

Καρτάλης, Κ., & Φείδας, Χ. (2007). *Αρχές και Εφαρμογές Δορυφορικής Τηλεπισκόπησης*. Β. Γκιούρδας Εκδοτική.

Ιστοχώροι

<http://el.wikipedia.org/wiki/Τηλεπισκόπηση>, Ανάκτηση 2014.

http://en.wikipedia.org/wiki/Big_data, Ανάκτηση 2014.

http://en.wikipedia.org/wiki/Principal_component_analysis, Ανάκτηση 2014.

http://en.wikipedia.org/wiki/Statistical_classification, Ανάκτηση 2014.

http://en.wikipedia.org/wiki/Support_vector_machine, Ανάκτηση 2014.

<http://hadoop.apache.org>, Ανάκτηση 2014.

http://landsat.usgs.gov/about_project_descriptions.php, Ανάκτηση 2014.

<http://scikit-learn.org/stable/modules/svm.html>, Ανάκτηση 2014.

<http://scikit-learn.org/stable/modules/tree.html>, Ανάκτηση 2014.

<https://www.brockmann-consult.de/cms/web/beam/>, Ανάκτηση 2014.

<http://www.earthlook.org/index.html>, Ανάκτηση 2014.

<https://www.earthserver.eu>, Ανάκτηση 2014.

<http://www.ecognition.com>, Ανάκτηση 2014.

<http://www.exelisvis.com/ProductsServices/ENVI/ENVI.aspx>, Ανάκτηση 2014.

<https://www.gnu.org/philosophy/free-sw.html>, Ανάκτηση 2014.

<https://www.grass.osgeo.org>, Ανάκτηση 2014.

<http://www.hexagongeospatial.com/products/ERDAS-IMAGINE/Details.aspx>, Ανάκτηση 2014.

<https://www.opticks.org/confluence/display/opticks/Welcome+To+Opticks>, Ανάκτηση 2014.

<http://www.pcigeomatics.com>, Ανάκτηση 2014

<https://www.qgis.org/en/site/>, Ανάκτηση 2014

<http://www.rasdaman.com>, Ανάκτηση 2014.

<https://www.saga-gis.org/en/index.html>, Ανάκτηση 2014.

<https://www.statsoft.com/Textbook/Data-Mining-Techniques>, Ανάκτηση 2014.

ΠΑΡΑΡΤΗΜΑ Ι

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1.1: Χαρακτηριστικά ελεύθερων λογισμικών Τηλεπισκόπησης	7
Πίνακας 1.2: Χαρακτηριστικά εμπορικών λογισμικών Τηλεπισκόπησης	8
Πίνακας 2.1: Δεδομένα διαγνωστικής απεικόνισης στις ΗΠΑ	15
Πίνακας 3.1: Δέκτες και Κανάλια Landsat 8	36
Πίνακας 3.2: Συνδυασμοί Καναλιών Landsat 8	36
Πίνακας 3.3: Χαρακτηριστικά Δεδομένων Landsat 8	38
Πίνακας 3.4: Παραδείγματα εφαρμογών ανά τομέα προγράμματος Landsat	41
Πίνακας 3.5: Κριτήρια Σύνθεσης Συγκρίνοντας δύο διαδοχικές λήψεις εικονοστοιχείου	44
Πίνακας 3.6: Τελικά Προϊόντα Προγράμματος WELD	44
Πίνακας 4.1: Ανασκόπηση μεθόδων ανίχνευσης μεταβολών με άλγεβρα εικόνας ...	53
Πίνακας 4.2: Ανασκόπηση μεθόδων ανίχνευσης μεταβολών με εφαρμογή μετασχηματισμού	55
Πίνακας 4.3: Σύγκριση επιβλεπόμενης και μη επιβλεπόμενης ταξινόμησης για ανίχνευση μεταβολών	57
Πίνακας 4.4: Ελεύθερα λογισμικά για διαδικασίες data mining	59
Πίνακας 4.5: Πλεονεκτήματα-Μειονεκτήματα SVM	66
Πίνακας 4.6: Πλεονεκτήματα-Μειονεκτήματα decision tree	69
Πίνακας 4.7: Πλεονεκτήματα-Μειονεκτήματα decision tree	73
Πίνακας 4.8: Συγκεντρωτικά αποτελέσματα σειράς ταξινομήσεων παραδειγμάτων	89
Πίνακας 5.1: Τιμές εικονοστοιχείων καναλιού BQA για ύπαρξη συννέφου	98
Πίνακας 5.2: Αριθμοί Bit καναλιού BQA για ύπαρξη συννέφου	98
Πίνακας 6.1: Επιλεγμένες σκηνές Landsat 8	109
Πίνακας 6.2: Έλεγχος ακρίβειας σημείων ελέγχου για σκηνές 184/32	111
Πίνακας 6.3: Έλεγχος ακρίβειας σημείων ελέγχου για σκηνές 184/33	111
Πίνακας 6.4: Έλεγχος ακρίβειας σημείων ελέγχου για επικαλυπτόμενο τμήμα	112
Πίνακας 6.5: Στατιστικά εικόνας αναφοράς πριν την εφαρμογή των ραδιομετρικών διορθώσεων	114
Πίνακας 6.6: Στατιστικά εικόνας αναφοράς μετά την εφαρμογή των ραδιομετρικών διορθώσεων	115
Πίνακας 6.7: Παράδειγμα εφαρμογής αλγορίθμου pansharpening του orfeotoolbox	116
Πίνακας 6.8: Στατιστικά εικόνας αναφοράς μετά την εφαρμογή του pansharpening	116
Πίνακας 6.9: Παραδείγματα κατηγοριοποίησης τύπων καλλιεργειών	119
Πίνακας 6.10: Θεματικές κατηγορίες για το μοντέλο εκπαίδευσης	122
Πίνακας 6.11: Αξιολόγηση αλγορίθμου ανίχνευσης συννέφων και σκιών	126

Πίνακας 6.12: Παραδείγματα εφαρμογής του αλγορίθμου ανίχνευσης-αφαίρεσης συννέφων/σκιών	128
Πίνακας 6.13: Στατιστικά και εμβοδομέτρηση από την ταξινόμηση.....	132
Πίνακας 6.14: Λανθασμένες ενδείξεις στην κατηγορία των καλλιεργειών	133
Πίνακας 6.15: Στατιστικά και εμβοδομέτρηση από τη νέα ταξινόμηση.....	136
Πίνακας 6.16: Μέτρια και μεγάλη μεταβολή στις χρήσεις γης για τα έτη 2013-2014	139

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Ορισμός 1 (V^3) των χαρακτηριστικών των Big Data	12
Εικόνα 2.2: Ορισμός 2 (V^3) των χαρακτηριστικών των Big Data	13
Εικόνα 2.3: Ορισμός 3 (V^3) των χαρακτηριστικών των Big Data	13
Εικόνα 2.4: Αρχιτεκτονική DODS	18
Εικόνα 2.5: MOVAS platform	20
Εικόνα 2.6: Αρχιτεκτονική NADM	21
Εικόνα 2.7: Στάδια MODISAzure	21
Εικόνα 2.8: Αρχιτεκτονική MODISAzure	21
Εικόνα 2.9: Εφαρμογή STARFM algorithm σε NDVI	22
Εικόνα 2.10: Εφαρμογή STARFM algorithm σε time series reflectance.....	22
Εικόνα 2.11: Συγγώνευση laser-scanning and hyperspectral reflective data.....	23
Εικόνα 2.12: Chain classifications	24
Εικόνα 2.13: Παράδειγμα Αρχιτεκτονικής Hadoop	27
Εικόνα 2.14: Παράδειγμα Αρχιτεκτονικής Rasdaman	29
Εικόνα 2.15: IBM Big Data Platform	30
Εικόνα 3.1: Έγχρωμο σύνθετο 4-3-2 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία	38
Εικόνα 3.2: Έγχρωμο σύνθετο 7-6-4 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία	39
Εικόνα 3.3: Έγχρωμο σύνθετο 5-4-3 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία	39
Εικόνα 3.4: Έγχρωμο σύνθετο 5-6-2 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία	40
Εικόνα 3.5: Έγχρωμο σύνθετο 6-5-2 στην περιοχή της λίμνης Κάρλας στη Θεσσαλία	40
Εικόνα 3.6: Αρχιτεκτονική Υπολογιστικού Συστήματος WELD.....	45
Εικόνα 3.7: Παράδειγμα προϊόντος WELD CONUS σε εβδομαδιαία κλίμακα (15-21 Ιουλίου, 2008).....	46
Εικόνα 3.8: Παράδειγμα προϊόντος WELD CONUS σε μηνιαία κλίμακα (Ιούλιος, 2008)	46
Εικόνα 3.9: Παράδειγμα προϊόντος WELD CONUS σε χρονιαία κλίμακα (έτος 2008)	47
Εικόνα 4.1: Παράδειγμα εφαρμογής πράξης αφαίρεσης	51
Εικόνα 4.2: Παράδειγμα εφαρμογής πράξης λόγου	52
Εικόνα 4.3: Παράδειγμα CVA σε δύο κανάλια	53
Εικόνα 4.4: Αλγεβρική πράξη CVA.....	53
Εικόνα 4.5: Ανίχνευση μεταβολών με PCA.....	54
Εικόνα 4.6: Παράδειγμα ταξινομητή SVM	61
Εικόνα 4.7: Πρόβλημα βελτιστοποίησης περιθωρίου σε γραμμικό ταξινομητή SVM63	
Εικόνα 4.8: Δέντρο απόφασης για την εύρεση συγκεκριμένων επιθέτων	67

Εικόνα 4.9: Δέντρο απόφασης για την ταξινόμηση εικονοστοιχείων.....	68
Εικόνα 4.10: Ψευδοκώδικας αλγορίθμου τερματισμού επέκτασης δέντρου απόφασης	69
Εικόνα 4.11: Παράδειγμα εφαρμογής αλγορίθμου KNN.....	71
Εικόνα 4.12: Ψευδοκώδικας αλγορίθμου KNN	71
Εικόνα 4.13: Δεδομένα εκπαίδευσης χωρίς την εφαρμογή του CNN.....	72
Εικόνα 4.14: 1NN ταξινόμηση χωρίς εφαρμογή του CNN	72
Εικόνα 4.15: Δεδομένα εκπαίδευσης με εφαρμογή του CNN	72
Εικόνα 4.16: 1NN ταξινόμηση με εφαρμογή του CNN.....	72
Εικόνα 4.17: Εικόνα QB_1_ortho.tif και πολύγωνα από QuickBird 4-καναλιών	75
Εικόνα 4.18: Εικόνα QB_2_ortho.tif και πολύγωνα από QuickBird 4-καναλιών	76
Εικόνα 4.19: Ταξινομημένη εικόνα QB_2_ortho.tif με αλγόριθμο svm	80
Εικόνα 4.20: Ταξινομημένη εικόνα QB_2_ortho.tif με αλγόριθμο decision tree	84
Εικόνα 4.21: Ταξινομημένη εικόνα QB_2_ortho.tif με αλγόριθμο k-nearest neighbors	87
Εικόνα 5.1: Ανίχνευση σκιών συννέφου με δημιουργία ζώνης	100
Χάρτης 6.1: Γεωγραφική θέση Περιφέρειας Θεσσαλίας	107
Χάρτης 6.2: Χρήσεις γης Περιφέρειας Θεσσαλίας	107
Εικόνα 6.1: Σημεία ελέγχου για τις σκηνές 184/32	110
Εικόνα 6.2: Σημεία ελέγχου για τις σκηνές 184/33	110
Εικόνα 6.3: Σημεία ελέγχου για το επικαλυπτόμενο τμήμα	112
Εικόνα 6.4: Μωσαϊκό Θεσσαλίας για την ημερομηνία: 25 Ιουνίου 2013.....	117
Εικόνα 6.5: Υποπεριοχή 1 – Εύρύτερη περιοχή λίμνης Κάρλας.....	118
Εικόνα 6.6: Υποπεριοχή 2 – Εύρύτερη περιοχή πόλης Λάρισας	118
Εικόνα 6.7: Υποπεριοχή 3 – Εύρύτερη περιοχή λίμνης Βεγορίτιδας	118
Εικόνα 6.8: Υποπεριοχή 4 – Εύρύτερη περιοχή λίμνης Τριχωνίδας	118
Εικόνα 6.9: Περιοχή για την εφαρμογή του μοντέλου εκπαίδευσης – Ευρύτερη περιοχή του οικισμού της Καρδίτσας 25/6/13.....	124
Εικόνα 6.10: Περιοχή για την αντικατάσταση των συννέφων της εικόνας αναφοράς – Ευρύτερη περιοχή του οικισμού της Καρδίτσας 10/7/13.....	125
Εικόνα 6.11: Μάσκα συννέφων και σκιών	125
Εικόνα 6.12: Τελική εικόνα μετά την εφαρμογή του αλγορίθμου ανίχνευσης και αφαίρεσης συννέφων και σκιών	127
Εικόνα 6.13: Ταξινόμηση περιοχής μελέτης με τον αλγόριθμο SVM	130
Εικόνα 6.14: Ταξινομημένη εικόνα μετά τη συγχώνευση των όμοιων κλάσεων	131
Εικόνα 6.15: Ταξινομημένη εικόνα μετά την εφαρμογή του φίλτρου MajorityVoting	132
Εικόνα 6.16: Περιοχή μελέτης, ημερομηνία: 27/6/14	135
Εικόνα 6.17: Ταξινόμηση περιοχής μελέτης, ημερομηνία: 27/6/14.....	136
Εικόνα 6.18: Ανίχνευση μεταβολών στην περιοχή μελέτης μεταξύ των ετών 2013-2014	138

Εικόνα 6.19: Ταξινόμηση της ευρύτερης περιοχής της Θεσσαλίας, 06/2013	141
Εικόνα 6.20: TMS μωσαϊκού Θεσσαλίας	143
Εικόνα 6.21: TMS ταξινόμηση μωσαϊκού Θεσσαλίας	144
Εικόνα 6.22: Οριοθέτηση περιοχής μελέτης	145
Εικόνα 6.23: Περιοχή μελέτης απαλλαγμένη από σύννεφα.....	145
Εικόνα 6.24: Ταξινόμηση περιοχής μελέτης.....	146

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 2.1: Αύξηση ψηφιακού περιεχομένου	14
Διάγραμμα 3.1: Σύγκριση Καναλιών Landsat 7-Landsat 8	37
Διάγραμμα 3.2: WELD - Διαδικασία Παραγωγής Τελικού Προϊόντος	43
Διάγραμμα 4.1: Post classification analysis in change detection	56
Διάγραμμα 4.2: Unsupervised classification analysis in change detection	57
Διάγραμμα 5.1: Προτεινόμενη Μεθοδολογία	93
Διάγραμμα 5.2: Λογικό διάγραμμα αλγορίθμου ανίχνευσης και αφαίρεσης των συννέφων και των σκιών τους σε δορυφορικές εικόνες Landsat 8	103
Διάγραμμα 6.1: Αρχικό ιστόγραμμα εικόνας	114
Διάγραμμα 6.2: Ιστόγραμμα εικόνας με τιμές ανακλαστικότητας	114
Διάγραμμα 6.3: Ιστόγραμμα εικόνας μετά την εφαρμογή του pansharpening	116
Διάγραμμα 6.4: Αρχικό ιστόγραμμα περιοχής μελέτης	127
Διάγραμμα 6.5: Ιστόγραμμα μετά την αφαίρεση των συννέφων	127

Orfeo Toolbox – Train Images Classifier

```

1  /*=====
2  Program: ORFEO Toolbox
3  Language: C++
4  Date:   $Date$
5  Version: $Revision$
6
7  Copyright (c) Centre National d'Etudes Spatiales. All rights reserved.
8  See OTBCopyright.txt for details.
9
10 This software is distributed WITHOUT ANY WARRANTY; without even
11 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
12 PURPOSE. See the above copyright notices for more information.
13
14 =====*/
15
16 #include "otbTrainImagesClassifier.h"
17
18 namespace otb
19 {
20 namespace Wrapper
21 {
22
23 void TrainImagesClassifier::DoInit()
24 {
25   SetName("TrainImagesClassifier");
26   SetDescription(
27     "Train a classifier from multiple pairs of images and training vector data.");
28
29   //Documentation
30   SetDocName("Train a classifier from multiple images");
31   SetDocLongDescription(
32     "This application performs a classifier training from multiple pairs of input images and training vector
33     data. "
34     "Samples are composed of pixel values in each band optionally centered and reduced using an XML
35     statistics file produced by "
36     "the ComputeImagesStatistics application.\n The training vector data must contain polygons with a
37     positive integer field "
38     "representing the class label. The name of this field can be set using the \"Class label field\"
39     parameter. Training and validation "
40     "sample lists are built such that each class is equally represented in both lists. One parameter allows
41     to control the ratio "
42     "between the number of samples in training and validation sets. Two parameters allow to manage the
43     size of the training and "
44     "validation sets per class and per image.\n Several classifier parameters can be set depending on the
45     chosen classifier. In the "
46     "validation process, the confusion matrix is organized the following way: rows = reference labels,
47     columns = produced labels. "

```

```

48 "In the header of the optional confusion matrix output file, the validation (reference) and predicted
49 (produced) class labels"
50 " are ordered according to the rows/columns of the confusion matrix.\n This application is based on
51 LibSVM and on OpenCV Machine Learning "
52 "classifiers, and is compatible with OpenCV 2.3.1 and later.");
53 SetDocLimitations("None");
54 SetDocAuthors("OTB-Team");
55 SetDocSeeAlso("OpenCV documentation for machine learning
56 http://docs.opencv.org/modules/ml/doc/ml.html ");
57
58 AddDocTag(Tags::Learning);
59
60 //Group IO
61 AddParameter(ParameterType_Group, "io", "Input and output data");
62 SetParameterDescription("io", "This group of parameters allows to set input and output data.");
63 AddParameter(ParameterType_InputImageList, "io.il", "Input Image List");
64 SetParameterDescription("io.il", "A list of input images.");
65 AddParameter(ParameterType_InputVectorDataList, "io.vd", "Input Vector Data List");
66 SetParameterDescription("io.vd", "A list of vector data to select the training samples.");
67 AddParameter(ParameterType_InputFilename, "io.imstat", "Input XML image statistics file");
68 MandatoryOff("io.imstat");
69 SetParameterDescription("io.imstat",
70 "Input XML file containing the mean and the standard deviation of the input images.");
71 AddParameter(ParameterType_OutputFilename, "io.confmatout", "Output confusion matrix");
72 SetParameterDescription("io.confmatout", "Output file containing the confusion matrix (.csv
73 format).");
74 MandatoryOff("io.confmatout");
75 AddParameter(ParameterType_OutputFilename, "io.out", "Output model");
76 SetParameterDescription("io.out", "Output file containing the model estimated (.txt format).");
77
78 //Elevation
79 ElevationParametersHandler::AddElevationParameters(this, "elev");
80
81 //Group Sample list
82 AddParameter(ParameterType_Group, "sample", "Training and validation samples parameters");
83 SetParameterDescription("sample",
84 "This group of parameters allows to set training and validation sample lists parameters.");
85
86 AddParameter(ParameterType_Int, "sample.mt", "Maximum training sample size per class");
87 //MandatoryOff("mt");
88 SetDefaultParameterInt("sample.mt", 1000);
89 SetParameterDescription("sample.mt", "Maximum size per class (in pixels) of the training sample list
90 (default = 1000) (no limit = -1).
91 If equal to -1, then the maximal size of the available training sample list per class will be equal to the
92 surface area of the smallest class multiplied by the training sample ratio.");
93 AddParameter(ParameterType_Int, "sample.mv", "Maximum validation sample size per class");
94 //MandatoryOff("mv");
95 SetDefaultParameterInt("sample.mv", 1000);

```

```

96 SetParameterDescription("sample.mv", "Maximum size per class (in pixels) of the validation sample
97 list (default = 1000) (no limit = -1).
98 If equal to -1, then the maximal size of the available validation sample list per class will be equal to
99 the surface area of the smallest class multiplied by the validation sample ratio.");
100
101 AddParameter(ParameterType_Empty, "sample.edg", "On edge pixel inclusion");
102 SetParameterDescription("sample.edg",
103 "Takes pixels on polygon edge into consideration when building training and validation samples.");
104 MandatoryOff("sample.edg");
105
106 AddParameter(ParameterType_Float, "sample.vtr", "Training and validation sample ratio");
107 SetParameterDescription("sample.vtr",
108 "Ratio between training and validation samples (0.0 = all training, 1.0 = all validation) (default =
109 0.5).");
110 SetParameterFloat("sample.vtr", 0.5);
111
112 AddParameter(ParameterType_String, "sample.vfn", "Name of the discrimination field");
113 SetParameterDescription("sample.vfn", "Name of the field used to discriminate class labels in the
114 input vector data files.");
115 SetParameterString("sample.vfn", "Class");
116
117 AddParameter(ParameterType_Choice, "classifier", "Classifier to use for the training");
118 SetParameterDescription("classifier", "Choice of the classifier to use for the training.");
119
120 //Group LibSVM
121 InitLibSVMParams();
122
123 #ifdef OTB_USE_OPENCV
124 InitSVMParams();
125 InitBoostParams();
126 InitDecisionTreeParams();
127 InitGradientBoostedTreeParams();
128 InitNeuralNetworkParams();
129 InitNormalBayesParams();
130 InitRandomForestsParams();
131 InitKNNParams();
132 #endif
133
134 AddRANDParameter();
135 //Doc example parameter settings
136 SetDocExampleParameterValue("io.il", "QB_1_ortho.tif");
137 SetDocExampleParameterValue("io.vd", "VectorData_QB1.shp");
138 SetDocExampleParameterValue("io.imstat", "EstimateImageStatisticsQB1.xml");
139 SetDocExampleParameterValue("sample.mv", "100");
140 SetDocExampleParameterValue("sample.mt", "100");
141 SetDocExampleParameterValue("sample.vtr", "0.5");
142 SetDocExampleParameterValue("sample.edg", "false");
143 SetDocExampleParameterValue("sample.vfn", "Class");
144 SetDocExampleParameterValue("classifier", "libsvm");

```



```

145 SetDocExampleParameterValue("classifier.libsvm.k", "linear");
146 SetDocExampleParameterValue("classifier.libsvm.c", "1");
147 SetDocExampleParameterValue("classifier.libsvm.opt", "false");
148 SetDocExampleParameterValue("io.out", "svmModelQB1.txt");
149 SetDocExampleParameterValue("io.confmatout", "svmConfusionMatrixQB1.csv");
150 }
151
152 void TrainImagesClassifier::DoUpdateParameters()
153 {
154 //Nothing to do here : all parameters are independent
155 }
156
157 void TrainImagesClassifier::LogConfusionMatrix(ConfusionMatrixCalculatorType* confMatCalc)
158 {
159 ConfusionMatrixCalculatorType::ConfusionMatrixType matrix = confMatCalc->GetConfusionMatrix();
160
161 //Compute minimal width
162 size_t minwidth = 0;
163
164 for (unsigned int i = 0; i < matrix.Rows(); i++)
165 {
166 for (unsigned int j = 0; j < matrix.Cols(); j++)
167 {
168 std::ostringstream os;
169 os << matrix(i, j);
170 size_t size = os.str().size();
171
172 if (size > minwidth)
173 {
174 minwidth = size;
175 }
176 }
177 }
178
179 MapOfIndicesType mapOfIndices = confMatCalc->GetMapOfIndices();
180
181 MapOfIndicesType::const_iterator it = mapOfIndices.begin();
182 MapOfIndicesType::const_iterator end = mapOfIndices.end();
183
184 for (; it != end; ++it)
185 {
186 std::ostringstream os;
187 os << "[" << it->second << "]";
188
189 size_t size = os.str().size();
190 if (size > minwidth)
191 {
192 minwidth = size;
193 }

```

```

194 }
195
196 //Generate matrix string, with 'minwidth' as size specifier
197 std::ostringstream os;
198
199 //Header line
200 for (size_t i = 0; i < minwidth; ++i)
201     os << " ";
202     os << " ";
203
204     it = mapOfIndices.begin();
205     end = mapOfIndices.end();
206     for (; it != end; ++it)
207     {
208         os << "[" << it->second << "]" << " ";
209     }
210
211     os << std::endl;
212
213     //Each line of confusion matrix
214     for (unsigned int i = 0; i < matrix.Rows(); i++)
215     {
216         ConfusionMatrixCalculatorType::ClassLabelType label = mapOfIndices[i];
217         os << "[" << std::setw(minwidth - 2) << label << "]" << " ";
218         for (unsigned int j = 0; j < matrix.Cols(); j++)
219         {
220             os << std::setw(minwidth) << matrix(i, j) << " ";
221         }
222         os << std::endl;
223     }
224
225     otbAppLogINFO("Confusion matrix (rows = reference labels, columns = produced labels):\n" <<
226     os.str());
227 }
228
229 void TrainImagesClassifier::Classify(ListSampleType::Pointer validationListSample,
230 LabelListSampleType::Pointer predictedList)
231 {
232     //Classification
233     ModelPointerType model =
234     MachineLearningModelFactoryType::CreateMachineLearningModel(GetParameterString("io.out"),
235     MachineLearningModelFactoryType::ReadMode);
236
237     if (model.IsNull())
238     {
239         otbAppLogFATAL(<< "Error when loading model " << GetParameterString("io.out"));
240     }
241
242     model->Load(GetParameterString("io.out"));

```

```

243 model->SetInputListSample(validationListSample);
244 model->SetTargetListSample(predictedList);
245 model->PredictAll();
246 }
247
248 void TrainImagesClassifier::DoExecute()
249 {
250     GetLogger()->Debug("Entering DoExecute\n");
251     //Create training and validation for list samples and label list samples
252     ConcatenateLabelListSampleFilterType::Pointer concatenateTrainingLabels =
253     ConcatenateLabelListSampleFilterType::New();
254     ConcatenateListSampleFilterType::Pointer concatenateTrainingSamples =
255     ConcatenateListSampleFilterType::New();
256     ConcatenateLabelListSampleFilterType::Pointer concatenateValidationLabels =
257     ConcatenateLabelListSampleFilterType::New();
258     ConcatenateListSampleFilterType::Pointer concatenateValidationSamples =
259     ConcatenateListSampleFilterType::New();
260
261     MeasurementType meanMeasurementVector;
262     MeasurementType stddevMeasurementVector;
263
264     //-----
265     //Load measurements from images
266     unsigned int nbBands = 0;
267     //Iterate over all input images
268
269     FloatVectorImageListType* imageList = GetParameterImageList("io.il");
270     VectorDataListType* vectorDataList = GetParameterVectorDataList("io.vd");
271
272     vdreproj = VectorDataReprojectionType::New();
273
274     //Iterate over all input images
275     for (unsigned int imgIndex = 0; imgIndex < imageList->Size(); ++imgIndex)
276     {
277         FloatVectorImageType::Pointer image = imageList->GetNthElement(imgIndex);
278         image->UpdateOutputInformation();
279
280         if (imgIndex == 0)
281         {
282             nbBands = image->GetNumberOfComponentsPerPixel();
283         }
284
285         //read the Vectordata
286         VectorDataType::Pointer vectorData = vectorDataList->GetNthElement(imgIndex);
287         vectorData->Update();
288
289         vdreproj->SetInputImage(image);
290         vdreproj->SetInput(vectorData);
291         vdreproj->SetUseOutputSpacingAndOriginFromImage(false);

```

```

292
293 //Setup the DEM Handler
294 otb::Wrapper::ElevationParametersHandler::SetupDEMHandlerFromElevationParameters(this,
295 "elev");
296
297 vdreproj->Update();
298
299 //Sample list generator
300 ListSampleGeneratorType::Pointer sampleGenerator = ListSampleGeneratorType::New();
301
302 sampleGenerator->SetInput(image);
303 sampleGenerator->SetInputVectorData(vdreproj->GetOutput());
304
305 sampleGenerator->SetClassKey(GetParameterString("sample.vfn"));
306 sampleGenerator->SetMaxTrainingSize(GetParameterInt("sample.mt"));
307 sampleGenerator->SetMaxValidationSize(GetParameterInt("sample.mv"));
308 sampleGenerator->SetValidationTrainingProportion(GetParameterFloat("sample.vtr"));
309
310 //take pixel located on polygon edge into consideration
311 if (IsParameterEnabled("sample.edg"))
312 {
313 sampleGenerator->SetPolygonEdgeInclusion(true);
314 }
315
316 sampleGenerator->Update();
317
318 //Concatenate training and validation samples from the image
319 concatenateTrainingLabels->AddInput(sampleGenerator->GetTrainingListLabel());
320 concatenateTrainingSamples->AddInput(sampleGenerator->GetTrainingListSample());
321 concatenateValidationLabels->AddInput(sampleGenerator->GetValidationListLabel());
322 concatenateValidationSamples->AddInput(sampleGenerator->GetValidationListSample());
323 }
324 //Update
325 concatenateTrainingSamples->Update();
326 concatenateTrainingLabels->Update();
327 concatenateValidationSamples->Update();
328 concatenateValidationLabels->Update();
329
330 if (concatenateTrainingSamples->GetOutput()->Size() == 0)
331 {
332 otbAppLogFATAL("No training samples, cannot perform SVM training.");
333 }
334
335 if (concatenateValidationSamples->GetOutput()->Size() == 0)
336 {
337 otbAppLogWARNING("No validation samples.");
338 }
339
340 if (IsParameterEnabled("io.imstat"))

```

```

341 {
342   StatisticsReader::Pointer statisticsReader = StatisticsReader::New();
343   statisticsReader->SetFileName(GetParameterString("io.imstat"));
344   meanMeasurementVector = statisticsReader->GetStatisticVectorByName("mean");
345   stddevMeasurementVector = statisticsReader->GetStatisticVectorByName("stddev");
346 }
347 else
348 {
349   meanMeasurementVector.SetSize(nbBands);
350   meanMeasurementVector.Fill(0.);
351   stddevMeasurementVector.SetSize(nbBands);
352   stddevMeasurementVector.Fill(1.);
353 }
354
355 //Shift scale the samples
356 ShiftScaleFilterType::Pointer trainingShiftScaleFilter = ShiftScaleFilterType::New();
357 trainingShiftScaleFilter->SetInput(concatenateTrainingSamples->GetOutput());
358 trainingShiftScaleFilter->SetShifts(meanMeasurementVector);
359 trainingShiftScaleFilter->SetScales(stddevMeasurementVector);
360 trainingShiftScaleFilter->Update();
361
362 ListSampleType::Pointer validationListSample=ListSampleType::New();
363
364 //Test if the validation test is empty
365 if ( concatenateValidationSamples->GetOutput()->Size() != 0 )
366 {
367   ShiftScaleFilterType::Pointer validationShiftScaleFilter = ShiftScaleFilterType::New();
368   validationShiftScaleFilter->SetInput(concatenateValidationSamples->GetOutput());
369   validationShiftScaleFilter->SetShifts(meanMeasurementVector);
370   validationShiftScaleFilter->SetScales(stddevMeasurementVector);
371   validationShiftScaleFilter->Update();
372   validationListSample = validationShiftScaleFilter->GetOutput();
373 }
374
375 ListSampleType::Pointer listSample;
376 LabelListSampleType::Pointer labelListSample;
377 //-----
378 //Balancing training sample (if needed)
379 //if (IsParameterEnabled("sample.b"))
380 //{
381   ///Balance the list sample.
382   //otbAppLogINFO("Number of training samples before balancing: " << concatenateTrainingSamples-
383   >GetOutput()->Size())
384   //BalancingListSampleFilterType::Pointer balancingFilter = BalancingListSampleFilterType::New();
385   //balancingFilter->SetInput(trainingShiftScaleFilter->GetOutput());
386   //balancingFilter->SetInputLabel(concatenateTrainingLabels->GetOutput());
387   //balancingFilter->SetBalancingFactor(GetParameterInt("sample.b"));
388   //balancingFilter->Update();
389   //listSample = balancingFilter->GetOutput();

```

```

390 //labelListSample = balancingFilter->GetOutputLabelSampleList();
391 //otbAppLogINFO("Number of samples after balancing: " << balancingFilter->GetOutput()->Size());
392
393 //}
394 //else
395 //{
396 listSample = trainingShiftScaleFilter->GetOutput();
397 labelListSample = concatenateTrainingLabels->GetOutput();
398 otbAppLogINFO("Number of training samples: " << concatenateTrainingSamples->GetOutput()-
399 >Size());
400 //}
401 //-----
402 //Split the data set into training/validation set
403 ListSampleType::Pointer trainingListSample = listSample;
404 LabelListSampleType::Pointer trainingLabeledListSample = labelListSample;
405
406 LabelListSampleType::Pointer validationLabeledListSample = concatenateValidationLabels-
407 >GetOutput();
408 otbAppLogINFO("Size of training set: " << trainingListSample->Size());
409 otbAppLogINFO("Size of validation set: " << validationListSample->Size());
410 otbAppLogINFO("Size of labeled training set: " << trainingLabeledListSample->Size());
411 otbAppLogINFO("Size of labeled validation set: " << validationLabeledListSample->Size());
412
413 //-----
414 //Estimate model
415 //-----
416 LabelListSampleType::Pointer predictedList = LabelListSampleType::New();
417 const std::string classifierType = GetParameterString("classifier");
418 if (classifierType == "libsvm")
419 {
420 TrainLibSVM(trainingListSample, trainingLabeledListSample);
421 }
422 #ifdef OTB_USE_OPENCV
423 else if (classifierType == "svm")
424 {
425 TrainSVM(trainingListSample, trainingLabeledListSample);
426 }
427 else if (classifierType == "boost")
428 {
429 TrainBoost(trainingListSample, trainingLabeledListSample);
430 }
431 else if (classifierType == "dt")
432 {
433 TrainDecisionTree(trainingListSample, trainingLabeledListSample);
434 }
435 else if (classifierType == "gbt")
436 {
437 TrainGradientBoostedTree(trainingListSample, trainingLabeledListSample);
438 }

```

```

439 else if (classifierType == "ann")
440 {
441   TrainNeuralNetwork(trainingListSample, trainingLabeledListSample);
442 }
443 else if (classifierType == "bayes")
444 {
445   TrainNormalBayes(trainingListSample, trainingLabeledListSample);
446 }
447 else if (classifierType == "rf")
448 {
449   TrainRandomForests(trainingListSample, trainingLabeledListSample);
450 }
451 else if (classifierType == "knn")
452 {
453   TrainKNN(trainingListSample, trainingLabeledListSample);
454 }
455 #endif
456
457 //-----
458 //Performances estimation
459 //-----
460 ListSampleType::Pointer performanceListSample=ListSampleType::New();
461 LabelListSampleType::Pointer performanceLabeledListSample=LabelListSampleType::New();
462
463 //Test the input validation set size
464 if(validationLabeledListSample->Size() != 0)
465 {
466   performanceListSample = validationListSample;
467   performanceLabeledListSample = validationLabeledListSample;
468 }
469 else
470 {
471   otbAppLogWARNING("The validation set is empty. The performance estimation is done using the
472   input training set in this case.");
473   performanceListSample = trainingListSample;
474   performanceLabeledListSample = trainingLabeledListSample;
475 }
476
477 Classify(performanceListSample, predictedList);
478
479 ConfusionMatrixCalculatorType::Pointer confMatCalc = ConfusionMatrixCalculatorType::New();
480
481 std::cout << "predicted list size == " << predictedList->Size() << std::endl;
482 std::cout << "validationLabeledListSample size == " << performanceLabeledListSample->Size() <<
483 std::endl;
484 confMatCalc->SetReferenceLabels(performanceLabeledListSample);
485 confMatCalc->SetProducedLabels(predictedList);
486
487 confMatCalc->Compute();

```



```

488
489 otbAppLogINFO("training performances");
490 LogConfusionMatrix(confMatCalc);
491
492 for (unsigned int itClasses = 0; itClasses < confMatCalc->GetNumberOfClasses(); itClasses++)
493 {
494     ConfusionMatrixCalculatorType::ClassLabelType classLabel = confMatCalc-
495     >GetMapOfIndices()[itClasses];
496
497     otbAppLogINFO("Precision of class [" << classLabel << "] vs all: " << confMatCalc-
498     >GetPrecisions()[itClasses]);
499     otbAppLogINFO("Recall of class  [" << classLabel << "] vs all: " << confMatCalc-
500     >GetRecalls()[itClasses]);
501     otbAppLogINFO(
502     "F-score of class  [" << classLabel << "] vs all: " << confMatCalc->GetFScores()[itClasses] << "\n");
503 }
504 otbAppLogINFO("Global performance, Kappa index: " << confMatCalc->GetKappaIndex());
505
506
507 if (this->HasValue("io.confmatout"))
508 {
509     //Writing the confusion matrix in the output .CSV file
510
511     MapOfIndicesType::iterator itMapOfIndicesValid, itMapOfIndicesPred;
512     ClassLabelType labelValid = 0;
513
514     ConfusionMatrixType confusionMatrix = confMatCalc->GetConfusionMatrix();
515     MapOfIndicesType mapOfIndicesValid = confMatCalc->GetMapOfIndices();
516
517     unsigned int nbClassesPred = mapOfIndicesValid.size();
518
519     //////////////////////////////////////
520     //Filling the 2 headers for the output file
521     const std::string commentValidStr = "#Reference labels (rows):";
522     const std::string commentPredStr = "#Produced labels (columns):";
523     const char separatorChar = ',';
524     std::ostringstream ossHeaderValidLabels, ossHeaderPredLabels;
525
526     //Filling ossHeaderValidLabels and ossHeaderPredLabels for the output file
527     ossHeaderValidLabels << commentValidStr;
528     ossHeaderPredLabels << commentPredStr;
529
530     itMapOfIndicesValid = mapOfIndicesValid.begin();
531
532     while (itMapOfIndicesValid != mapOfIndicesValid.end())
533     {
534         //labels labelValid of mapOfIndicesValid are already sorted in otbConfusionMatrixCalculator
535         labelValid = itMapOfIndicesValid->second;
536         otbAppLogINFO("mapOfIndicesValid[" << itMapOfIndicesValid->first << "] = " << labelValid);

```

```

537
538 ossHeaderValidLabels << labelValid;
539 ossHeaderPredLabels << labelValid;
540
541 ++itMapOfIndicesValid;
542
543 if (itMapOfIndicesValid != mapOfIndicesValid.end())
544 {
545     ossHeaderValidLabels << separatorChar;
546     ossHeaderPredLabels << separatorChar;
547 }
548 else
549 {
550     ossHeaderValidLabels << std::endl;
551     ossHeaderPredLabels << std::endl;
552 }
553 }
554
555 std::ofstream outFile;
556 outFile.open(this->GetParameterString("io.confmatout").c_str());
557 outFile << std::fixed;
558 outFile.precision(10);
559
560 //////////////////////////////////////
561 //Writing the 2 headers
562 outFile << ossHeaderValidLabels.str();
563 outFile << ossHeaderPredLabels.str();
564 //////////////////////////////////////
565
566 unsigned int indexLabelValid = 0, indexLabelPred = 0;
567
568 for (itMapOfIndicesValid = mapOfIndicesValid.begin(); itMapOfIndicesValid !=
569 mapOfIndicesValid.end(); ++itMapOfIndicesValid)
570 {
571     indexLabelPred = 0;
572
573     for (itMapOfIndicesPred = mapOfIndicesValid.begin(); itMapOfIndicesPred !=
574 mapOfIndicesValid.end(); ++itMapOfIndicesPred)
575     {
576         //Writing the confusion matrix (sorted in otbConfusionMatrixCalculator) in the output file
577         outFile << confusionMatrix(indexLabelValid, indexLabelPred);
578         if (indexLabelPred < (nbClassesPred - 1))
579         {
580             outFile << separatorChar;
581         }
582         else
583         {
584             outFile << std::endl;
585             ++indexLabelPred;

```

```
586     }
587
588     ++indexLabelValid;
589 }
590
591 outFile.close();
592 } // END if (this->HasValue("io.confmatout"))
593
594 //TODO: implement hyperplane distance classifier and performance validation (cf. object detection) ?
595
596 }
597
598
599 }
600 }
601
602 OTB_APPLICATION_EXPORT(otb::Wrapper::TrainImagesClassifier)
```

Orfeo Toolbox – Train Lib SVM

```

1  /*=====
2  Program: ORFEO Toolbox
3  Language: C++
4  Date:   $Date$
5  Version: $Revision$
6
7  Copyright (c) Centre National d'Etudes Spatiales. All rights reserved.
8  See OTBCopyright.txt for details.
9
10 This software is distributed WITHOUT ANY WARRANTY; without even
11 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
12 PURPOSE. See the above copyright notices for more information.
13
14 =====*/
15
16 #include "otbTrainImagesClassifier.h"
17
18 namespace otb
19 {
20 namespace Wrapper
21 {
22 void TrainImagesClassifier::InitLibSVMParams()
23 {
24     AddChoice("classifier.libsvm", "LibSVM classifier");
25     SetParameterDescription("classifier.libsvm", "This group of parameters allows to set SVM classifier
26 parameters.");
27     AddParameter(ParameterType_Choice, "classifier.libsvm.k", "SVM Kernel Type");
28     AddChoice("classifier.libsvm.k.linear", "Linear");
29     AddChoice("classifier.libsvm.k.rbf", "Gaussian radial basis function");
30     AddChoice("classifier.libsvm.k.poly", "Polynomial");
31     AddChoice("classifier.libsvm.k.sigmoid", "Sigmoid");
32     SetParameterString("classifier.libsvm.k", "linear");
33     SetParameterDescription("classifier.libsvm.k", "SVM Kernel Type.");
34     AddParameter(ParameterType_Float, "classifier.libsvm.c", "Cost parameter C");
35     SetParameterFloat("classifier.libsvm.c", 1.0);
36     SetParameterDescription(
37         "classifier.libsvm.c",
38         "SVM models have a cost parameter C (1 by default) to control the trade-off between training
39 errors and forcing rigid margins.");
40     AddParameter(ParameterType_Empty, "classifier.libsvm.opt", "Parameters optimization");
41     MandatoryOff("classifier.libsvm.opt");
42     SetParameterDescription("classifier.libsvm.opt", "SVM parameters optimization flag.");
43 }
44
45 void TrainImagesClassifier::TrainLibSVM(ListSampleType::Pointer trainingListSample,
46 LabelListSampleType::Pointer trainingLabeledListSample)
47 {
48

```

```
49 LibSVMType::Pointer libSVMClassifier = LibSVMType::New();
50 libSVMClassifier->SetInputListSample(trainingListSample);
51 libSVMClassifier->SetTargetListSample(trainingLabeledListSample);
52 //SVM Option
53 //TODO : Add other options ?
54 if (IsParameterEnabled("classifier.libsvm.opt"))
55 {
56     libSVMClassifier->SetParameterOptimization(true);
57 }
58 libSVMClassifier->SetC(GetParameterFloat("classifier.libsvm.c"));
59
60 switch (GetParameterInt("classifier.libsvm.k"))
61 {
62     case 0: // LINEAR
63         libSVMClassifier->SetKernelType(LINEAR);
64         break;
65     case 1: // RBF
66         libSVMClassifier->SetKernelType(RBF);
67         break;
68     case 2: // POLY
69         libSVMClassifier->SetKernelType(POLY);
70         break;
71     case 3: // SIGMOID
72         libSVMClassifier->SetKernelType(SIGMOID);
73         break;
74     default: // DEFAULT = LINEAR
75         libSVMClassifier->SetKernelType(LINEAR);
76         break;
77 }
78 libSVMClassifier->Train();
79 libSVMClassifier->Save(GetParameterString("io.out"));
80 }
81
82 } //end namespace wrapper
83 } //end namespace otb
```

Orfeo Toolbox – Train Decision Tree

```

1  /*=====
2  Program: ORFEO Toolbox
3  Language: C++
4  Date:   $Date$
5  Version: $Revision$
6
7
8  Copyright (c) Centre National d'Etudes Spatiales. All rights reserved.
9  See OTBCopyright.txt for details.
10
11
12  This software is distributed WITHOUT ANY WARRANTY; without even
13  the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
14  PURPOSE. See the above copyright notices for more information.
15
16  =====*/
17
18  #include "otbTrainImagesClassifier.h"
19
20  namespace otb
21  {
22  namespace Wrapper
23  {
24  void TrainImagesClassifier::InitDecisionTreeParams()
25  {
26      AddChoice("classifier.dt", "Decision Tree classifier");
27      SetParameterDescription("classifier.dt",
28          "This group of parameters allows to set Decision Tree classifier parameters. "
29          "See complete documentation here
30          \\url{http://docs.opencv.org/modules/ml/doc/decision_trees.html}.");
31      //MaxDepth
32      AddParameter(ParameterType_Int, "classifier.dt.max", "Maximum depth of the tree");
33      SetParameterInt("classifier.dt.max", 65535);
34      SetParameterDescription(
35          "classifier.dt.max", "The training algorithm attempts to split each node while its depth is smaller
36          than the maximum "
37          "possible depth of the tree. The actual depth may be smaller if the other termination criteria are
38          met, and/or "
39          "if the tree is pruned.");
40
41      //MinSampleCount
42      AddParameter(ParameterType_Int, "classifier.dt.min", "Minimum number of samples in each
43          node");
44      SetParameterInt("classifier.dt.min", 10);
45      SetParameterDescription("classifier.dt.min", "If the number of samples in a node is smaller than this
46          parameter, "
47          "then this node will not be split.");
48

```

```

49 //RegressionAccuracy
50 AddParameter(ParameterType_Float, "classifier.dt.ra", "Termination criteria for regression tree");
51 SetParameterFloat("classifier.dt.ra", 0.01);
52 SetParameterDescription("classifier.dt.min", "If all absolute differences between an estimated value
53 in a node "
54         "and the values of the train samples in this node are smaller than this regression
55 accuracy parameter, "
56         "then the node will not be split.");
57
58 //UseSurrogates : don't need to be exposed !
59 //AddParameter(ParameterType_Empty, "classifier.dt.sur", "Surrogate splits will be built");
60 //SetParameterDescription("classifier.dt.sur", "These splits allow to work with missing data and
61 compute variable importance correctly.");
62
63 //MaxCategories
64 AddParameter(ParameterType_Int, "classifier.dt.cat",
65         "Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal
66 split");
67 SetParameterInt("classifier.dt.cat", 10);
68 SetParameterDescription(
69     "classifier.dt.cat",
70     "Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal split.");
71
72 //CVFolds
73 AddParameter(ParameterType_Int, "classifier.dt.f", "K-fold cross-validations");
74 SetParameterInt("classifier.dt.f", 10);
75 SetParameterDescription(
76     "classifier.dt.f", "If cv_folds > 1, then it prunes a tree with K-fold cross-validation where K is equal
77 to cv_folds.");
78
79 //Use1seRule
80 AddParameter(ParameterType_Empty, "classifier.dt.r", "Set Use1seRule flag to false");
81 SetParameterDescription(
82     "classifier.dt.r",
83     "If true, then a pruning will be harsher. This will make a tree more compact and more resistant to
84 the training data noise but a bit less accurate.");
85
86 //TruncatePrunedTree
87 AddParameter(ParameterType_Empty, "classifier.dt.t", "Set TruncatePrunedTree flag to false");
88 SetParameterDescription("classifier.dt.t", "If true, then pruned branches are physically removed
89 from the tree.");
90
91 //Priors are not exposed.
92
93 }
94
95 void TrainImagesClassifier::TrainDecisionTree(ListSampleType::Pointer trainingListSample,
96                                             LabelListSampleType::Pointer trainingLabeledListSample)
97 {

```



```
98   DecisionTreeType::Pointer classifier = DecisionTreeType::New();
99   classifier->SetInputListSample(trainingListSample);
100  classifier->SetTargetListSample(trainingLabeledListSample);
101  classifier->SetMaxDepth(GetParameterInt("classifier.dt.max"));
102  classifier->SetMinSampleCount(GetParameterInt("classifier.dt.min"));
103  classifier->SetRegressionAccuracy(GetParameterFloat("classifier.dt.ra"));
104  classifier->SetMaxCategories(GetParameterInt("classifier.dt.cat"));
105  classifier->SetCVFolds(GetParameterInt("classifier.dt.f"));
106  if (IsParameterEnabled("classifier.dt.r"))
107  {
108      classifier->SetUse1seRule(false);
109  }
110  if (IsParameterEnabled("classifier.dt.t"))
111  {
112      classifier->SetTruncatePrunedTree(false);
113  }
114  classifier->Train();
115  classifier->Save(GetParameterString("io.out"));
116  }
117
118 } //end namespace wrapper
119 } //end namespace otb
120
121
```

Orfeo Toolbox – Train KNN

```

1  /*=====
2  Program: ORFEO Toolbox
3  Language: C++
4  Date:   $Date$
5  Version: $Revision$
6
7  Copyright (c) Centre National d'Etudes Spatiales. All rights reserved.
8  See OTBCopyright.txt for details.
9  This software is distributed WITHOUT ANY WARRANTY; without even
10 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
11 PURPOSE. See the above copyright notices for more information.
12
13 =====*/
14
15 #include "otbTrainImagesClassifier.h"
16
17 namespace otb
18 {
19     namespace Wrapper
20     {
21         void TrainImagesClassifier::InitKNNParams()
22         {
23             AddChoice("classifier.knn", "KNN classifier");
24             SetParameterDescription("classifier.knn", "This group of parameters allows to set KNN classifier
25 parameters. "
26             "See complete documentation here
27             \\url{http://docs.opencv.org/modules/ml/doc/k_nearest_neighbors.html}.");
28
29             //K parameter
30             AddParameter(ParameterType_Int, "classifier.knn.k", "Number of Neighbors");
31             SetParameterInt("classifier.knn.k", 32);
32             SetParameterDescription("classifier.knn.k", "The number of neighbors to use.");
33
34         }
35
36         void TrainImagesClassifier::TrainKNN(ListSampleType::Pointer trainingListSample,
37         LabelListSampleType::Pointer trainingLabeledListSample)
38         {
39             KNNTType::Pointer knnClassifier = KNNTType::New();
40             knnClassifier->SetInputListSample(trainingListSample);
41             knnClassifier->SetTargetListSample(trainingLabeledListSample);
42             knnClassifier->SetK(GetParameterInt("classifier.knn.k"));
43
44             knnClassifier->Train();
45             knnClassifier->Save(GetParameterString("io.out"));
46         }
47     } //end namespace wrapper
48 } //end namespace otb

```

Orfeo Toolbox – Bundle To Perfect Sensor

```

1  /*=====
2
3  Program: ORFEO Toolbox
4  Language: C++
5  Date:   $Date$
6  Version: $Revision$
7
8
9  Copyright (c) Centre National d'Etudes Spatiales. All rights reserved.
10 See OTBCopyright.txt for details.
11
12
13 This software is distributed WITHOUT ANY WARRANTY; without even
14 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
15 PURPOSE. See the above copyright notices for more information.
16
17 =====*/
18 #include "otbWrapperApplication.h"
19 #include "otbWrapperApplicationFactory.h"
20
21 #include "itkVectorIndexSelectionCastImageFilter.h"
22 #include "otbGenericRSResampleImageFilter.h"
23 #include "otbBCOInterpolateImageFunction.h"
24 #include "otbSimpleRcsPanSharpeningFusionImageFilter.h"
25 #include "itkFixedArray.h"
26
27 // Elevation handler
28 #include "otbWrapperElevationParametersHandler.h"
29
30 namespace otb
31 {
32 namespace Wrapper
33 {
34
35
36 class BundleToPerfectSensor : public Application
37 {
38 public:
39     /** Standard class typedefs. */
40     typedef BundleToPerfectSensor    Self;
41     typedef Application              Superclass;
42     typedef itk::SmartPointer<Self>  Pointer;
43     typedef itk::SmartPointer<const Self> ConstPointer;
44
45     /** Standard macro */
46     itkNewMacro(Self);
47
48     itkTypeMacro(BundleToPerfectSensor, otb::Application);

```

```

49 private:
50
51 void DoInit()
52 {
53     SetName("BundleToPerfectSensor");
54     SetDescription("Perform P+XS pansharpning");
55
56     // Documentation
57     SetDocName("Bundle to perfect sensor");
58     SetDocLongDescription("This application performs P+XS pansharpning.");
59     SetDocLimitations("None");
60     SetDocAuthors("OTB-Team");
61     SetDocSeeAlso(" ");
62
63     AddDocTag(Tags::Geometry);
64     AddDocTag(Tags::Pansharpning);
65
66     AddParameter(ParameterType_InputImage, "inp", "Input PAN Image");
67     SetParameterDescription("inp"," Input panchromatic image.");
68     AddParameter(ParameterType_InputImage, "inxs", "Input XS Image");
69     SetParameterDescription("inxs"," Input XS image.");
70
71     // Elevation
72     ElevationParametersHandler::AddElevationParameters(this, "elev");
73
74     AddParameter(ParameterType_Float, "lms", "Spacing of the deformation field");
75     SetParameterDescription("lms"," Spacing of the deformation field. Default is 10 times the PAN
76 image spacing.");
77     AddParameter(ParameterType_OutputImage, "out", "Output image");
78     SetParameterDescription("out"," Output image.");
79     AddRAMParameter();
80
81     MandatoryOff("lms");
82
83     // Doc example parameter settings
84     SetDocExampleParameterValue("inp", "QB_Toulouse_Ortho_PAN.tif");
85     SetDocExampleParameterValue("inxs", "QB_Toulouse_Ortho_XS.tif");
86     SetDocExampleParameterValue("out", "BundleToPerfectSensor.png uchar");
87
88 }
89
90 void DoUpdateParameters()
91 {
92     // Nothing to do here : all parameters are independent
93 }
94
95 void DoExecute()
96 {
97     FloatVectorImageType* panchroV = GetParameterImage("inp");

```

```

98   if ( panchroV->GetNumberOfComponentsPerPixel() != 1 )
99   {
100      itkExceptionMacro(<< "The panchromatic image must be a single channel image")
101   }
102
103   // Transform the PAN image to otb::Image
104   typedef otb::Image<FloatVectorImageType::InternalPixelType> FloatImageType;
105   typedef itk::VectorIndexSelectionCastImageFilter<FloatVectorImageType, FloatImageType>
106   VectorIndexSelectionCastImageFilterType;
107
108   VectorIndexSelectionCastImageFilterType::Pointer channelSelect =
109   VectorIndexSelectionCastImageFilterType::New();
110   m_Ref.push_back(channelSelect.GetPointer());
111   channelSelect->SetIndex(0);
112   channelSelect->SetInput(panchroV);
113   channelSelect->UpdateOutputInformation();
114   FloatImageType::Pointer panchro = channelSelect->GetOutput();
115
116
117   FloatVectorImageType* xs = GetParameterImage("inxs");
118
119   typedef otb::BCOInterpolateImageFunction<FloatVectorImageType> InterpolatorType;
120   typedef otb::GenericRSResampleImageFilter<FloatVectorImageType, FloatVectorImageType>
121   ResamplerType;
122   typedef otb::SimpleRcsPanSharpeningFusionImageFilter<FloatImageType, FloatVectorImageType,
123   FloatVectorImageType> FusionFilterType;
124
125   // Resample filter
126   ResamplerType::Pointer resampler = ResamplerType::New();
127   m_Ref.push_back(resampler.GetPointer());
128
129   InterpolatorType::Pointer interpolator = InterpolatorType::New();
130   resampler->SetInterpolator(interpolator);
131
132   // Setup the DEM Handler
133
134   otb::Wrapper::ElevationParametersHandler::SetupDEMHandlerFromElevationParameters(this,"elev");
135
136   // Set up output image informations
137   FloatVectorImageType::SpacingType spacing = panchro->GetSpacing();
138   FloatVectorImageType::IndexType start = panchro->GetLargestPossibleRegion().GetIndex();
139   FloatVectorImageType::SizeType size = panchro->GetLargestPossibleRegion().GetSize();
140   FloatVectorImageType::PointType origin = panchro->GetOrigin();
141
142   if(IsParameterEnabled("lms") && HasValue("lms"))
143   {
144      double defScalarSpacing = GetParameterFloat("lms");
145      otbAppLogINFO(<< "Generating coarse deformation field (spacing=" << defScalarSpacing << ")" <<
146      std::endl);

```

```

147     FloatVectorImageType::SpacingType defSpacing;
148
149     defSpacing[0] = defScalarSpacing;
150     defSpacing[1] = defScalarSpacing;
151
152     resampler->SetDisplacementFieldSpacing(defSpacing);
153 }
154 else
155 {
156     FloatVectorImageType::SpacingType defSpacing;
157     defSpacing[0]=10*spacing[0];
158     defSpacing[1]=10*spacing[1];
159     resampler->SetDisplacementFieldSpacing(defSpacing);
160 }
161
162     FloatVectorImageType::PixelType defaultValue;
163     itk::NumericTraits<FloatVectorImageType::PixelType>::SetLength(defaultValue, xs-
164 >GetNumberOfComponentsPerPixel());
165
166     resampler->SetInput(xs);
167     resampler->SetOutputOrigin(origin);
168     resampler->SetOutputSpacing(spacing);
169     resampler->SetOutputSize(size);
170     resampler->SetOutputStartIndex(start);
171     resampler->SetOutputKeywordList(panchro->GetImageKeywordlist());
172     resampler->SetOutputProjectionRef(panchro->GetProjectionRef());
173     resampler->SetEdgePaddingValue(defaultValue);
174
175     resampler->UpdateOutputInformation();
176     FloatVectorImageType::Pointer xsResample = resampler->GetOutput();
177
178     FusionFilterType::Pointer fusionFilter = FusionFilterType::New();
179     m_Ref.push_back(fusionFilter.GetPointer());
180
181     fusionFilter->SetPanInput(panchro);
182     fusionFilter->SetXsInput(resampler->GetOutput());
183     fusionFilter->UpdateOutputInformation();
184
185     SetParameterOutputImage("out", fusionFilter->GetOutput());
186 }
187     std::vector<itk::ProcessObject::Pointer> m_Ref;
188
189 };
190 }
191 }
192
193 OTB_APPLICATION_EXPORT(otb::Wrapper::BundleToPerfectSensor)

```

Orfeo Toolbox – Radiometric Indices

```

1  *=====
2
3  Program: ORFEO Toolbox
4  Language: C++
5  Date:   $Date$
6  Version: $Revision$
7
8
9  Copyright (c) Centre National d'Etudes Spatiales. All rights reserved.
10 See OTBCopyright.txt for details.
11
12
13 This software is distributed WITHOUT ANY WARRANTY; without even
14 the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
15 PURPOSE. See the above copyright notices for more information.
16
17 =====*/
18 #include "otbWrapperApplication.h"
19 #include "otbWrapperApplicationFactory.h"
20
21 #include "otbMultiChannelIRAndNIRIndexImageFilter.h"
22 #include "otbMultiChannelGAndRIndexImageFilter.h"
23 #include "otbMultiChannelIRAndGAndNIRIndexImageFilter.h"
24 #include "otbMultiChannelIRAndBAndNIRIndexImageFilter.h"
25 #include "otbMultiChannelIRAndGAndNIRIndexImageFilter.h"
26 #include "otbWaterIndicesFunctor.h"
27 #include "otbSoilIndicesFunctor.h"
28 #include "otbBuiltUpIndicesFunctor.h"
29
30 #include "otbImageList.h"
31 #include "otbImageListToVectorImageFilter.h"
32
33 #include "otbWrapperNumericalParameter.h"
34
35 namespace otb
36 {
37     namespace Wrapper
38     {
39
40         class RadiometricIndices: public Application
41         {
42         public:
43             /** Standard class typedefs. */
44             typedef RadiometricIndices    Self;
45             typedef Application            Superclass;
46             typedef itk::SmartPointer<Self>    Pointer;
47             typedef itk::SmartPointer<const Self> ConstPointer;
48

```



```

49  /** Standard macro */
50  itkNewMacro(Self);
51
52  itkTypeMacro(RadiometricIndices, otb::Wrapper::Application);
53
54  /** Output containers typedef */
55  typedef ObjectList<itk::ProcessObject>          FilterListType;
56  typedef ImageList<FloatImageType>              ImageListType;
57  typedef ImageListToVectorImageFilter<ImageListType, FloatVectorImageType>
58  ImageListToVectorImageFilterType;
59
60  /** Radiometric water indices functors typedef */
61  typedef Functor::SRWI<FloatVectorImageType::InternalPixelType,
62  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> SRWIFunctorType;
63  typedef Functor::NDWI<FloatVectorImageType::InternalPixelType,
64  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> NDWIFunctorType;
65  typedef Functor::NDWI2<FloatVectorImageType::InternalPixelType,
66  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> NDWI2FunctorType;
67  typedef Functor::MNDWI<FloatVectorImageType::InternalPixelType,
68  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> MNDWIFunctorType;
69  typedef Functor::NDPI<FloatVectorImageType::InternalPixelType,
70  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> NDPIFunctorType;
71  typedef Functor::NDTI<FloatVectorImageType::InternalPixelType,
72  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> NDTIFunctorType;
73  //typedef Functor::WaterSqrtSpectralAngleFunctor<FloatVectorImageType::InternalPixelType,
74  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType>
75  WaterSqrtSpectralAngleFunctor;
76
77  /** Radiometric vegetation indices functors typedef */
78  typedef Functor::NDVI<FloatVectorImageType::InternalPixelType,
79  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> NDVIFunctor;
80  typedef Functor::TNDVI<FloatVectorImageType::InternalPixelType,
81  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> TNDVIFunctor;
82  typedef Functor::RVI<FloatVectorImageType::InternalPixelType,
83  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> RVIFunctor;
84  typedef Functor::SAVI<FloatVectorImageType::InternalPixelType,
85  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> SAVIFunctor;
86  typedef Functor::TSAVI<FloatVectorImageType::InternalPixelType,
87  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> TSAVIFunctor;
88  typedef Functor::MSAVI<FloatVectorImageType::InternalPixelType,
89  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> MSAVIFunctor;
90  typedef Functor::MSAVI2<FloatVectorImageType::InternalPixelType,
91  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> MSAVI2Functor;
92  typedef Functor::GEMI<FloatVectorImageType::InternalPixelType,
93  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> GEMIFunctor;
94  typedef Functor::IPVI<FloatVectorImageType::InternalPixelType,
95  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> IPVIFunctor;
96  typedef Functor::LAIFromNDVILogarithmic<FloatVectorImageType::InternalPixelType,
97  FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> LAIFromNDVILogFunctor;

```

```

98     typedef Functor::LAIFromReflectancesLinear<FloatVectorImageType::InternalPixelType,
99     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> LAIFromReflLinearFunctor;
100     typedef Functor::LAIFromNDVIFormosat2Functor<FloatVectorImageType::InternalPixelType,
101     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> LAIFromNDVIFormoFunctor;
102
103     /** Radiometric soil indices functors typedef */
104     typedef Functor::IR<FloatVectorImageType::InternalPixelType,
105     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> IRFunctor;
106     typedef Functor::IC<FloatVectorImageType::InternalPixelType,
107     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> ICFunctor;
108     typedef Functor::IB<FloatVectorImageType::InternalPixelType,
109     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> IBFunctor;
110     typedef Functor::IB2<FloatVectorImageType::InternalPixelType,
111     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType, FloatImageType::PixelType>
112     IB2Functor;
113
114     /** Radiometric built up indices functors typedef */
115     typedef Functor::NDBI<FloatVectorImageType::InternalPixelType,
116     FloatVectorImageType::InternalPixelType, FloatImageType::PixelType> NDBIFunctor;
117
118     /** Radiometric indices filters typedef */
119     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, NDWIFunctorType>
120     NDWIFilterType;
121     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, NDWI2FunctorType>
122     NDWI2FilterType;
123     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType,
124     MNDWIFunctorType> MNDWIFilterType;
125     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, NDPIFunctorType>
126     NDPIFilterType;
127     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, NDTIFunctorType>
128     NDTIFilterType;
129     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, SRWIFunctorType>
130     SRWIFilterType;
131
132     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
133     NDVIFunctor> NDVIFilterType;
134     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
135     TNDVIFunctor> TNDVIFilterType;
136     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
137     RVIFunctor> RVIFilterType;
138     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
139     SAVIFunctor> SAVIFilterType;
140     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
141     TSAVIFunctor> TSAVIFilterType;
142     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
143     MSAVIFunctor> MSAVIFilterType;
144     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
145     MSAVI2Functor> MSAVI2FilterType;

```

```

146     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
147     GEMIFunctor>      GEMIFilterType;
148     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
149     IPVIFunctor>      IPVIFilterType;
150     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
151     LAIFromNDVILogFunctor> LAIFromNDVILogFilterType;
152     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
153     LAIFromReflLinearFunctor> LAIFromReflLinearFilterType;
154     typedef MultiChannelRAndNIRIndexImageFilter<FloatVectorImageType, FloatImageType,
155     LAIFromNDVIFormoFunctor> LAIFromNDVIFormoFilterType;
156
157     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, IRFunctor>
158     RIFilterType;
159     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, ICFunctor>
160     CIFilterType;
161     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, IBFunctor>
162     BIFilterType;
163     typedef itk::UnaryFunctorImageFilter<FloatVectorImageType, FloatImageType, IB2Functor>
164     BI2FilterType;
165
166     struct indiceSpec
167     {
168         std::string key;
169         std::string item;
170         std::string description;
171         std::string type;
172         std::string chan1;
173         std::string chan2;
174         std::string chan3;
175     };
176
177
178     private:
179     void DoInit()
180     {
181         SetName("RadiometricIndices");
182         SetDescription("Compute radiometric indices.");
183
184         // Documentation
185         SetDocName("Radiometric Indices");
186         SetDocLongDescription("This application computes radiometric indices using the relevant channels
187 of the input image. The output is a multi band image into which each channel is one of the selected
188 indices.");
189         SetDocLimitations("None");
190         SetDocAuthors("OTB-Team");
191         SetDocSeeAlso("otbVegetationIndicesFunctor, otbWaterIndicesFunctor and otbSoilIndicesFunctor
192 classes");
193
194         AddDocTag("Radiometric Indices");

```

```

195 AddDocTag(Tags::FeatureExtraction);
196
197 AddParameter(ParameterType_InputImage, "in", "Input Image");
198 SetParameterDescription("in", "Input image");
199
200 AddParameter(ParameterType_OutputImage, "out", "Output Image");
201 SetParameterDescription("out", "Radiometric indices output image");
202
203 AddRAMParameter();
204
205 AddParameter(ParameterType_Group, "channels", "Channels selection");
206 SetParameterDescription("channels", "Channels selection");
207
208 AddParameter(ParameterType_Int, "channels.blue", "Blue Channel");
209 SetParameterDescription("channels.blue", "Blue channel index");
210 SetDefaultParameterInt("channels.blue", 1);
211 AddParameter(ParameterType_Int, "channels.green", "Green Channel");
212 SetParameterDescription("channels.green", "Green channel index");
213 SetDefaultParameterInt("channels.green", 1);
214 AddParameter(ParameterType_Int, "channels.red", "Red Channel");
215 SetParameterDescription("channels.red", "Red channel index");
216 SetDefaultParameterInt("channels.red", 1);
217 AddParameter(ParameterType_Int, "channels.nir", "NIR Channel");
218 SetParameterDescription("channels.nir", "NIR channel index");
219 SetDefaultParameterInt("channels.nir", 1);
220 AddParameter(ParameterType_Int, "channels.mir", "Mir Channel");
221 SetParameterDescription("channels.mir", "Mir channel index");
222 SetDefaultParameterInt("channels.mir", 1);
223 //AddParameter(ParameterType_Int, "channels.rho860", "Rho860 Channel");
224 //SetParameterDescription("channels.rho860", "860nm band channel index");
225 //SetDefaultParameterInt("channels.rho860", 1);
226 //AddParameter(ParameterType_Int, "channels.rho1240", "Rho1240 Channel");
227 //SetParameterDescription("channels.rho1240", "1240nm band channel index");
228 //SetDefaultParameterInt("channels.rho1240", 1);
229
230 AddParameter(ParameterType_ListView, "list", "Available Radiometric Indices");
231 SetParameterDescription("list", "List of available radiometric indices with their relevant channels in
232 brackets:\n\
233 Vegetation:NDVI - Normalized difference vegetation index (Red, NIR)\n\
234 Vegetation:TNDVI - Transformed normalized difference vegetation index (Red, NIR)\n\
235 Vegetation:RVI - Ratio vegetation index (Red, NIR)\n\
236 Vegetation:SAVI - Soil adjusted vegetation index (Red, NIR)\n\
237 Vegetation:TSAVI - Transformed soil adjusted vegetation index (Red, NIR)\n\
238 Vegetation:MSAVI - Modified soil adjusted vegetation index (Red, NIR)\n\
239 Vegetation:MSAVI2 - Modified soil adjusted vegetation index 2 (Red, NIR)\n\
240 Vegetation:GEMI - Global environment monitoring index (Red, NIR)\n\
241 Vegetation:IPVI - Infrared percentage vegetation index (Red, NIR)\n\
242 \n\
243 Water:NDWI - Normalized difference water index (Gao 1996) (NIR, MIR)\n\

```

```

244 Water:NDWI2 - Normalized difference water index (Mc Feeters 1996) (Green, NIR)\n\
245 Water:MNDWI - Modified normalized difference water index (Xu 2006) (Green, MIR)\n\
246 Water:NDPI - Normalized difference pond index (Lacaux et al.) (MIR, Green)\n\
247 Water:NDTI - Normalized difference turbidity index (Lacaux et al.) (Red, Green)\n\
248 \n\
249 Soil:RI - Redness index (Red, Green)\n\
250 Soil:CI - Color index (Red, Green)\n\
251 Soil:BI - Brightness index (Red, Green)\n\
252 Soil:BI2 - Brightness index 2 (NIR, Red, Green)");
253
254 // Doc example parameter settings
255 SetDocExampleParameterValue("in", "qb_RoadExtract.tif");
256 SetDocExampleParameterValue("list", "Vegetation:NDVI Vegetation:RVI Vegetation:IPVI");
257 SetDocExampleParameterValue("out", "RadiometricIndicesImage.tif");
258
259 m_Map.clear();
260
261 // Add Available choices
262 indiceSpec s_NDVI;
263 s_NDVI.key = "list.ndvi";
264 s_NDVI.item = "Vegetation:NDVI";
265 s_NDVI.description = "";
266 s_NDVI.type = "NDVI";
267 s_NDVI.chan1 = "red";
268 s_NDVI.chan2 = "nir";
269 s_NDVI.chan3 = "";
270 m_Map.push_back(s_NDVI);
271
272 indiceSpec s_TNDVI;
273 s_TNDVI.key = "list.tndvi";
274 s_TNDVI.item = "Vegetation:TNDVI";
275 s_TNDVI.description = "";
276 s_TNDVI.type = "TNDVI";
277 s_TNDVI.chan1 = "red";
278 s_TNDVI.chan2 = "nir";
279 s_TNDVI.chan3 = "";
280 m_Map.push_back(s_TNDVI);
281
282 indiceSpec s_RVI;
283 s_RVI.key = "list.rvi";
284 s_RVI.item = "Vegetation:RVI";
285 s_RVI.description = "";
286 s_RVI.type = "RVI";
287 s_RVI.chan1 = "red";
288 s_RVI.chan2 = "nir";
289 s_RVI.chan3 = "";
290 m_Map.push_back(s_RVI);
291
292 indiceSpec s_SAVI;

```

```

293  s_SAVI.key = "list.savi";
294  s_SAVI.item = "Vegetation:SAVI";
295  s_SAVI.description = "";
296  s_SAVI.type = "SAVI";
297  s_SAVI.chan1 = "red";
298  s_SAVI.chan2 = "nir";
299  s_SAVI.chan3 = "";
300  m_Map.push_back(s_SAVI);
301
302  indiceSpec s_TSAVI;
303  s_TSAVI.key = "list.tsavi";
304  s_TSAVI.item = "Vegetation:TSAVI";
305  s_TSAVI.description = "";
306  s_TSAVI.type = "TSAVI";
307  s_TSAVI.chan1 = "red";
308  s_TSAVI.chan2 = "nir";
309  s_TSAVI.chan3 = "";
310  m_Map.push_back(s_TSAVI);
311
312  indiceSpec s_MSAVI;
313  s_MSAVI.key = "list.msavi";
314  s_MSAVI.item = "Vegetation:MSAVI";
315  s_MSAVI.description = "";
316  s_MSAVI.type = "MSAVI";
317  s_MSAVI.chan1 = "red";
318  s_MSAVI.chan2 = "nir";
319  s_MSAVI.chan3 = "";
320  m_Map.push_back(s_MSAVI);
321
322  indiceSpec s_MSAVI2;
323  s_MSAVI2.key = "list.msavi2";
324  s_MSAVI2.item = "Vegetation:MSAVI2";
325  s_MSAVI2.description = "";
326  s_MSAVI2.type = "MSAVI2";
327  s_MSAVI2.chan1 = "red";
328  s_MSAVI2.chan2 = "nir";
329  s_MSAVI2.chan3 = "";
330  m_Map.push_back(s_MSAVI2);
331
332  indiceSpec s_GEMI;
333  s_GEMI.key = "list.gemi";
334  s_GEMI.item = "Vegetation:GEMI";
335  s_GEMI.description = "";
336  s_GEMI.type = "GEMI";
337  s_GEMI.chan1 = "red";
338  s_GEMI.chan2 = "nir";
339  s_GEMI.chan3 = "";
340  m_Map.push_back(s_GEMI);
341

```

```

342 indiceSpec s_IPVI;
343 s_IPVI.key = "list.ipvi";
344 s_IPVI.item = "Vegetation:IPVI";
345 s_IPVI.description = "";
346 s_IPVI.type = "IPVI";
347 s_IPVI.chan1 = "red";
348 s_IPVI.chan2 = "nir";
349 s_IPVI.chan3 = "";
350 m_Map.push_back(s_IPVI);
351
352 indiceSpec s_LAIFromNDVILog;
353 s_LAIFromNDVILog.key = "list.laindvilog";
354 s_LAIFromNDVILog.item = "Vegetation:LAIFromNDVILog";
355 s_LAIFromNDVILog.description = "";
356 s_LAIFromNDVILog.type = "LAIFromNDVILog";
357 s_LAIFromNDVILog.chan1 = "red";
358 s_LAIFromNDVILog.chan2 = "nir";
359 s_LAIFromNDVILog.chan3 = "";
360 m_Map.push_back(s_LAIFromNDVILog);
361
362 indiceSpec s_LAIFromReflLinear;
363 s_LAIFromReflLinear.key = "list.lairefl";
364 s_LAIFromReflLinear.item = "Vegetation:LAIFromReflLinear";
365 s_LAIFromReflLinear.description = "";
366 s_LAIFromReflLinear.type = "LAIFromReflLinear";
367 s_LAIFromReflLinear.chan1 = "red";
368 s_LAIFromReflLinear.chan2 = "nir";
369 s_LAIFromReflLinear.chan3 = "";
370 m_Map.push_back(s_LAIFromReflLinear);
371
372 indiceSpec s_LAIFromNDVIFORMO;
373 s_LAIFromNDVIFORMO.key = "list.laindviformo";
374 s_LAIFromNDVIFORMO.item = "Vegetation:LAIFromNDVIFORMO";
375 s_LAIFromNDVIFORMO.description = "";
376 s_LAIFromNDVIFORMO.type = "LAIFromNDVIFORMO";
377 s_LAIFromNDVIFORMO.chan1 = "red";
378 s_LAIFromNDVIFORMO.chan2 = "nir";
379 s_LAIFromNDVIFORMO.chan3 = "";
380 m_Map.push_back(s_LAIFromNDVIFORMO);
381
382 indiceSpec s_NDWI;
383 s_NDWI.key = "list.ndwi";
384 s_NDWI.item = "Water:NDWI";
385 s_NDWI.description = "";
386 s_NDWI.type = "NDWI";
387 s_NDWI.chan1 = "nir";
388 s_NDWI.chan2 = "mir";
389 s_NDWI.chan3 = "";
390 m_Map.push_back(s_NDWI);

```

```
391
392     indiceSpec s_NDWI2;
393     s_NDWI2.key = "list.ndwi2";
394     s_NDWI2.item = "Water:NDWI2";
395     s_NDWI2.description = "";
396     s_NDWI2.type = "NDWI2";
397     s_NDWI2.chan1 = "green";
398     s_NDWI2.chan2 = "nir";
399     s_NDWI2.chan3 = "";
400     m_Map.push_back(s_NDWI2);
401
402     indiceSpec s_MNDWI;
403     s_MNDWI.key = "list.mndwi";
404     s_MNDWI.item = "Water:MNDWI";
405     s_MNDWI.description = "";
406     s_MNDWI.type = "MNDWI";
407     s_MNDWI.chan1 = "green";
408     s_MNDWI.chan2 = "mir";
409     s_MNDWI.chan3 = "";
410     m_Map.push_back(s_MNDWI);
411
412     indiceSpec s_NDPI;
413     s_NDPI.key = "list.ndpi";
414     s_NDPI.item = "Water:NDPI";
415     s_NDPI.description = "";
416     s_NDPI.type = "NDPI";
417     s_NDPI.chan1 = "mir";
418     s_NDPI.chan2 = "green";
419     s_NDPI.chan3 = "";
420     m_Map.push_back(s_NDPI);
421
422     indiceSpec s_NDTI;
423     s_NDTI.key = "list.ndti";
424     s_NDTI.item = "Water:NDTI";
425     s_NDTI.description = "";
426     s_NDTI.type = "NDTI";
427     s_NDTI.chan1 = "red";
428     s_NDTI.chan2 = "green";
429     s_NDTI.chan3 = "";
430     m_Map.push_back(s_NDTI);
431
432     indiceSpec s_SRWI;
433     s_SRWI.key = "list.srwi";
434     s_SRWI.item = "Water:SRWI";
435     s_SRWI.description = "";
436     s_SRWI.type = "SRWI";
437     s_SRWI.chan1 = "rho860";
438     s_SRWI.chan2 = "rho1240";
439     s_SRWI.chan3 = "";
```



```
440 //m_Map.push_back(s_SRWI);
441
442 indiceSpec s_RI;
443 s_RI.key = "list.ri";
444 s_RI.item = "Soil:RI";
445 s_RI.description = "";
446 s_RI.type = "RI";
447 s_RI.chan1 = "red";
448 s_RI.chan2 = "green";
449 s_RI.chan3 = "";
450 m_Map.push_back(s_RI);
451
452 indiceSpec s_CI;
453 s_CI.key = "list.ci";
454 s_CI.item = "Soil:CI";
455 s_CI.description = "";
456 s_CI.type = "CI";
457 s_CI.chan1 = "red";
458 s_CI.chan2 = "green";
459 s_CI.chan3 = "";
460 m_Map.push_back(s_CI);
461
462 indiceSpec s_BI;
463 s_BI.key = "list.bi";
464 s_BI.item = "Soil:BI";
465 s_BI.description = "";
466 s_BI.type = "BI";
467 s_BI.chan1 = "red";
468 s_BI.chan2 = "green";
469 s_BI.chan3 = "";
470 m_Map.push_back(s_BI);
471
472 indiceSpec s_BI2;
473 s_BI2.key = "list.bi2";
474 s_BI2.item = "Soil:BI2";
475 s_BI2.description = "";
476 s_BI2.type = "BI2";
477 s_BI2.chan1 = "nir";
478 s_BI2.chan2 = "red";
479 s_BI2.chan3 = "green";
480 m_Map.push_back(s_BI2);
481
482 ClearChoices("list");
483 for ( unsigned int i=0; i<m_Map.size(); i++ )
484 {
485     AddChoice(m_Map[i].key, m_Map[i].item);
486     //SetParameterDescription(m_Map[i].item, m_Map[i].description);
487 }
488 }
```

```

489 void DoUpdateParameters()
490 {
491     //Nothing to do here
492 }
493
494 #define otbRadiometricWaterIndicesMacro( type ) \
495 { \
496     type##FilterType::Pointer I_##type##Filter = type##FilterType::New(); \
497     std::ostringstream oss; \
498     oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan1; \
499     I_##type##Filter->GetFunctor().SetIndex1(this->GetParameterInt(oss.str()));\
500     oss.str(""); \
501     oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan2; \
502     I_##type##Filter->GetFunctor().SetIndex2(this->GetParameterInt(oss.str()));\
503     I_##type##Filter->SetInput(inImage); \
504     m_FilterList->PushBack( I_##type##Filter ); \
505     m_ImageList->PushBack( I_##type##Filter->GetOutput() ); \
506     otbAppLogINFO(<< m_Map[GetSelectedItems("list")[idx]].item << " added.");\
507 }
508
509 #define otbRadiometricVegetationIndicesMacro( type ) \
510 { \
511     type##FilterType::Pointer I_##type##Filter = type##FilterType::New(); \
512     std::ostringstream oss; \
513     oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan1; \
514     I_##type##Filter->SetRedIndex(this->GetParameterInt(oss.str())); \
515     oss.str(""); \
516     oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan2; \
517     I_##type##Filter->SetNIRIndex(this->GetParameterInt(oss.str())); \
518     I_##type##Filter->SetInput(inImage); \
519     m_FilterList->PushBack( I_##type##Filter ); \
520     m_ImageList->PushBack( I_##type##Filter->GetOutput() ); \
521     otbAppLogINFO(<<m_Map[GetSelectedItems("list")[idx]].item<<" added.");\
522 }
523
524 #define otbRadiometricSoilIndicesMacro( type ) \
525 { \
526     type##FilterType::Pointer I_##type##Filter = type##FilterType::New(); \
527     std::ostringstream oss; \
528     oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan1; \
529     I_##type##Filter->GetFunctor().SetRedIndex(this->GetParameterInt(oss.str()));\
530     oss.str(""); \
531     oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan2; \
532     I_##type##Filter->GetFunctor().SetGreenIndex(this->GetParameterInt(oss.str()));\
533     I_##type##Filter->SetInput(inImage); \
534     m_FilterList->PushBack( I_##type##Filter ); \
535     m_ImageList->PushBack( I_##type##Filter->GetOutput() ); \
536     otbAppLogINFO(<< m_Map[GetSelectedItems("list")[idx]].item << " added.");\ }
537

```

```

538 void DoExecute()
539 {
540
541     int nbChan = GetParameterImage("in")->GetNumberOfComponentsPerPixel();
542
543     if ( (this->GetParameterInt("channels.blue") <= nbChan)
544         && (this->GetParameterInt("channels.green") <= nbChan)
545         && (this->GetParameterInt("channels.red") <= nbChan)
546         && (this->GetParameterInt("channels.nir") <= nbChan)
547         && (this->GetParameterInt("channels.mir") <= nbChan))
548     {
549
550         m_FilterList = FilterListType::New();
551         m_ImageList = ImageListType::New();
552         m_Concatener = ImageListToVectorImageFilterType::New();
553
554         FloatVectorImageType* inImage = GetParameterImage("in");
555
556         for (unsigned int idx = 0; idx < GetSelectedItems("list").size(); ++idx)
557         {
558
559             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:NDVI")
560                 otbRadiometricVegetationIndicesMacro(NDVI);
561             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:TNDVI")
562                 otbRadiometricVegetationIndicesMacro(TNDVI);
563             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:RVI")
564                 otbRadiometricVegetationIndicesMacro(RVI);
565             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:SAVI")
566                 otbRadiometricVegetationIndicesMacro(SAVI);
567             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:TSAVI")
568                 otbRadiometricVegetationIndicesMacro(TSAVI);
569             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:MSAVI")
570                 otbRadiometricVegetationIndicesMacro(MSAVI);
571             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:MSAVI2")
572                 otbRadiometricVegetationIndicesMacro(MSAVI2);
573             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:GEMI")
574                 otbRadiometricVegetationIndicesMacro(GEMI);
575             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:IPVI")
576                 otbRadiometricVegetationIndicesMacro(IPVI);
577             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:LAIFromNDVILog")
578                 otbRadiometricVegetationIndicesMacro(LAIFromNDVILog);
579             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:LAIFromReflLinear")
580                 otbRadiometricVegetationIndicesMacro(LAIFromReflLinear);
581             if (m_Map[GetSelectedItems("list")[idx]].item == "Vegetation:LAIFromNDVIFormo")
582                 otbRadiometricVegetationIndicesMacro(LAIFromNDVIFormo);
583
584             if (m_Map[GetSelectedItems("list")[idx]].item == "Water:NDWI")
585                 otbRadiometricWaterIndicesMacro(NDWI);
586             if (m_Map[GetSelectedItems("list")[idx]].item == "Water:NDWI2")

```

```

587     otbRadiometricWaterIndicesMacro(NDWI2);
588     if (m_Map[GetSelectedItems("list")[idx]].item == "Water:MNDWI")
589         otbRadiometricWaterIndicesMacro(MNDWI);
590     if (m_Map[GetSelectedItems("list")[idx]].item == "Water:NDPI")
591         otbRadiometricWaterIndicesMacro(NDPI);
592     if (m_Map[GetSelectedItems("list")[idx]].item == "Water:NDTI")
593         otbRadiometricWaterIndicesMacro(NDTI);
594     if (m_Map[GetSelectedItems("list")[idx]].item == "Water:SRWI")
595         otbRadiometricWaterIndicesMacro(SRWI);
596
597     if (m_Map[GetSelectedItems("list")[idx]].item == "Soil:RI")
598         otbRadiometricSoilIndicesMacro(RI);
599     if (m_Map[GetSelectedItems("list")[idx]].item == "Soil:CI")
600         otbRadiometricSoilIndicesMacro(CI);
601     if (m_Map[GetSelectedItems("list")[idx]].item == "Soil:BI")
602         otbRadiometricSoilIndicesMacro(BI);
603     if (m_Map[GetSelectedItems("list")[idx]].item == "Soil:BI2")
604     {
605         BI2FilterType::Pointer I_BI2Filter = BI2FilterType::New();
606         std::ostringstream oss;
607         oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan1;
608         std::cout << "flag" << std::endl;
609         I_BI2Filter->GetFunctor().SetNIRIndex(this->GetParameterInt(oss.str()));
610         oss.str("");
611         oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan2;
612         I_BI2Filter->GetFunctor().SetRedIndex(this->GetParameterInt(oss.str()));
613         oss.str("");
614         oss<<"channels."<<m_Map[GetSelectedItems("list")[idx]].chan3;
615         I_BI2Filter->GetFunctor().SetGreenIndex(this->GetParameterInt(oss.str()));
616         I_BI2Filter->SetInput(inImage);
617         m_FilterList->PushBack( I_BI2Filter );
618         m_ImageList->PushBack( I_BI2Filter->GetOutput() );
619         otbAppLogINFO(<< m_Map[GetSelectedItems("list")[idx]].item << " added.");
620     }
621
622 }
623
624 if( m_ImageList->Size() == 0 )
625 {
626     itkExceptionMacro(<< "No indices selected...");
627 }
628
629 m_Concatener->SetInput( m_ImageList );
630 m_Concatener->UpdateOutputInformation();
631
632 SetParameterOutputImage("out", m_Concatener->GetOutput());
633 }
634 else
635 {

```

```
636     itkExceptionMacro(<< "At least one needed channel has an invalid index");
637     }
638 }
639 }
640
641 FilterListType::Pointer      m_FilterList;
642 ImageListType::Pointer      m_ImageList;
643 ImageListToVectorImageFilterType::Pointer m_Concatener;
644 std::vector<indiceSpec>      m_Map;
645
646 };
647
648 }
649 }
650
651 OTB_APPLICATION_EXPORT(otb::Wrapper::RadiometricIndices)
```

Στατιστικό μοντέλο εκμάθησης svm από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif

```
svm_type c_svc
kernel_type linear
nr_class 4
total_sv 18
rho 3.507969172 -0.5265227112 0.9011764676 -0.8192977566 0.4121355246 1.113857467
label 1 2 3 4
nr_sv 4 7 1 6
0 0 0.1861299820566693 1:-0.22793816 2:-0.20506153 3:-1.3819143 4:-2.1551623
0.1096928826749781 0 0 1:0.52602005 2:0.16954096 3:-1.0769331 4:-2.0045798
1 0 0 1:0.87325764 2:0.22308162 3:-0.938223 4:-1.9708388
1 0.02810191306927319 0.1716242281284742 1:0.59056956 2:0.33626771 3:-1.0180635 4:-
1.9035034
-0 0.01975315088404521 1 1:0.51918638 2:0.74404782 3:0.22116554 4:-0.40536505
-0.3867525273272107 0 0 1:1.0923817 2:0.26742443 3:-0.44957125 4:-1.1177096
-0.7229403553477672 0 0 1:-0.13279749 2:-0.50712782 3:-0.65578693 4:-1.2628374
-0 0 0.629606093070924 1:1.1739084 2:1.088613 3:0.72650462 4:-0.30276313
-0 0.02186793169751746 0.04892227846298627 1:2.1184113 2:1.3085759 3:0.81207371
4:0.16380885
-1 0 0 1:1.1073231 2:0.40286994 3:-0.41858587 4:-1.1014111
-0 0 1 1:0.63491797 2:0.37391514 3:0.43663087 4:-0.41614082
-0.02810191306927319 -0.04162108258156267 0.06634581684319824 1:-0.28115812 2:0.29490373
3:-0.66247219 4:6.4799008
-0.3577542101851436 -1 -0 1:-0.59130108 2:-0.44805753 3:-0.036175009 4:-0.19218913
-0 -0 -0.03315544947773136 1:-0.37998575 2:-0.034882613 3:0.63292205 4:0.64792341
-0 -0.5548679437427684 -0 1:-0.48404604 2:-0.32037449 3:0.086971149 4:-0.020812383
-0 -0 -0.03319036736546688 1:1.5262835 2:2.4847221 3:3.29074 4:3.0004454
-0 -0.1236604277911418 -0 1:5.4023557 2:5.4113045 3:4.509882 4:2.3038111
-0 -1 -0 1:1.3873367 2:1.3147218 3:1.4126949 4:0.62524724
```

Στατιστικό μοντέλο εκμάθησης decision tree από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif

```
%YAML:1.0
my_tree: !!opencv-ml-tree
  is_classifier: 1
  var_all: 4
  var_count: 4
  ord_var_count: 4
  cat_var_count: 0
  training_params:
    use_surrogates: 1
    max_categories: 5
    max_depth: 25
    min_sample_count: 1
    cross_validation_folds: 5
    use_1se_rule: 1
    truncate_pruned_tree: 1
  priors: !!opencv-matrix
    rows: 1
    cols: 4
    dt: d
    data: [ 1., 1., 1., 1. ]
  var_type: [ 0, 0, 0, 0 ]
  cat_count: !!opencv-matrix
    rows: 1
    cols: 1
    dt: i
    data: [ 4 ]
  cat_map: !!opencv-matrix
    rows: 1
    cols: 4
    dt: i
    data: [ 1, 2, 3, 4 ]
  best_tree_idx: 1
  nodes:
    -
      depth: 0
      sample_count: 365
      value: 2.
      norm_class_idx: 1
      Tn: 4
      complexity: 2
      alpha: 77.
      node_risk: 261.
      tree_risk: 129.
      tree_error: 37.
      splits:
        - { var:3, quality:1.8231007385253906e+02,
```

```
le:-1.0252189636230469e-01 }  
-  
depth: 1  
sample_count: 190  
value: 2.  
norm_class_idx: 1  
Tn: 3  
complexity: 1  
alpha: 69.  
node_risk: 88.  
tree_risk: 69.  
tree_error: 19.  
splits:  
- { var:2, quality:190., le:-7.2315478324890137e-01 }  
-  
depth: 2  
sample_count: 88  
value: 1.  
norm_class_idx: 0  
Tn: 2147483647  
complexity: 1  
alpha: 0.  
node_risk: 0.  
tree_risk: 0.  
tree_error: 0.  
-  
depth: 2  
sample_count: 102  
value: 2.  
norm_class_idx: 1  
Tn: 2147483647  
complexity: 1  
alpha: 0.  
node_risk: 0.  
tree_risk: 0.  
tree_error: 0.  
-  
depth: 1  
sample_count: 175  
value: 4.  
norm_class_idx: 3  
Tn: 2  
complexity: 1  
alpha: 58.  
node_risk: 78.  
tree_risk: 60.  
tree_error: 18.  
splits:  
- { var:2, quality:1.7108081054687500e+02,
```



```
le:-2.8275257349014282e-01 }
```

```
-
```

```
depth: 2  
sample_count: 76  
value: 3.  
norm_class_idx: 2  
Tn: 2147483647  
complexity: 1  
alpha: 0.  
node_risk: 0.  
tree_risk: 0.  
tree_error: 0.
```

```
-
```

```
depth: 2  
sample_count: 99  
value: 4.  
norm_class_idx: 3  
Tn: 1  
complexity: 1  
alpha: 1.  
node_risk: 2.  
tree_risk: 2.  
tree_error: 0.
```

Στατιστικό μοντέλο εκμάθησης k-nearest neighbors από τα δεδομένα εκπαίδευσης της QB_1_ortho.tif

```

K=2
IsRegression=0
1 0.574725 -0.0881969 -1.20999 -2.0242
1 -0.022378 -0.37621 -1.46891 -2.22004
1 -0.351686 -0.766349 -1.73658 -2.49761
1 -0.332374 -0.680991 -1.68901 -2.39247
1 -0.908954 -1.18107 -2.0573 -2.59099
1 -0.176886 -0.33567 -1.36647 -2.20127
1 -0.648679 -1.12469 -1.95873 -2.56343
1 -0.232871 -0.267891 -1.44247 -2.18412
1 0.405979 -0.0796854 -1.2968 -2.06952
1 -0.193807 -0.151114 -1.45318 -2.15015
1 -1.09792 -1.0947 -1.96131 -2.53312
1 -0.668586 -0.620692 -1.80315 -2.34966
1 0.62145 0.191105 -1.06501 -2.01587
1 -0.0366013 -0.434384 -1.54888 -2.37808
1 -0.753179 -1.17507 -1.8552 -2.50154
1 0.224793 -0.143902 -1.30119 -2.11827
1 -0.433965 -0.703328 -1.8526 -2.46678
1 -0.142029 -0.218354 -1.68271 -2.25684
1 0.086267 -0.238549 -1.29917 -2.11066
1 0.460608 0.222302 -1.05358 -2.03576
1 0.139833 -0.0284115 -1.35688 -2.17913
1 0.0521634 -0.330871 -1.58549 -2.36213
1 -0.0114884 -0.636482 -1.39455 -2.31169
1 0.0914464 -0.224875 -1.26395 -2.28469
1 0.516769 -0.00733234 -1.2115 -1.99489
1 -0.468755 -0.736998 -1.76217 -2.47997
1 0.400949 -0.0558805 -1.3805 -2.2383
1 0.76966 0.236983 -1.03797 -1.9613
1 0.14714 -0.215034 -1.26838 -2.15191
1 0.366689 -0.0345708 -1.20396 -1.93715
1 0.830633 0.339833 -0.963085 -1.85064
1 0.379129 0.267144 -0.935238 -1.96405
1 -0.213407 -0.240034 -1.33651 -2.26726
1 0.180845 -0.0937316 -1.30241 -2.13033
1 0.211348 -0.0505104 -1.29306 -2.17794
1 -0.265313 -0.469866 -1.61367 -2.4204
1 0.292697 -0.0348631 -1.30184 -1.97095
1 0.0313875 -0.499743 -1.59604 -2.33827
1 -0.326935 -0.461983 -1.68836 -2.33176
1 0.00088308 -0.481807 -1.58442 -2.38833
1 0.116096 -0.245472 -1.38558 -2.24225
1 -0.357434 -0.86108 -1.71639 -2.38399
1 -0.40765 -0.321467 -1.68058 -2.15995
1 0.205087 -0.221099 -1.30998 -2.21674

```

1	-0.116536	-0.0167318	-1.37937	-2.14379
1	0.130091	-0.00662647	-1.33587	-2.13541
1	-0.433089	-0.843046	-1.83232	-2.57758
1	-0.0957384	-0.42111	-1.58126	-2.30329
1	-0.155174	-0.331802	-1.48804	-2.39542
1	-0.289324	-0.859465	-1.71455	-2.33432
1	-0.550205	-0.82252	-1.77805	-2.41448
1	0.0969434	-0.619607	-1.67895	-2.23576
1	-0.458237	-0.885543	-1.98685	-2.5355
1	-0.229786	-0.514742	-1.59113	-2.28433
1	0.0143373	-0.206362	-1.49828	-2.27013
1	-1.27097	-1.3088	-2.28351	-2.71162
1	-0.035992	-0.450217	-1.57174	-2.27808
1	0.154576	-0.198827	-1.32707	-2.21071
1	-0.894317	-1.18212	-2.08826	-2.7151
1	0.300733	-0.266175	-1.43292	-2.2025
1	-0.240531	-0.849989	-1.72815	-2.44654
1	-0.266908	-0.441193	-1.66606	-2.34202
1	-0.447576	-0.974558	-2.05061	-2.48534
1	-0.693219	-0.915698	-1.91415	-2.5113
1	0.269362	-0.182266	-1.34939	-2.28826
1	-1.0413	-1.32296	-2.04845	-2.61901
1	0.365129	-0.0528142	-1.3704	-2.24416
1	-0.200637	-0.59139	-1.62441	-2.37652
1	-0.204055	-0.704218	-1.65096	-2.41923
1	-0.236348	-0.340209	-1.67275	-2.3254
1	-0.196328	-0.597947	-1.76546	-2.39854
1	-0.514684	-0.73125	-1.79608	-2.55819
1	0.219878	-0.338106	-1.38219	-2.08307
1	-1.07384	-1.34417	-2.16045	-2.70176
1	-1.06918	-1.45878	-2.32433	-2.73646
1	-1.02447	-1.61509	-2.06594	-2.68409
1	-1.22824	-1.53375	-2.36092	-2.74156
1	-1.26801	-1.65075	-2.36779	-2.75815
1	-1.35813	-1.36344	-2.12787	-2.67625
1	-0.531562	-0.903996	-1.98951	-2.58143
1	-0.584038	-0.741943	-1.89109	-2.4385
1	-0.865183	-1.35586	-2.12411	-2.62144
1	-1.12198	-0.983162	-2.03544	-2.57405
1	-1.04609	-1.47274	-2.24412	-2.64549
1	-1.192	-1.3126	-2.26982	-2.74184
1	-0.906714	-1.22618	-1.99905	-2.61249
1	-0.786752	-1.12849	-1.94826	-2.54349
1	-1.15163	-1.54149	-2.25179	-2.68475
1	-0.847334	-1.25116	-1.95667	-2.45929
1	-0.904418	-1.22994	-1.99272	-2.67214
1	-1.16642	-1.33657	-2.19443	-2.63055
1	-0.919185	-1.13089	-1.90189	-2.55845
1	-1.09308	-1.29666	-2.20576	-2.73221

1	-0.945345	-1.0005	-1.98335	-2.5929
2	1.16182	0.301988	0.204952	-0.467225
2	1.09342	0.842161	0.365701	-0.502169
2	0.278324	0.10611	-0.0124146	-0.508523
2	0.820555	0.278554	-0.294896	-1.07848
2	0.976692	0.234565	-0.308716	-1.00909
2	1.13447	0.200337	-0.33678	-0.977574
2	1.30493	0.622958	0.445604	-0.354899
2	1.09238	0.267424	-0.449571	-1.11771
2	0.958183	0.271522	-0.116354	-0.929439
2	1.11368	0.67828	0.217492	-0.625389
2	1.25256	0.994732	0.522204	-0.367262
2	1.17605	0.651031	-0.158686	-0.846974
2	0.560657	0.417752	0.166965	-0.591552
2	1.47821	0.769287	-0.108762	-0.88024
2	1.29655	0.590429	0.0260888	-0.845221
2	1.26035	0.863928	0.038249	-0.743096
2	1.4952	0.646834	-0.0930711	-0.841023
2	1.20432	0.3633	-0.376933	-1.11728
2	0.81088	-0.0829883	-0.494147	-1.09792
2	1.09145	0.337484	-0.173729	-0.984555
2	0.951644	0.239136	-0.281334	-0.903881
2	1.64194	1.14265	0.856453	-0.176156
2	0.831105	0.503104	0.438179	-0.507885
2	0.704899	0.524546	0.158006	-1.00265
2	0.913426	0.22308	-0.271561	-1.0719
2	1.17581	0.383846	-0.335184	-1.03497
2	1.61023	0.696826	-0.0600479	-0.790539
2	0.961315	0.470264	-0.268986	-0.841832
2	0.660187	0.493392	-0.0195916	-0.945913
2	1.7382	1.10736	0.430927	-0.364697
2	1.97215	1.14454	0.669528	-0.160304
2	2.00626	1.13199	1.02946	-0.0105501
2	1.48299	0.743856	0.611936	-0.304951
2	1.16952	0.392904	-0.266171	-0.95703
2	1.14876	0.404954	-0.16485	-0.971182
2	1.5591	0.919062	-0.0434439	-0.753572
2	0.868251	0.484167	-0.138561	-0.898795
2	1.58356	0.487956	-0.119401	-0.891603
2	1.12589	0.444968	-0.222631	-0.975667
2	0.918708	0.428841	-0.20195	-0.972942
2	7.54946	5.9311	3.56332	1.54803
2	1.12265	0.754863	0.119602	-0.872945
2	1.18593	0.783698	0.616392	-0.547771
2	1.93862	1.15662	0.655246	-0.200582
2	1.377	1.38203	0.86445	-0.219578
2	1.74982	1.35833	0.945507	-0.351688
2	0.589494	-0.29812	-0.62419	-1.09883
2	1.15213	0.41788	-0.185128	-1.00051

2	1.54246	0.71484	-0.120138	-0.961363
2	1.67164	0.750825	0.140288	-0.737457
2	1.61585	0.779698	0.0199872	-0.737103
2	1.26249	0.190721	-0.274742	-0.973636
2	1.0398	0.1061	-0.304742	-1.04566
2	1.21486	0.0951321	-0.183424	-1.02126
2	1.13333	0.327965	-0.115358	-1.04082
2	0.740907	0.428541	0.763889	-0.28699
2	0.931922	0.705646	0.21353	-0.442447
2	1.00199	0.437864	-0.0536388	-0.798586
2	1.93716	1.37514	0.257141	-0.667901
2	1.06742	0.428318	-0.371158	-1.00728
2	1.08507	0.515876	-0.117225	-0.917967
2	1.67308	0.826899	0.208859	-0.739673
2	1.55579	0.827611	0.162272	-0.754151
2	1.65642	0.73411	0.0761101	-0.760124
2	1.20074	0.304655	-0.290581	-1.02085
2	0.769346	0.165981	-0.395976	-1.00481
2	1.26991	0.496916	-0.0693222	-0.843358
2	1.02188	0.636656	0.167972	-0.561658
2	2.84911	2.40747	1.21138	-0.0726716
2	0.992768	0.339226	-0.0496079	-1.01492
2	1.20127	0.64041	-0.181977	-0.797416
2	1.74146	0.867594	0.0322885	-0.805758
2	1.78314	0.916626	-0.0716555	-0.792694
2	1.09036	0.702448	0.0364644	-0.73571
2	1.0124	0.440533	-0.165382	-0.937233
2	1.00254	0.469601	-0.152664	-1.04464
2	0.930762	0.260478	-0.372217	-1.02292
2	1.1502	-0.0302739	-0.371978	-1.15679
2	0.508091	0.0615902	-0.330452	-1.10584
2	0.740267	0.374965	0.166455	-0.515893
2	1.37541	0.639905	0.0307543	-0.696749
2	1.18977	0.365857	5.29468e-05	-0.952538
2	1.28535	0.367046	-0.121055	-0.941216
2	1.11102	0.250351	-0.128949	-0.986253
2	1.38534	0.603172	-0.146085	-0.717871
2	1.81754	1.03322	0.109969	-0.703387
2	0.735778	0.453694	-0.000252421	-0.673532
2	0.845039	0.522546	-0.239342	-1.00395
2	1.30696	0.522459	-0.184735	-0.896823
2	1.63575	0.696442	0.0412476	-0.772224
2	2.18026	1.26736	0.340355	-0.498139
2	1.06122	0.507683	0.275017	-0.608171
2	1.13436	0.478694	0.210237	-0.543821
2	1.66959	0.614245	0.232447	-0.17118
2	1.28088	0.651164	0.151389	-0.637503
2	1.0941	0.485175	-0.127759	-0.954145
2	1.65608	0.618866	-0.161255	-0.834681

2	0.97807	0.74091	0.21265	-0.439929
2	1.4845	0.918396	0.562583	-0.239589
2	1.2807	0.58221	0.392459	-0.572181
2	1.31574	0.701177	0.0471128	-0.739479
3	-0.885255	-0.272057	-1.14023	7.83991
3	-0.839833	-0.198667	-1.15726	7.90203
3	-0.983224	-0.243095	-1.16827	7.7705
3	-0.82236	-0.245599	-1.14209	7.87211
3	-0.925635	-0.414201	-1.19319	8.32379
3	-0.91745	-0.38303	-1.17914	7.85909
3	-0.932413	-0.420041	-1.21966	8.28749
3	-1.01069	-0.42147	-1.2169	7.94991
3	-1.00173	-0.284925	-1.18849	7.8891
3	-1.08722	-0.426212	-1.35625	8.28834
3	-1.00276	-0.200142	-1.13064	8.13053
3	-0.773608	-0.110253	-0.962531	7.74129
3	-0.792412	-0.0254358	-0.958486	7.30213
3	-0.590802	-0.183582	-0.931649	7.82902
3	-1.05889	-0.170131	-1.15115	8.12434
3	-0.584059	-0.0486109	-0.887609	7.36248
3	-0.44018	0.177405	-0.766444	6.77766
3	-0.927666	-0.318912	-0.969681	7.55039
3	-0.859293	-0.0813338	-0.936786	7.19034
3	-0.791996	-0.146615	-0.824127	7.16295
3	-1.03557	-0.293462	-1.02127	8.14843
3	-0.966628	-0.288504	-0.990665	7.42047
3	-1.10802	-0.458388	-1.13495	8.03888
3	-1.10802	-0.326753	-1.03367	7.99596
3	-0.873824	-0.167208	-1.05502	7.51422
3	-0.568652	-0.0276038	-0.836368	7.20202
3	-0.643421	-0.0954027	-0.870387	7.37694
3	-0.893874	-0.29914	-1.06182	7.71015
3	-0.205329	0.262474	-0.622806	7.09317
3	-0.702519	-0.113792	-0.917645	7.3541
3	-0.715615	-0.13506	-0.944147	7.31605
3	-0.357187	0.27751	-0.632851	6.86917
3	-0.408552	0.225575	-0.629244	6.99896
3	-0.721937	0.0569362	-0.753991	7.09466
3	-0.58605	-0.0877905	-0.859043	7.23671
3	-0.44794	0.190714	-0.684105	6.97927
3	-0.401109	0.252335	-0.697641	6.94159
3	-0.558594	0.0582214	-0.757767	6.99552
3	-0.485138	0.171314	-0.617721	6.65823
3	-0.689097	-0.199093	-0.857294	6.94947
3	-1.05447	-0.473472	-1.25449	8.13199
3	-0.50356	-0.0447441	-0.820568	7.26463
3	-0.48359	-0.0227584	-0.888843	7.29597
3	-0.888177	-0.0262579	-0.80402	7.36269
3	-0.704552	0.0419052	-0.707672	6.87742

3	-0.629351	-0.131363	-0.841193	7.58256
3	-0.576046	-0.144151	-0.878248	7.43875
3	-0.81517	-0.136168	-1.00245	7.5529
3	-1.00538	-0.21801	-1.11533	7.77199
3	-0.849204	-0.151276	-0.885989	7.45589
3	-1.09285	-0.385761	-1.18926	7.8679
3	-0.943997	-0.275462	-1.00574	7.89042
3	-0.788341	-0.175374	-0.949018	7.68247
3	-1.18612	-0.297086	-1.05806	7.71114
3	-0.948232	-0.092892	-0.880419	7.51872
3	-0.948357	-0.128148	-0.839199	7.41027
3	-0.857934	-0.0531342	-1.0405	7.56314
3	-0.963819	-0.327957	-1.0649	7.7571
3	-1.00262	-0.0922129	-0.825183	7.43213
3	-0.945772	-0.0221802	-0.814875	7.12792
3	-0.919266	-0.312411	-1.0316	7.6625
3	-0.6957	0.0417293	-0.976397	7.51494
3	-1.00694	-0.296627	-1.24055	8.08802
3	-1.09291	-0.533371	-1.23954	7.96895
3	-0.806886	-0.368272	-1.09561	7.77184
3	-0.831691	-0.229008	-0.996002	7.40103
3	-0.502341	0.208091	-0.491513	6.79714
3	-0.255759	0.231815	-0.482342	6.88269
3	-0.570725	-0.117616	-0.886617	7.43735
3	-0.825409	-0.47386	-1.31822	7.9134
3	-0.911601	-0.377936	-1.00841	7.67443
3	-0.500182	0.307761	-0.596504	6.51966
3	-0.4136	0.168616	-0.626289	6.75218
3	-1.02499	-0.22402	-1.03157	7.4177
3	-1.02685	-0.387036	-1.11541	7.72274
3	-0.281158	0.294904	-0.662472	6.4799
3	-0.947121	-0.190129	-1.03931	7.46715
3	-0.901101	-0.140458	-0.887802	7.11036
3	-1.04383	-0.157577	-1.02524	7.36241
3	-0.922258	-0.44905	-1.15497	7.3282
4	-0.520479	-0.0186776	0.227191	0.172485
4	-0.369698	-0.468761	0.234697	0.117127
4	-0.429207	-0.449232	0.513517	0.33815
4	-0.782645	-0.292518	0.178444	0.0936377
4	-0.70806	-0.534847	-0.00600352	-0.182685
4	-0.529125	0.216566	1.34996	0.709945
4	-0.173698	0.385934	1.45291	0.810096
4	-0.18454	0.539835	0.904614	0.476157
4	0.214533	0.736663	1.83652	1.37839
4	-0.745125	-0.400346	0.0726365	-0.0992708
4	-0.484046	-0.320374	0.0869711	-0.0208124
4	-0.199161	0.336539	0.815239	0.403461
4	0.398666	1.24127	2.33079	1.51529
4	0.557607	1.49632	1.82344	1.72484

4	0.513263	1.20168	1.80964	1.53051
4	0.104369	1.03162	2.21029	1.61562
4	0.449561	1.24651	1.86452	1.28244
4	0.490398	1.46156	2.54525	1.93796
4	0.399421	1.53757	2.67866	1.82925
4	2.42471	3.50982	4.15126	3.61871
4	2.40537	3.62338	3.92644	3.56904
4	2.02657	2.8898	3.6171	2.94962
4	2.17085	3.48756	4.13057	3.75779
4	2.84113	4.31756	4.79705	4.30194
4	2.77916	4.13445	4.70061	4.1195
4	2.93763	4.00779	4.82487	4.16684
4	2.90885	3.92804	5.03317	4.2046
4	3.12913	3.88854	4.91315	3.91495
4	2.94964	3.84551	4.68247	3.80553
4	1.69174	2.75615	4.06187	3.31973
4	2.7668	3.83011	4.77895	4.1334
4	2.78188	4.14631	5.09679	4.18788
4	0.0985518	0.961267	1.70042	0.947978
4	0.242051	1.22502	1.98395	1.16922
4	0.769878	1.08908	2.19675	1.63649
4	0.676413	1.03814	1.68336	1.47904
4	0.205651	0.747644	1.56836	1.18323
4	0.320561	0.932617	1.86302	0.880253
4	0.547651	1.177	2.12657	1.06627
4	0.254862	0.657062	1.48459	1.09282
4	-0.869178	-0.809863	0.0935842	-0.289969
4	0.966118	1.37588	2.4189	1.36339
4	0.930253	1.62066	2.65501	1.70502
4	0.538599	1.50069	2.53158	1.6966
4	0.115727	0.701446	1.48792	0.831284
4	0.854951	1.31425	2.26679	1.25478
4	0.837174	1.96448	3.007	1.93076
4	1.06082	1.96019	2.69227	1.85271
4	0.60354	1.11328	2.03942	1.17799
4	0.656481	1.39707	2.43772	1.43118
4	1.14806	1.93947	2.54678	1.88653
4	0.478705	1.04246	1.9496	1.06183
4	0.519715	1.30888	2.33309	1.3959
4	0.800027	1.93891	2.75475	1.93707
4	0.414855	1.1204	1.79678	1.04624
4	1.67998	2.2709	2.92123	1.91292
4	1.25009	1.57968	1.94379	1.68167
4	0.787615	1.34598	1.77346	1.62868
4	0.818337	1.17624	1.96813	1.09713
4	0.940115	1.52222	2.75981	1.71296
4	0.602995	1.10184	2.5067	1.86318
4	0.110488	0.441506	1.3346	1.17055
4	0.185623	0.715133	1.87274	1.03934

4	-0.161239	0.316723	1.80092	1.03884
4	0.265627	0.711051	2.2613	1.3102
4	0.781466	1.56379	2.63259	1.57743
4	0.545806	1.23873	2.52354	1.57003
4	0.482669	1.616	3.21497	2.03753
4	0.970949	1.74237	2.71975	1.87092
4	0.0240846	0.641891	1.59463	0.908065
4	0.393346	0.48876	1.66566	0.815802
4	0.880506	1.34629	2.41643	1.68996
4	0.045314	0.368563	1.31803	0.628974
4	0.293282	0.951824	2.17415	1.17568
4	0.334705	1.4272	2.81983	1.68217
4	0.0853829	1.06576	2.2059	1.34721
4	0.763149	1.37034	2.50762	1.57881
4	0.845753	1.80017	2.77887	2.09751
4	0.220164	0.606665	1.57617	0.886662
4	-0.0702514	0.207505	1.16813	0.739303
4	-0.147434	0.345253	1.31116	0.598994
4	-0.0157324	0.490966	1.55385	0.829062
4	0.228002	0.930987	2.21363	1.23543
4	0.527172	1.35042	2.69953	1.61331
4	0.912723	1.65605	2.65798	1.60173
4	0.205977	0.813298	1.71451	1.11405
4	0.653249	1.63003	3.21367	2.08656
4	0.806283	1.99421	3.15221	1.87307
4	0.912635	2.02522	3.12638	1.93284
4	1.6523	2.7365	4.31546	3.09312
4	1.70364	2.78301	4.12033	3.00214
4	0.961147	2.28917	3.33475	2.20652
4	2.05299	3.79665	5.57516	4.12493
4	1.88615	2.93239	4.08398	2.80346
4	2.02833	2.81397	3.67878	2.88088
4	1.76007	2.65333	3.62972	2.89343
4	2.29719	4.05061	5.90472	4.58269
4	2.77169	4.23047	6.14181	4.63368
4	0.967596	1.76315	3.42445	2.54527

Στατιστικά Μοντέλου Εκπαίδευσης για την Περιοχή της Θεσσαλίας

Application.logger (INFO) Precision of class [1] vs all: 1
 Application.logger (INFO) Recall of class [1] vs all: 1
 Application.logger (INFO) F-score of class [1] vs all: 1

Application.logger (INFO) Precision of class [2] vs all: 1
 Application.logger (INFO) Recall of class [2] vs all: 1
 Application.logger (INFO) F-score of class [2] vs all: 1

Application.logger (INFO) Precision of class [3] vs all: 1
 Application.logger (INFO) Recall of class [3] vs all: 1
 Application.logger (INFO) F-score of class [3] vs all: 1

Application.logger (INFO) Precision of class [4] vs all: 1
 Application.logger (INFO) Recall of class [4] vs all: 1
 Application.logger (INFO) F-score of class [4] vs all: 1

Application.logger (INFO) Precision of class [5] vs all: 1
 Application.logger (INFO) Recall of class [5] vs all: 1
 Application.logger (INFO) F-score of class [5] vs all: 1

Application.logger (INFO) Precision of class [6] vs all: 1
 Application.logger (INFO) Recall of class [6] vs all: 1
 Application.logger (INFO) F-score of class [6] vs all: 1

Application.logger (INFO) Precision of class [7] vs all: 1
 Application.logger (INFO) Recall of class [7] vs all: 1
 Application.logger (INFO) F-score of class [7] vs all: 1

Application.logger (INFO) Precision of class [8] vs all: 1
 Application.logger (INFO) Recall of class [8] vs all: 1
 Application.logger (INFO) F-score of class [8] vs all: 1

Application.logger (INFO) Precision of class [9] vs all: 1
 Application.logger (INFO) Recall of class [9] vs all: 1
 Application.logger (INFO) F-score of class [9] vs all: 1

Application.logger (INFO) Precision of class [10] vs all: 0
 Application.logger (INFO) Recall of class [10] vs all: 0
 Application.logger (INFO) F-score of class [10] vs all: 0

Application.logger (INFO) Precision of class [11] vs all: 1
 Application.logger (INFO) Recall of class [11] vs all: 0.8
 Application.logger (INFO) F-score of class [11] vs all: 0.888889

Application.logger (INFO) Precision of class [12] vs all: 0.666667
 Application.logger (INFO) Recall of class [12] vs all: 0.666667
 Application.logger (INFO) F-score of class [12] vs all: 0.666667

Application.logger (INFO) Precision of class [13] vs all: 0.636364
Application.logger (INFO) Recall of class [13] vs all: 0.777778
Application.logger (INFO) F-score of class [13] vs all: 0.7

Application.logger (INFO) Precision of class [14] vs all: 1
Application.logger (INFO) Recall of class [14] vs all: 1
Application.logger (INFO) F-score of class [14] vs all: 1

Application.logger (INFO) Precision of class [15] vs all: 1
Application.logger (INFO) Recall of class [15] vs all: 1
Application.logger (INFO) F-score of class [15] vs all: 1

Application.logger (INFO) Precision of class [16] vs all: 1
Application.logger (INFO) Recall of class [16] vs all: 1
Application.logger (INFO) F-score of class [16] vs all: 1

Application.logger (INFO) Precision of class [17] vs all: 0.769231
Application.logger (INFO) Recall of class [17] vs all: 1
Application.logger (INFO) F-score of class [17] vs all: 0.869565

Application.logger (INFO) Precision of class [18] vs all: 0
Application.logger (INFO) Recall of class [18] vs all: 0
Application.logger (INFO) F-score of class [18] vs all: 0

Application.logger (INFO) Precision of class [19] vs all: 0
Application.logger (INFO) Recall of class [19] vs all: 0
Application.logger (INFO) F-score of class [19] vs all: 0

Application.logger (INFO) Precision of class [20] vs all: 0.909091
Application.logger (INFO) Recall of class [20] vs all: 1
Application.logger (INFO) F-score of class [20] vs all: 0.952381

Application.logger (INFO) Precision of class [21] vs all: 0.666667
Application.logger (INFO) Recall of class [21] vs all: 0.285714
Application.logger (INFO) F-score of class [21] vs all: 0.4

Application.logger (INFO) Precision of class [22] vs all: 1
Application.logger (INFO) Recall of class [22] vs all: 0.142857
Application.logger (INFO) F-score of class [22] vs all: 0.25

Application.logger (INFO) Precision of class [23] vs all: 1
Application.logger (INFO) Recall of class [23] vs all: 0.25
Application.logger (INFO) F-score of class [23] vs all: 0.4

Application.logger (INFO) Precision of class [24] vs all: 1
Application.logger (INFO) Recall of class [24] vs all: 1
Application.logger (INFO) F-score of class [24] vs all: 1

Application.logger (INFO) Precision of class [25] vs all: 1

Application.logger (INFO) Recall of class [25] vs all: 1
Application.logger (INFO) F-score of class [25] vs all: 1

Application.logger (INFO) Precision of class [26] vs all: 0.75
Application.logger (INFO) Recall of class [26] vs all: 0.6
Application.logger (INFO) F-score of class [26] vs all: 0.666667

Application.logger (INFO) Precision of class [27] vs all: 0.5
Application.logger (INFO) Recall of class [27] vs all: 0.142857
Application.logger (INFO) F-score of class [27] vs all: 0.222222

Application.logger (INFO) Precision of class [28] vs all: 1
Application.logger (INFO) Recall of class [28] vs all: 1
Application.logger (INFO) F-score of class [28] vs all: 1

Application.logger (INFO) Precision of class [29] vs all: 1
Application.logger (INFO) Recall of class [29] vs all: 1
Application.logger (INFO) F-score of class [29] vs all: 1

Application.logger (INFO) Precision of class [30] vs all: 0.9
Application.logger (INFO) Recall of class [30] vs all: 1
Application.logger (INFO) F-score of class [30] vs all: 0.947368

Application.logger (INFO) Precision of class [31] vs all: 1
Application.logger (INFO) Recall of class [31] vs all: 1
Application.logger (INFO) F-score of class [31] vs all: 1

Application.logger (INFO) Precision of class [32] vs all: 1
Application.logger (INFO) Recall of class [32] vs all: 1
Application.logger (INFO) F-score of class [32] vs all: 1

Application.logger (INFO) Precision of class [33] vs all: 1
Application.logger (INFO) Recall of class [33] vs all: 1
Application.logger (INFO) F-score of class [33] vs all: 1

Application.logger (INFO) Precision of class [34] vs all: 0
Application.logger (INFO) Recall of class [34] vs all: 0
Application.logger (INFO) F-score of class [34] vs all: 0

Application.logger (INFO) Precision of class [35] vs all: 0.615385
Application.logger (INFO) Recall of class [35] vs all: 1
Application.logger (INFO) F-score of class [35] vs all: 0.761905

Application.logger (INFO) Precision of class [36] vs all: 1
Application.logger (INFO) Recall of class [36] vs all: 1
Application.logger (INFO) F-score of class [36] vs all: 1

Application.logger (INFO) Precision of class [37] vs all: 0.909091
Application.logger (INFO) Recall of class [37] vs all: 1

Application.logger (INFO) F-score of class [37] vs all: 0.952381

Application.logger (INFO) Precision of class [38] vs all: 0.888889

Application.logger (INFO) Recall of class [38] vs all: 1

Application.logger (INFO) F-score of class [38] vs all: 0.941176

Application.logger (INFO) Precision of class [39] vs all: 0.909091

Application.logger (INFO) Recall of class [39] vs all: 1

Application.logger (INFO) F-score of class [39] vs all: 0.952381

Application.logger (INFO) Precision of class [40] vs all: 0.777778

Application.logger (INFO) Recall of class [40] vs all: 1

Application.logger (INFO) F-score of class [40] vs all: 0.875

Application.logger (INFO) Precision of class [41] vs all: 0.9

Application.logger (INFO) Recall of class [41] vs all: 1

Application.logger (INFO) F-score of class [41] vs all: 0.947368

Application.logger (INFO) Precision of class [42] vs all: 1

Application.logger (INFO) Recall of class [42] vs all: 1

Application.logger (INFO) F-score of class [42] vs all: 1

Application.logger (INFO) Precision of class [43] vs all: 0.8

Application.logger (INFO) Recall of class [43] vs all: 0.666667

Application.logger (INFO) F-score of class [43] vs all: 0.727273

Application.logger (INFO) Precision of class [44] vs all: 0.75

Application.logger (INFO) Recall of class [44] vs all: 1

Application.logger (INFO) F-score of class [44] vs all: 0.857143

Application.logger (INFO) Precision of class [45] vs all: 1

Application.logger (INFO) Recall of class [45] vs all: 1

Application.logger (INFO) F-score of class [45] vs all: 1

Application.logger (INFO) Precision of class [46] vs all: 1

Application.logger (INFO) Recall of class [46] vs all: 0.4

Application.logger (INFO) F-score of class [46] vs all: 0.571429

Application.logger (INFO) Precision of class [47] vs all: 1

Application.logger (INFO) Recall of class [47] vs all: 0.75

Application.logger (INFO) F-score of class [47] vs all: 0.857143

Application.logger (INFO) Precision of class [48] vs all: 0.5

Application.logger (INFO) Recall of class [48] vs all: 0.666667

Application.logger (INFO) F-score of class [48] vs all: 0.571429

Application.logger (INFO) Precision of class [49] vs all: 0.714286

Application.logger (INFO) Recall of class [49] vs all: 0.769231

Application.logger (INFO) F-score of class [49] vs all: 0.740741

Application.logger (INFO) Precision of class [50] vs all: 0.4
Application.logger (INFO) Recall of class [50] vs all: 0.363636
Application.logger (INFO) F-score of class [50] vs all: 0.380952

Application.logger (INFO) Precision of class [51] vs all: 0.4
Application.logger (INFO) Recall of class [51] vs all: 0.615385
Application.logger (INFO) F-score of class [51] vs all: 0.484848

Application.logger (INFO) Precision of class [52] vs all: 0.818182
Application.logger (INFO) Recall of class [52] vs all: 0.9
Application.logger (INFO) F-score of class [52] vs all: 0.857143

Application.logger (INFO) Precision of class [53] vs all: 0.882353
Application.logger (INFO) Recall of class [53] vs all: 1
Application.logger (INFO) F-score of class [53] vs all: 0.9375

Application.logger (INFO) Precision of class [54] vs all: 1
Application.logger (INFO) Recall of class [54] vs all: 0.916667
Application.logger (INFO) F-score of class [54] vs all: 0.956522

Application.logger (INFO) Precision of class [55] vs all: 1
Application.logger (INFO) Recall of class [55] vs all: 0.785714
Application.logger (INFO) F-score of class [55] vs all: 0.88

Application.logger (INFO) Precision of class [56] vs all: 0.866667
Application.logger (INFO) Recall of class [56] vs all: 0.928571
Application.logger (INFO) F-score of class [56] vs all: 0.896552

Application.logger (INFO) Precision of class [57] vs all: 1
Application.logger (INFO) Recall of class [57] vs all: 0.933333
Application.logger (INFO) F-score of class [57] vs all: 0.965517

Application.logger (INFO) Precision of class [58] vs all: 0.9375
Application.logger (INFO) Recall of class [58] vs all: 1
Application.logger (INFO) F-score of class [58] vs all: 0.967742

Application.logger (INFO) Precision of class [59] vs all: 0.857143
Application.logger (INFO) Recall of class [59] vs all: 0.705882
Application.logger (INFO) F-score of class [59] vs all: 0.774194

Application.logger (INFO) Precision of class [60] vs all: 0.928571
Application.logger (INFO) Recall of class [60] vs all: 1
Application.logger (INFO) F-score of class [60] vs all: 0.962963

Application.logger (INFO) Precision of class [61] vs all: 1
Application.logger (INFO) Recall of class [61] vs all: 1
Application.logger (INFO) F-score of class [61] vs all: 1

Application.logger (INFO) Precision of class [62] vs all: 1
Application.logger (INFO) Recall of class [62] vs all: 0.9375
Application.logger (INFO) F-score of class [62] vs all: 0.967742

Application.logger (INFO) Precision of class [63] vs all: 1
Application.logger (INFO) Recall of class [63] vs all: 1
Application.logger (INFO) F-score of class [63] vs all: 1

Application.logger (INFO) Precision of class [64] vs all: 1
Application.logger (INFO) Recall of class [64] vs all: 0.5
Application.logger (INFO) F-score of class [64] vs all: 0.666667

Application.logger (INFO) Precision of class [65] vs all: 0.666667
Application.logger (INFO) Recall of class [65] vs all: 1
Application.logger (INFO) F-score of class [65] vs all: 0.8

Application.logger (INFO) Precision of class [66] vs all: 0.9
Application.logger (INFO) Recall of class [66] vs all: 1
Application.logger (INFO) F-score of class [66] vs all: 0.947368

Application.logger (INFO) Precision of class [67] vs all: 1
Application.logger (INFO) Recall of class [67] vs all: 1
Application.logger (INFO) F-score of class [67] vs all: 1

Application.logger (INFO) Precision of class [68] vs all: 1
Application.logger (INFO) Recall of class [68] vs all: 1
Application.logger (INFO) F-score of class [68] vs all: 1

Application.logger (INFO) Precision of class [69] vs all: 1
Application.logger (INFO) Recall of class [69] vs all: 0.888889
Application.logger (INFO) F-score of class [69] vs all: 0.941176

Application.logger (INFO) Precision of class [70] vs all: 1
Application.logger (INFO) Recall of class [70] vs all: 1
Application.logger (INFO) F-score of class [70] vs all: 1

Application.logger (INFO) Precision of class [71] vs all: 0.888889
Application.logger (INFO) Recall of class [71] vs all: 1
Application.logger (INFO) F-score of class [71] vs all: 0.941176

Application.logger (INFO) Precision of class [72] vs all: 1
Application.logger (INFO) Recall of class [72] vs all: 0.75
Application.logger (INFO) F-score of class [72] vs all: 0.857143

Application.logger (INFO) Precision of class [73] vs all: 0.642857
Application.logger (INFO) Recall of class [73] vs all: 0.75
Application.logger (INFO) F-score of class [73] vs all: 0.692308

Application.logger (INFO) Precision of class [74] vs all: 1

Application.logger (INFO) Recall of class [74] vs all: 1
Application.logger (INFO) F-score of class [74] vs all: 1

Application.logger (INFO) Precision of class [75] vs all: 1
Application.logger (INFO) Recall of class [75] vs all: 1
Application.logger (INFO) F-score of class [75] vs all: 1

Application.logger (INFO) Precision of class [76] vs all: 1
Application.logger (INFO) Recall of class [76] vs all: 1
Application.logger (INFO) F-score of class [76] vs all: 1

Application.logger (INFO) Precision of class [77] vs all: 1
Application.logger (INFO) Recall of class [77] vs all: 1
Application.logger (INFO) F-score of class [77] vs all: 1

Application.logger (INFO) Precision of class [78] vs all: 1
Application.logger (INFO) Recall of class [78] vs all: 1
Application.logger (INFO) F-score of class [78] vs all: 1

Application.logger (INFO) Precision of class [79] vs all: 0.944444
Application.logger (INFO) Recall of class [79] vs all: 1
Application.logger (INFO) F-score of class [79] vs all: 0.971429

Application.logger (INFO) Precision of class [80] vs all: 1
Application.logger (INFO) Recall of class [80] vs all: 1
Application.logger (INFO) F-score of class [80] vs all: 1

Application.logger (INFO) Precision of class [81] vs all: 1
Application.logger (INFO) Recall of class [81] vs all: 1
Application.logger (INFO) F-score of class [81] vs all: 1

Application.logger (INFO) Precision of class [82] vs all: 0.75
Application.logger (INFO) Recall of class [82] vs all: 0.6
Application.logger (INFO) F-score of class [82] vs all: 0.666667

Application.logger (INFO) Precision of class [83] vs all: 0.666667
Application.logger (INFO) Recall of class [83] vs all: 0.923077
Application.logger (INFO) F-score of class [83] vs all: 0.774194

Application.logger (INFO) Precision of class [84] vs all: 0.5
Application.logger (INFO) Recall of class [84] vs all: 0.8
Application.logger (INFO) F-score of class [84] vs all: 0.615385

Application.logger (INFO) Precision of class [85] vs all: 1
Application.logger (INFO) Recall of class [85] vs all: 0.555556
Application.logger (INFO) F-score of class [85] vs all: 0.714286

Application.logger (INFO) Precision of class [86] vs all: 1
Application.logger (INFO) Recall of class [86] vs all: 0.923077

Application.logger (INFO) F-score of class [86] vs all: 0.96

Application.logger (INFO) Precision of class [87] vs all: 0.9375
Application.logger (INFO) Recall of class [87] vs all: 1
Application.logger (INFO) F-score of class [87] vs all: 0.967742

Application.logger (INFO) Precision of class [88] vs all: 0.62963
Application.logger (INFO) Recall of class [88] vs all: 0.894737
Application.logger (INFO) F-score of class [88] vs all: 0.73913

Application.logger (INFO) Precision of class [89] vs all: 1
Application.logger (INFO) Recall of class [89] vs all: 0.22222
Application.logger (INFO) F-score of class [89] vs all: 0.363636

Application.logger (INFO) Precision of class [90] vs all: 1
Application.logger (INFO) Recall of class [90] vs all: 0.7
Application.logger (INFO) F-score of class [90] vs all: 0.823529

Application.logger (INFO) Precision of class [91] vs all: 0.8
Application.logger (INFO) Recall of class [91] vs all: 0.923077
Application.logger (INFO) F-score of class [91] vs all: 0.857143

Application.logger (INFO) Precision of class [92] vs all: 1
Application.logger (INFO) Recall of class [92] vs all: 0.909091
Application.logger (INFO) F-score of class [92] vs all: 0.952381

Application.logger (INFO) Precision of class [93] vs all: 0.954545
Application.logger (INFO) Recall of class [93] vs all: 1
Application.logger (INFO) F-score of class [93] vs all: 0.976744

Application.logger (INFO) Precision of class [94] vs all: 0.769231
Application.logger (INFO) Recall of class [94] vs all: 0.714286
Application.logger (INFO) F-score of class [94] vs all: 0.740741

Application.logger (INFO) Precision of class [95] vs all: 0.8
Application.logger (INFO) Recall of class [95] vs all: 0.615385
Application.logger (INFO) F-score of class [95] vs all: 0.695652

Application.logger (INFO) Precision of class [96] vs all: 0.636364
Application.logger (INFO) Recall of class [96] vs all: 0.7
Application.logger (INFO) F-score of class [96] vs all: 0.666667

Application.logger (INFO) Precision of class [97] vs all: 0.9
Application.logger (INFO) Recall of class [97] vs all: 0.692308
Application.logger (INFO) F-score of class [97] vs all: 0.782609

Application.logger (INFO) Precision of class [98] vs all: 1
Application.logger (INFO) Recall of class [98] vs all: 1
Application.logger (INFO) F-score of class [98] vs all: 1

Application.logger (INFO) Precision of class [99] vs all: 1
Application.logger (INFO) Recall of class [99] vs all: 1
Application.logger (INFO) F-score of class [99] vs all: 1

Application.logger (INFO) Precision of class [100] vs all: 0.75
Application.logger (INFO) Recall of class [100] vs all: 0.545455
Application.logger (INFO) F-score of class [100] vs all: 0.631579

Application.logger (INFO) Precision of class [101] vs all: 0.882353
Application.logger (INFO) Recall of class [101] vs all: 0.9375
Application.logger (INFO) F-score of class [101] vs all: 0.909091

Application.logger (INFO) Precision of class [102] vs all: 0.733333
Application.logger (INFO) Recall of class [102] vs all: 1
Application.logger (INFO) F-score of class [102] vs all: 0.846154

Application.logger (INFO) Precision of class [103] vs all: 1
Application.logger (INFO) Recall of class [103] vs all: 1
Application.logger (INFO) F-score of class [103] vs all: 1

Application.logger (INFO) Precision of class [104] vs all: 0.909091
Application.logger (INFO) Recall of class [104] vs all: 1
Application.logger (INFO) F-score of class [104] vs all: 0.952381

Application.logger (INFO) Precision of class [105] vs all: 0.826087
Application.logger (INFO) Recall of class [105] vs all: 1
Application.logger (INFO) F-score of class [105] vs all: 0.904762

Application.logger (INFO) Precision of class [106] vs all: 1
Application.logger (INFO) Recall of class [106] vs all: 0.75
Application.logger (INFO) F-score of class [106] vs all: 0.857143

Application.logger (INFO) Precision of class [107] vs all: 1
Application.logger (INFO) Recall of class [107] vs all: 1
Application.logger (INFO) F-score of class [107] vs all: 1

Application.logger (INFO) Precision of class [108] vs all: 0.777778
Application.logger (INFO) Recall of class [108] vs all: 1
Application.logger (INFO) F-score of class [108] vs all: 0.875

Application.logger (INFO) Precision of class [109] vs all: 0.8
Application.logger (INFO) Recall of class [109] vs all: 0.923077
Application.logger (INFO) F-score of class [109] vs all: 0.857143

Application.logger (INFO) Precision of class [110] vs all: 1
Application.logger (INFO) Recall of class [110] vs all: 1
Application.logger (INFO) F-score of class [110] vs all: 1

Application.logger (INFO) Precision of class [111] vs all: 0.888889
Application.logger (INFO) Recall of class [111] vs all: 1
Application.logger (INFO) F-score of class [111] vs all: 0.941176

Application.logger (INFO) Precision of class [112] vs all: 1
Application.logger (INFO) Recall of class [112] vs all: 1
Application.logger (INFO) F-score of class [112] vs all: 1

Application.logger (INFO) Precision of class [113] vs all: 1
Application.logger (INFO) Recall of class [113] vs all: 1
Application.logger (INFO) F-score of class [113] vs all: 1

Application.logger (INFO) Precision of class [114] vs all: 0.916667
Application.logger (INFO) Recall of class [114] vs all: 1
Application.logger (INFO) F-score of class [114] vs all: 0.956522

Application.logger (INFO) Precision of class [115] vs all: 0.833333
Application.logger (INFO) Recall of class [115] vs all: 1
Application.logger (INFO) F-score of class [115] vs all: 0.909091

Application.logger (INFO) Precision of class [116] vs all: 1
Application.logger (INFO) Recall of class [116] vs all: 1
Application.logger (INFO) F-score of class [116] vs all: 1

Application.logger (INFO) Precision of class [117] vs all: 0.941176
Application.logger (INFO) Recall of class [117] vs all: 1
Application.logger (INFO) F-score of class [117] vs all: 0.969697

Application.logger (INFO) Precision of class [118] vs all: 1
Application.logger (INFO) Recall of class [118] vs all: 1
Application.logger (INFO) F-score of class [118] vs all: 1

Application.logger (INFO) Precision of class [119] vs all: 1
Application.logger (INFO) Recall of class [119] vs all: 1
Application.logger (INFO) F-score of class [119] vs all: 1

Application.logger (INFO) Precision of class [120] vs all: 1
Application.logger (INFO) Recall of class [120] vs all: 1
Application.logger (INFO) F-score of class [120] vs all: 1

Application.logger (INFO) Precision of class [121] vs all: 1
Application.logger (INFO) Recall of class [121] vs all: 1
Application.logger (INFO) F-score of class [121] vs all: 1

Application.logger (INFO) Precision of class [122] vs all: 1
Application.logger (INFO) Recall of class [122] vs all: 1
Application.logger (INFO) F-score of class [122] vs all: 1

Application.logger (INFO) Precision of class [123] vs all: 1

Application.logger (INFO) Recall of class [123] vs all: 1
Application.logger (INFO) F-score of class [123] vs all: 1

Application.logger (INFO) Precision of class [124] vs all: 1
Application.logger (INFO) Recall of class [124] vs all: 1
Application.logger (INFO) F-score of class [124] vs all: 1

Application.logger (INFO) Precision of class [125] vs all: 1
Application.logger (INFO) Recall of class [125] vs all: 1
Application.logger (INFO) F-score of class [125] vs all: 1

Application.logger (INFO) Precision of class [126] vs all: 1
Application.logger (INFO) Recall of class [126] vs all: 1
Application.logger (INFO) F-score of class [126] vs all: 1

Application.logger (INFO) Precision of class [127] vs all: 1
Application.logger (INFO) Recall of class [127] vs all: 1
Application.logger (INFO) F-score of class [127] vs all: 1

Application.logger (INFO) Precision of class [128] vs all: 1
Application.logger (INFO) Recall of class [128] vs all: 1
Application.logger (INFO) F-score of class [128] vs all: 1

Application.logger (INFO) Precision of class [129] vs all: 0.916667
Application.logger (INFO) Recall of class [129] vs all: 1
Application.logger (INFO) F-score of class [129] vs all: 0.956522

Application.logger (INFO) Precision of class [130] vs all: 1
Application.logger (INFO) Recall of class [130] vs all: 1
Application.logger (INFO) F-score of class [130] vs all: 1

Application.logger (INFO) Precision of class [131] vs all: 0.909091
Application.logger (INFO) Recall of class [131] vs all: 1
Application.logger (INFO) F-score of class [131] vs all: 0.952381

Application.logger (INFO) Precision of class [132] vs all: 1
Application.logger (INFO) Recall of class [132] vs all: 0.916667
Application.logger (INFO) F-score of class [132] vs all: 0.956522

Application.logger (INFO) Precision of class [133] vs all: 1
Application.logger (INFO) Recall of class [133] vs all: 1
Application.logger (INFO) F-score of class [133] vs all: 1

Application.logger (INFO) Precision of class [134] vs all: 1
Application.logger (INFO) Recall of class [134] vs all: 0.916667
Application.logger (INFO) F-score of class [134] vs all: 0.956522

Application.logger (INFO) Precision of class [135] vs all: 1
Application.logger (INFO) Recall of class [135] vs all: 0.857143

Application.logger (INFO) F-score of class [135] vs all: 0.923077

Application.logger (INFO) Precision of class [136] vs all: 0.75

Application.logger (INFO) Recall of class [136] vs all: 1

Application.logger (INFO) F-score of class [136] vs all: 0.857143

Application.logger (INFO) Precision of class [137] vs all: 1

Application.logger (INFO) Recall of class [137] vs all: 0.7

Application.logger (INFO) F-score of class [137] vs all: 0.823529

Application.logger (INFO) Precision of class [138] vs all: 1

Application.logger (INFO) Recall of class [138] vs all: 0.714286

Application.logger (INFO) F-score of class [138] vs all: 0.833333

Application.logger (INFO) Precision of class [139] vs all: 0.461538

Application.logger (INFO) Recall of class [139] vs all: 0.545455

Application.logger (INFO) F-score of class [139] vs all: 0.5

Application.logger (INFO) Precision of class [140] vs all: 0.714286

Application.logger (INFO) Recall of class [140] vs all: 0.833333

Application.logger (INFO) F-score of class [140] vs all: 0.769231

Application.logger (INFO) Precision of class [141] vs all: 0.578947

Application.logger (INFO) Recall of class [141] vs all: 0.846154

Application.logger (INFO) F-score of class [141] vs all: 0.6875

Application.logger (INFO) Precision of class [142] vs all: 1

Application.logger (INFO) Recall of class [142] vs all: 0.666667

Application.logger (INFO) F-score of class [142] vs all: 0.8

Application.logger (INFO) Precision of class [143] vs all: 1

Application.logger (INFO) Recall of class [143] vs all: 0.8

Application.logger (INFO) F-score of class [143] vs all: 0.888889

Application.logger (INFO) Precision of class [144] vs all: 0.727273

Application.logger (INFO) Recall of class [144] vs all: 0.615385

Application.logger (INFO) F-score of class [144] vs all: 0.666667

Application.logger (INFO) Precision of class [145] vs all: 0.6

Application.logger (INFO) Recall of class [145] vs all: 0.75

Application.logger (INFO) F-score of class [145] vs all: 0.666667

Application.logger (INFO) Precision of class [146] vs all: 0.857143

Application.logger (INFO) Recall of class [146] vs all: 0.75

Application.logger (INFO) F-score of class [146] vs all: 0.8

Application.logger (INFO) Precision of class [147] vs all: 1

Application.logger (INFO) Recall of class [147] vs all: 1

Application.logger (INFO) F-score of class [147] vs all: 1

Application.logger (INFO) Precision of class [148] vs all: 0.714286
Application.logger (INFO) Recall of class [148] vs all: 0.714286
Application.logger (INFO) F-score of class [148] vs all: 0.714286

Application.logger (INFO) Precision of class [149] vs all: 0.75
Application.logger (INFO) Recall of class [149] vs all: 0.375
Application.logger (INFO) F-score of class [149] vs all: 0.5

Application.logger (INFO) Precision of class [150] vs all: 0.6875
Application.logger (INFO) Recall of class [150] vs all: 0.916667
Application.logger (INFO) F-score of class [150] vs all: 0.785714

Application.logger (INFO) Precision of class [151] vs all: 1
Application.logger (INFO) Recall of class [151] vs all: 0.857143
Application.logger (INFO) F-score of class [151] vs all: 0.923077

Application.logger (INFO) Precision of class [152] vs all: 1
Application.logger (INFO) Recall of class [152] vs all: 0.583333
Application.logger (INFO) F-score of class [152] vs all: 0.736842

Application.logger (INFO) Precision of class [153] vs all: 0.857143
Application.logger (INFO) Recall of class [153] vs all: 0.666667
Application.logger (INFO) F-score of class [153] vs all: 0.75

Application.logger (INFO) Precision of class [154] vs all: 1
Application.logger (INFO) Recall of class [154] vs all: 0.857143
Application.logger (INFO) F-score of class [154] vs all: 0.923077

Application.logger (INFO) Precision of class [155] vs all: 1
Application.logger (INFO) Recall of class [155] vs all: 1
Application.logger (INFO) F-score of class [155] vs all: 1

Application.logger (INFO) Precision of class [156] vs all: 1
Application.logger (INFO) Recall of class [156] vs all: 1
Application.logger (INFO) F-score of class [156] vs all: 1

Application.logger (INFO) Precision of class [157] vs all: 1
Application.logger (INFO) Recall of class [157] vs all: 1
Application.logger (INFO) F-score of class [157] vs all: 1

Application.logger (INFO) Precision of class [158] vs all: 0.928571
Application.logger (INFO) Recall of class [158] vs all: 0.928571
Application.logger (INFO) F-score of class [158] vs all: 0.928571

Application.logger (INFO) Precision of class [159] vs all: 1
Application.logger (INFO) Recall of class [159] vs all: 0.8
Application.logger (INFO) F-score of class [159] vs all: 0.888889

Application.logger (INFO) Precision of class [160] vs all: 1
Application.logger (INFO) Recall of class [160] vs all: 1
Application.logger (INFO) F-score of class [160] vs all: 1

Application.logger (INFO) Precision of class [161] vs all: 1
Application.logger (INFO) Recall of class [161] vs all: 1
Application.logger (INFO) F-score of class [161] vs all: 1

Application.logger (INFO) Precision of class [162] vs all: 0.888889
Application.logger (INFO) Recall of class [162] vs all: 1
Application.logger (INFO) F-score of class [162] vs all: 0.941176

Application.logger (INFO) Precision of class [163] vs all: 1
Application.logger (INFO) Recall of class [163] vs all: 0.75
Application.logger (INFO) F-score of class [163] vs all: 0.857143

Application.logger (INFO) Precision of class [164] vs all: 0.857143
Application.logger (INFO) Recall of class [164] vs all: 0.75
Application.logger (INFO) F-score of class [164] vs all: 0.8

Application.logger (INFO) Precision of class [165] vs all: 1
Application.logger (INFO) Recall of class [165] vs all: 1
Application.logger (INFO) F-score of class [165] vs all: 1

Application.logger (INFO) Precision of class [166] vs all: 1
Application.logger (INFO) Recall of class [166] vs all: 0.833333
Application.logger (INFO) F-score of class [166] vs all: 0.909091

Application.logger (INFO) Precision of class [167] vs all: 1
Application.logger (INFO) Recall of class [167] vs all: 0.875
Application.logger (INFO) F-score of class [167] vs all: 0.933333

Application.logger (INFO) Precision of class [168] vs all: 0.9
Application.logger (INFO) Recall of class [168] vs all: 1
Application.logger (INFO) F-score of class [168] vs all: 0.947368

Application.logger (INFO) Precision of class [169] vs all: 0.888889
Application.logger (INFO) Recall of class [169] vs all: 1
Application.logger (INFO) F-score of class [169] vs all: 0.941176

Application.logger (INFO) Precision of class [170] vs all: 0.916667
Application.logger (INFO) Recall of class [170] vs all: 1
Application.logger (INFO) F-score of class [170] vs all: 0.956522

Application.logger (INFO) Precision of class [171] vs all: 0.636364
Application.logger (INFO) Recall of class [171] vs all: 1
Application.logger (INFO) F-score of class [171] vs all: 0.777778

Application.logger (INFO) Precision of class [172] vs all: 1

Application.logger (INFO) Recall of class [172] vs all: 1
Application.logger (INFO) F-score of class [172] vs all: 1

Application.logger (INFO) Precision of class [173] vs all: 1
Application.logger (INFO) Recall of class [173] vs all: 1
Application.logger (INFO) F-score of class [173] vs all: 1

Application.logger (INFO) Precision of class [174] vs all: 1
Application.logger (INFO) Recall of class [174] vs all: 0.5
Application.logger (INFO) F-score of class [174] vs all: 0.666667

Application.logger (INFO) Precision of class [175] vs all: 1
Application.logger (INFO) Recall of class [175] vs all: 1
Application.logger (INFO) F-score of class [175] vs all: 1

Application.logger (INFO) Precision of class [176] vs all: 0.888889
Application.logger (INFO) Recall of class [176] vs all: 1
Application.logger (INFO) F-score of class [176] vs all: 0.941176

Application.logger (INFO) Precision of class [177] vs all: 0.833333
Application.logger (INFO) Recall of class [177] vs all: 0.833333
Application.logger (INFO) F-score of class [177] vs all: 0.833333

Application.logger (INFO) Precision of class [178] vs all: 0.9
Application.logger (INFO) Recall of class [178] vs all: 1
Application.logger (INFO) F-score of class [178] vs all: 0.947368

Application.logger (INFO) Precision of class [179] vs all: 0
Application.logger (INFO) Recall of class [179] vs all: 0
Application.logger (INFO) F-score of class [179] vs all: 0

Application.logger (INFO) Precision of class [180] vs all: 0.888889
Application.logger (INFO) Recall of class [180] vs all: 1
Application.logger (INFO) F-score of class [180] vs all: 0.941176

Application.logger (INFO) Precision of class [181] vs all: 1
Application.logger (INFO) Recall of class [181] vs all: 1
Application.logger (INFO) F-score of class [181] vs all: 1

Application.logger (INFO) Precision of class [182] vs all: 0.666667
Application.logger (INFO) Recall of class [182] vs all: 0.545455
Application.logger (INFO) F-score of class [182] vs all: 0.6

Application.logger (INFO) Precision of class [183] vs all: 0.875
Application.logger (INFO) Recall of class [183] vs all: 0.7
Application.logger (INFO) F-score of class [183] vs all: 0.777778

Application.logger (INFO) Precision of class [184] vs all: 0.705882
Application.logger (INFO) Recall of class [184] vs all: 0.8

Application.logger (INFO) F-score of class [184] vs all: 0.75

Application.logger (INFO) Precision of class [185] vs all: 1

Application.logger (INFO) Recall of class [185] vs all: 1

Application.logger (INFO) F-score of class [185] vs all: 1

Application.logger (INFO) Precision of class [186] vs all: 0.888889

Application.logger (INFO) Recall of class [186] vs all: 0.941176

Application.logger (INFO) F-score of class [186] vs all: 0.914286

Application.logger (INFO) Precision of class [187] vs all: 1

Application.logger (INFO) Recall of class [187] vs all: 1

Application.logger (INFO) F-score of class [187] vs all: 1

Application.logger (INFO) Precision of class [188] vs all: 1

Application.logger (INFO) Recall of class [188] vs all: 0.947368

Application.logger (INFO) F-score of class [188] vs all: 0.972973

Application.logger (INFO) Precision of class [189] vs all: 0.846154

Application.logger (INFO) Recall of class [189] vs all: 1

Application.logger (INFO) F-score of class [189] vs all: 0.916667

Application.logger (INFO) Precision of class [190] vs all: 0.923077

Application.logger (INFO) Recall of class [190] vs all: 0.923077

Application.logger (INFO) F-score of class [190] vs all: 0.923077

Application.logger (INFO) Precision of class [191] vs all: 1

Application.logger (INFO) Recall of class [191] vs all: 0.923077

Application.logger (INFO) F-score of class [191] vs all: 0.96

Application.logger (INFO) Precision of class [192] vs all: 0.769231

Application.logger (INFO) Recall of class [192] vs all: 1

Application.logger (INFO) F-score of class [192] vs all: 0.869565

Application.logger (INFO) Precision of class [193] vs all: 1

Application.logger (INFO) Recall of class [193] vs all: 0.916667

Application.logger (INFO) F-score of class [193] vs all: 0.956522

Application.logger (INFO) Precision of class [194] vs all: 0.833333

Application.logger (INFO) Recall of class [194] vs all: 0.909091

Application.logger (INFO) F-score of class [194] vs all: 0.869565

Application.logger (INFO) Precision of class [195] vs all: 0.722222

Application.logger (INFO) Recall of class [195] vs all: 1

Application.logger (INFO) F-score of class [195] vs all: 0.83871

Application.logger (INFO) Precision of class [196] vs all: 1

Application.logger (INFO) Recall of class [196] vs all: 0.545455

Application.logger (INFO) F-score of class [196] vs all: 0.705882

Application.logger (INFO) Precision of class [197] vs all: 0.785714

Application.logger (INFO) Recall of class [197] vs all: 0.846154

Application.logger (INFO) F-score of class [197] vs all: 0.814815

Application.logger (INFO) Precision of class [198] vs all: 0.666667

Application.logger (INFO) Recall of class [198] vs all: 0.571429

Application.logger (INFO) F-score of class [198] vs all: 0.615385

Application.logger (INFO) Precision of class [199] vs all: 0.8

Application.logger (INFO) Recall of class [199] vs all: 0.666667

Application.logger (INFO) F-score of class [199] vs all: 0.727273

Application.logger (INFO) Precision of class [200] vs all: 0.888889

Application.logger (INFO) Recall of class [200] vs all: 1

Application.logger (INFO) F-score of class [200] vs all: 0.941176

Application.logger (INFO) Precision of class [201] vs all: 0.625

Application.logger (INFO) Recall of class [201] vs all: 0.909091

Application.logger (INFO) F-score of class [201] vs all: 0.740741

Application.logger (INFO) Precision of class [202] vs all: 0.705882

Application.logger (INFO) Recall of class [202] vs all: 0.8

Application.logger (INFO) F-score of class [202] vs all: 0.75

Application.logger (INFO) Precision of class [203] vs all: 0.571429

Application.logger (INFO) Recall of class [203] vs all: 0.8

Application.logger (INFO) F-score of class [203] vs all: 0.666667

Application.logger (INFO) Precision of class [204] vs all: 0.6

Application.logger (INFO) Recall of class [204] vs all: 0.5

Application.logger (INFO) F-score of class [204] vs all: 0.545455

Application.logger (INFO) Precision of class [205] vs all: 0.875

Application.logger (INFO) Recall of class [205] vs all: 0.875

Application.logger (INFO) F-score of class [205] vs all: 0.875

Application.logger (INFO) Precision of class [206] vs all: 1

Application.logger (INFO) Recall of class [206] vs all: 1

Application.logger (INFO) F-score of class [206] vs all: 1

Application.logger (INFO) Precision of class [207] vs all: 0.9375

Application.logger (INFO) Recall of class [207] vs all: 1

Application.logger (INFO) F-score of class [207] vs all: 0.967742

Application.logger (INFO) Precision of class [208] vs all: 1

Application.logger (INFO) Recall of class [208] vs all: 1

Application.logger (INFO) F-score of class [208] vs all: 1

Application.logger (INFO) Precision of class [209] vs all: 0.571429
Application.logger (INFO) Recall of class [209] vs all: 0.5
Application.logger (INFO) F-score of class [209] vs all: 0.533333

Application.logger (INFO) Precision of class [210] vs all: 0.625
Application.logger (INFO) Recall of class [210] vs all: 0.714286
Application.logger (INFO) F-score of class [210] vs all: 0.666667

Application.logger (INFO) Precision of class [211] vs all: 0.666667
Application.logger (INFO) Recall of class [211] vs all: 0.727273
Application.logger (INFO) F-score of class [211] vs all: 0.695652

Application.logger (INFO) Precision of class [212] vs all: 0.769231
Application.logger (INFO) Recall of class [212] vs all: 0.769231
Application.logger (INFO) F-score of class [212] vs all: 0.769231

Application.logger (INFO) Precision of class [213] vs all: 0.769231
Application.logger (INFO) Recall of class [213] vs all: 0.769231
Application.logger (INFO) F-score of class [213] vs all: 0.769231

Application.logger (INFO) Precision of class [214] vs all: 0.909091
Application.logger (INFO) Recall of class [214] vs all: 0.769231
Application.logger (INFO) F-score of class [214] vs all: 0.833333

Application.logger (INFO) Precision of class [215] vs all: 1
Application.logger (INFO) Recall of class [215] vs all: 1
Application.logger (INFO) F-score of class [215] vs all: 1

Application.logger (INFO) Precision of class [216] vs all: 0.75
Application.logger (INFO) Recall of class [216] vs all: 1
Application.logger (INFO) F-score of class [216] vs all: 0.857143

Application.logger (INFO) Precision of class [217] vs all: 1
Application.logger (INFO) Recall of class [217] vs all: 0.583333
Application.logger (INFO) F-score of class [217] vs all: 0.736842

Application.logger (INFO) Global performance, Kappa index: 0.874038

ΠΑΡΑΡΤΗΜΑ II

Μετατροπή Ψηφιακών Τιμών σε Τιμές Φαινόμενης Ανακλαστικότητας για τον Δέκτη OLI του Δορυφόρου Landsat 8 σε γλώσσα C

```

1  /* Landsat_8 Dns_to_TOA_Refl v1.0 Ver.: OK (c) 07/14 - Dimitrios Bliziotis */
2  /*
3  Name:                DN2TOAREFL
4  Version:             1.0
5  Copyright:           DimBliz @ 2014
6  Author:              Dimitrios Bliziotis
7  Date:                12/07/14 18:56
8  Modified:
9  Description:  Lansat_8_Radiometric
10 Usage:      $DN2TOAREFL L8_Image_in rows cols bands Sun_Elevation Output_name
11
12 Copyright (C) July 2014 DimBliz <dbliziot@gmail.com>
13
14 This program is free software: you can redistribute it and/or modify
15 it under the terms of the GNU General Public License as published by
16 the Free Software Foundation, either version 3 of the License, or
17 (at your option) any later version.
18
19 This program is distributed in the hope that it will be useful,
20 but WITHOUT ANY WARRANTY; without even the implied warranty of
21 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
22 GNU General Public License for more details.
23
24 You should have received a copy of the GNU General Public License
25 along with this program. If not, see <http://www.gnu.org/licenses/>.
26 */
27
28 #include <stdio.h>
29 #include <stdlib.h>
30 #include <string.h>
31 #include <math.h>
32
33 #define IMAGE_IN(x,y,z) (image_in [(((x)*bands)+(y))*cols+(z)])
34 #define IMAGE_OUT(x,y,z) (image_out [(((x)*bands)+(y))*cols+(z)])
35
36
37 int main(int argc, const char * argv[])
38 {
39     FILE *fpin,*fpout;
40     unsigned short int *image_in;
41     float *image_out;
42     int rows;
43     int cols;

```

```

44  int bands;
45  const double REFLECTANCE_MULT_BAND= 2.0000E-05 ;
46  const double REFLECTANCE_ADD_BAND= -0.100000 ;
47  double SUN_ELEVATION;
48  int i,j,k;
49
50  rows= atoi(argv[2]);
51  cols= atoi(argv[3]);
52  bands= atoi(argv[4]);
53  SUN_ELEVATION= atof(argv[5]);
54
55  // Open Input Image
56  if ((fpin=fopen(argv[1],"rb"))==NULL){
57      printf("Error: Bad InputImageFile: <%s>\n",argv[1]);
58      exit(1);
59  }
60
61  // Alocate Memory for Input Image
62  if ((image_in=malloc(rows*cols*bands*sizeof(unsigned short int)))==NULL){
63      printf("Error: Not enough memory available\n");
64      exit(1);
65  }
66
67  // Read Input Image
68  if (fread(image_in,sizeof(unsigned short int),rows*cols*bands,fpin)!=rows*cols*bands){
69      printf("Error: Input file size does not match image dimensions\n");
70      exit(1);
71  }
72
73  // Create Output File
74  if ((fpout=fopen(argv[6],"wb"))==NULL){
75      exit(1);
76  }
77
78  // Alocate Memory for Output Image
79  if ((image_out=malloc(rows*cols*bands*sizeof(float)))==NULL){
80      printf("Error: Not enough memory available\n");
81      exit(1);
82  }
83
84  // Digital Numbers (DNs) to TOA Reflectance
85  for (i=0; i<rows; i++) {
86      for (j=0; j<bands; j++) {
87          for (k=0;k<cols;k++) {
88
89              IMAGE_OUT(i,j,k)=
90              ((REFLECTANCE_MULT_BAND*IMAGE_IN(i,j,k))+REFLECTANCE_ADD_BAND)/(sin(SUN_ELEVATION*atan(1)*4/180));
91
92              if (IMAGE_OUT(i, j, k)<0)

```

```
93         IMAGE_OUT(i, j, k)=0;
94
95     }
96
97 }
98 }
99
100 // Write new Image
101 fwrite(image_out,sizeof(float),rows*cols*bands,fpout);
102
103 // Free memory
104 free(image_in);
105 free(image_out);
106
107 return 0;
108
109 }
```

Μετατροπή Ψηφιακών Τιμών σε Τιμές Θερμοκρασίας για το Κανάλι 10 (TIRS) του Δορυφόρου Landsat 8 σε γλώσσα C

```

1  /* Landsat_8 Dns_to_K_B10 v1.0 Ver.: OK (c) 07/14 - Dimitrios Bliziotis */
2  /*
3  Name:                DN2Kelvin
4  Version:             1.0
5  Copyright:           DimBliz @ 2014
6  Author:              Dimitrios Bliziotis
7  Date:                12/07/14 18:56
8  Modified:
9  Description:  Lansat_8_Radiometric
10 Usage:       $DN2Kelvin L8_Band_10 rows cols Output_name
11
12 Copyright (C) July 2014 DimBliz <dbliziot@gmail.com>
13
14 This program is free software: you can redistribute it and/or modify
15 it under the terms of the GNU General Public License as published by
16 the Free Software Foundation, either version 3 of the License, or
17 (at your option) any later version.
18
19 This program is distributed in the hope that it will be useful,
20 but WITHOUT ANY WARRANTY; without even the implied warranty of
21 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
22 GNU General Public License for more details.
23
24 You should have received a copy of the GNU General Public License
25 along with this program. If not, see <http://www.gnu.org/licenses/>.
26 */
27
28
29 #include <stdio.h>
30 #include <stdlib.h>
31 #include <string.h>
32 #include <math.h>
33
34 #define IMAGE_IN(x,y,z) (image_in [(((x)*bands)+(y))*cols+(z)])
35 #define IMAGE_OUT(x,y,z) (image_out [(((x)*bands)+(y))*cols+(z)])
36
37
38 int main(int argc, const char * argv[])
39 {
40     FILE *fpin,*fpout;
41     unsigned short int *image_in;
42     float *image_out;
43     int rows;
44     int cols;
45     int bands=1;
46     const double RADIANCE_MULT_BAND= 3.3420E-04 ;
47     const double RADIANCE_ADD_BAND= 0.10000 ;

```

```

48  const double K1_10= 774.89 ;
49  const double K2_10= 1321.08 ;
50  int i,j,k;
51
52  rows= atoi(argv[2]);
53  cols= atoi(argv[3]);
54
55
56  // Open Input Image
57  if ((fpin=fopen(argv[1],"rb"))==NULL){
58      printf("Error: Bad InputImageFile: <%s>\n",argv[1]);
59      exit(1);
60  }
61
62  // Allocate Memory for Input Image
63  if ((image_in=malloc(rows*cols*bands*sizeof(unsigned short int)))==NULL){
64      printf("Error: Not enough memory available\n");
65      exit(1);
66  }
67
68  // Read Input Image
69  if (fread(image_in,sizeof(unsigned short int),rows*cols*bands,fpin)!=rows*cols*bands){
70      printf("Error: Input file size does not match image dimensions\n");
71      exit(1);
72  }
73
74
75  // Create Output File
76  if ((fpout=fopen(argv[4],"wb"))==NULL){
77      exit(1);
78  }
79
80
81  // Allocate Memory for Output Image
82  if ((image_out=malloc(rows*cols*bands*sizeof(float)))==NULL){
83      printf("Error: Not enough memory available\n");
84      exit(1);
85  }
86
87  // Digital Numbers (DNs) to Kelvin Temperature
88  for (i=0; i<rows; i++) {
89      for (j=0; j<bands; j++) {
90          for (k=0; k<cols; k++) {
91
92              IMAGE_OUT(i,j,k)= ((RADIANCE_MULT_BAND*IMAGE_IN(i,j,k))+RADIANCE_ADD_BAND);
93              IMAGE_OUT(i,j,k)= K2_10/(log((K1_10/IMAGE_OUT(i, j, k))+1));
94          }
95      }
96  }

```



```
97     }
98
99     // Write new Image
100    fwrite(image_out,sizeof(float),rows*cols*bands,fpout);
101
102    // Free memory
103    free(image_in);
104    free(image_out);
105
106    return 0;
107
108 }
```

Μετατροπή Ψηφιακών Τιμών σε Τιμές Θερμοκρασίας για το Κανάλι 11 (TIRS) του Δορυφόρου Landsat 8 σε γλώσσα C

```

1  /* Landsat_8 Dns_to_K_B11 v1.0 Ver.: OK (c) 07/14 - Dimitrios Bliziotis */
2  /*
3  Name:                DN2Kelvin
4  Version:             1.0
5  Copyright:           DimBliz @ 2014
6  Author:              Dimitrios Bliziotis
7  Date:                12/07/14 18:56
8  Modified:
9  Description:         Lansat_8_Radiometric
10 Usage:               $DN2Kelvin L8_Band_11 rows cols Output_name
11
12 Copyright (C) July 2014 DimBliz <dbliziot@gmail.com>
13
14 This program is free software: you can redistribute it and/or modify
15 it under the terms of the GNU General Public License as published by
16 the Free Software Foundation, either version 3 of the License, or
17 (at your option) any later version.
18
19 This program is distributed in the hope that it will be useful,
20 but WITHOUT ANY WARRANTY; without even the implied warranty of
21 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
22 GNU General Public License for more details.
23
24 You should have received a copy of the GNU General Public License
25 along with this program. If not, see <http://www.gnu.org/licenses/>.
26 */
27
28 #include <stdio.h>
29 #include <stdlib.h>
30 #include <string.h>
31 #include <math.h>
32
33 #define IMAGE_IN(x,y,z) (image_in [(((x)*bands)+(y))*cols+(z)])
34 #define IMAGE_OUT(x,y,z) (image_out [(((x)*bands)+(y))*cols+(z)])
35
36
37 int main(int argc, const char * argv[])
38 {
39     FILE *fpin,*fpout;
40     unsigned short int *image_in;
41     float *image_out;
42     int rows;
43     int cols;
44     int bands=1;
45     const double RADIANCE_MULT_BAND= 3.3420E-04 ;
46     const double RADIANCE_ADD_BAND= 0.10000 ;
47     const double K1_11= 480.89 ;

```

```

48  const double K2_11= 1201.14 ;
49  int i,j,k;
50
51  rows= atoi(argv[2]);
52  cols= atoi(argv[3]);
53
54
55  // Open Input Image
56  if ((fpin=fopen(argv[1],"rb"))==NULL){
57      printf("Error: Bad InputImageFile: <%s>\n",argv[1]);
58      exit(1);
59  }
60
61  // Alocate Memory for Input Image
62  if ((image_in=malloc(rows*cols*bands*sizeof(unsigned short int)))==NULL){
63      printf("Error: Not enough memory available\n");
64      exit(1);
65  }
66
67  // Read Input Image
68  if (fread(image_in,sizeof(unsigned short int),rows*cols*bands,fpin)!=rows*cols*bands){
69      printf("Error: Input file size does not match image dimensions\n");
70      exit(1);
71  }
72
73
74  // Create Output File
75  if ((fpout=fopen(argv[4],"wb"))==NULL){
76      exit(1);
77  }
78
79
80  // Alocate Memory for Output Image
81  if ((image_out=malloc(rows*cols*bands*sizeof(float)))==NULL){
82      printf("Error: Not enough memory available\n");
83      exit(1);
84  }
85
86  // Digital Numbers (DNs) to Kelvin Temperature
87  for (i=0; i<rows; i++) {
88      for (j=0; j<bands; j++) {
89          for (k=0;k<cols;k++) {
90
91              IMAGE_OUT(i,j,k)= ((RADIANCE_MULT_BAND*IMAGE_IN(i,j,k))+RADIANCE_ADD_BAND);
92              IMAGE_OUT(i,j,k)= K2_11/(log((K1_11/IMAGE_OUT(i, j, k))+1));
93              //printf("%f \t", IMAGE_OUT(i, j, k));
94          }
95      }
96  }

```

```
97     }
98
99     // Write new Image
100    fwrite(image_out,sizeof(float),rows*cols*bands,fpout);
101
102    // Free memory
103    free(image_in);
104    free(image_out);
105
106    return 0;
107
108 }
```

Αλγόριθμος Ανίχνευσης και Αφαίρεσης των Συννέφων και των Σκιών τους σε Δορυφορικές Εικόνες Landsat 8 σε γλώσσα C

```

1  /* Landsat_8 Cloud-Shadow Detection and Removal v1.0 Ver.: OK (c) 07/14 - Dimitrios Bliziotis */
2  /*
3  Name:                Cloud_Shadow_Process
4  Version:             1.0
5  Copyright:          DimBliz @ 2014
6  Author:             Dimitrios Bliziotis
7  Date:               12/07/14 18:56
8  Modified:
9  Description:  Lansat_8_Cloud-Shadow Detection and Removal
10 Usage:    $Cloud_Shadow_Process Band_clouds Band_1_clouds Band_3_clouds Band_6_clouds
11 Band_without_clouds Band_1_without_clouds Band_3_without_clouds Band_6_without_clouds
12 Band_cloud_free rows cols
13
14 Copyright (C) July 2014 DimBliz <dbliziot@gmail.com>
15
16 This program is free software: you can redistribute it and/or modify
17 it under the terms of the GNU General Public License as published by
18 the Free Software Foundation, either version 3 of the License, or
19 (at your option) any later version.
20
21 This program is distributed in the hope that it will be useful,
22 but WITHOUT ANY WARRANTY; without even the implied warranty of
23 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
24 GNU General Public License for more details.
25
26 You should have received a copy of the GNU General Public License
27 along with this program. If not, see <http://www.gnu.org/licenses/>.
28 */
29
30 #include <stdio.h>
31 #include <stdlib.h>
32 #include <string.h>
33 #include <math.h>
34
35 #define BAND_CLOUD(x,y,z) (band_cloud [(((x)*bands)+(y))*cols+(z)])
36 #define BAND_1_CLOUD(x,y,z) (band_1_cloud [(((x)*bands)+(y))*cols+(z)])
37 #define BAND_3_CLOUD(x,y,z) (band_3_cloud [(((x)*bands)+(y))*cols+(z)])
38 #define BAND_6_CLOUD(x,y,z) (band_6_cloud [(((x)*bands)+(y))*cols+(z)])
39 #define BAND_NO_CLOUD(x,y,z) (band_no_cloud [(((x)*bands)+(y))*cols+(z)])
40 #define BAND_1_NO_CLOUD(x,y,z) (band_1_no_cloud [(((x)*bands)+(y))*cols+(z)])
41 #define BAND_3_NO_CLOUD(x,y,z) (band_3_no_cloud [(((x)*bands)+(y))*cols+(z)])
42 #define BAND_6_NO_CLOUD(x,y,z) (band_6_no_cloud [(((x)*bands)+(y))*cols+(z)])
43 #define AERO_GREEN_CLOUD(x,y,z) (aero_green_cloud [(((x)*bands)+(y))*cols+(z)])
44 #define AERO_GREEN_NO_CLOUD(x,y,z) (aero_green_no_cloud [(((x)*bands)+(y))*cols+(z)])
45 #define MASK(x,y,z) (mask [(((x)*bands)+(y))*cols+(z)])
46 #define IMAGE_OUT(x,y,z) (image_out [(((x)*bands)+(y))*cols+(z)])
47 #define sqr(x) ((x)*(x))

```

```

48 int main(int argc, const char * argv[])
49 {
50     FILE *fp1,*fp2,*fp3,*fp4,*fp5,*fp6,*fp7,*fp8,*fpout;
51     float *band_cloud,*band_1_cloud,*band_3_cloud,*band_6_cloud, *band_no_cloud,
52     *band_1_no_cloud,*band_3_no_cloud,*band_6_no_cloud;
53     float *image_out;
54     float *aero_green_cloud, *aero_green_no_cloud;
55     int rows;
56     int cols;
57     int bands;
58     int i,j,k,l;
59     unsigned char *mask;
60
61     rows= atoi(argv[10]);
62     cols= atoi(argv[11]);
63     bands= 1;
64
65     // Reading necessary input for cloudy image
66
67     // 1
68
69     // Open Input Band
70     if ((fp1=fopen(argv[1],"rb"))==NULL){
71         printf("Error: Bad InputImageFile: <%s>\n",argv[1]);
72         exit(1);
73     }
74
75     // Allocate Memory for Input Band
76     if ((band_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
77         printf("Error: Not enough memory available\n");
78         exit(1);
79     }
80
81     // Read Input Band
82     if (fread(band_cloud,sizeof(float),rows*cols*bands,fp1)!=rows*cols*bands){
83         printf("Error: Input file size does not match image dimensions\n");
84         exit(1);
85     }
86
87     // 2
88
89     // Open Input Band 1 (Coastal Aerosol)
90     if ((fp2=fopen(argv[2],"rb"))==NULL){
91         printf("Error: Bad InputImageFile: <%s>\n",argv[2]);
92         exit(1);
93     }
94
95     // Allocate Memory for Input Band 1 (Coastal Aerosol)
96     if ((band_1_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){

```

```

97     printf("Error: Not enough memory available\n");
98     exit(1);
99 }
100
101 // Read Input Band 1 (Coastal Aerosol)
102 if (fread(band_1_cloud,sizeof(float),rows*cols*bands,fp2)!=rows*cols*bands){
103     printf("Error: Input file size does not match image dimensions\n");
104     exit(1);
105 }
106
107 // 3
108
109 // Open Input Band 3 (Green)
110 if ((fp3=fopen(argv[3],"rb"))==NULL){
111     printf("Error: Bad InputImageFile: <%s>\n",argv[3]);
112     exit(1);
113 }
114
115 // Allocate Memory for Input Band 3 (Green)
116 if ((band_3_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
117     printf("Error: Not enough memory available\n");
118     exit(1);
119 }
120
121 // Read Input Band 3 (Green)
122 if (fread(band_3_cloud,sizeof(float),rows*cols*bands,fp3)!=rows*cols*bands){
123     printf("Error: Input file size does not match image dimensions\n");
124     exit(1);
125 }
126
127 // 4
128
129 // Open Input Band 6 (SWIR)
130 if ((fp4=fopen(argv[4],"rb"))==NULL){
131     printf("Error: Bad InputImageFile: <%s>\n",argv[4]);
132     exit(1);
133 }
134
135 // Allocate Memory for Input Band 6 (SWIR)
136 if ((band_6_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
137     printf("Error: Not enough memory available\n");
138     exit(1);
139 }
140
141 // Read Input Band 6 (SWIR)
142 if (fread(band_6_cloud,sizeof(float),rows*cols*bands,fp4)!=rows*cols*bands){
143     printf("Error: Input file size does not match image dimensions\n");
144     exit(1);
145 }

```

```

146
147 // Reading necessary input for cloud free image
148
149 // 1
150
151 // Open Input Band
152 if ((fp5=fopen(argv[5],"rb"))==NULL){
153     printf("Error: Bad InputImageFile: <%s>\n",argv[5]);
154     exit(1);
155 }
156
157 // Alocate Memory for Input Band
158 if ((band_no_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
159     printf("Error: Not enough memory available\n");
160     exit(1);
161 }
162
163 // Read Input Band
164 if (fread(band_no_cloud,sizeof(float),rows*cols*bands,fp5)!=rows*cols*bands){
165     printf("Error: Input file size does not match image dimensions\n");
166     exit(1);
167 }
168
169 // 2
170
171 // Open Input Band 1 (Coastal Aerosol)
172 if ((fp6=fopen(argv[6],"rb"))==NULL){
173     printf("Error: Bad InputImageFile: <%s>\n",argv[6]);
174     exit(1);
175 }
176
177 // Alocate Memory for Input Band 1 (Coastal Aerosol)
178 if ((band_1_no_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
179     printf("Error: Not enough memory available\n");
180     exit(1);
181 }
182
183 // Read Input Band 1 (Coastal Aerosol)
184 if (fread(band_1_no_cloud,sizeof(float),rows*cols*bands,fp6)!=rows*cols*bands){
185     printf("Error: Input file size does not match image dimensions\n");
186     exit(1);
187 }
188
189 // 3
190
191 // Open Input Band 3 (Green)
192 if ((fp7=fopen(argv[7],"rb"))==NULL){
193     printf("Error: Bad InputImageFile: <%s>\n",argv[7]);
194     exit(1);

```



```

195     }
196
197     // Allocate Memory for Input Band 3 (Green)
198     if ((band_3_no_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
199         printf("Error: Not enough memory available\n");
200         exit(1);
201     }
202
203     // Read Input Band 3 (Green)
204     if (fread(band_3_no_cloud,sizeof(float),rows*cols*bands,fp7)!=rows*cols*bands){
205         printf("Error: Input file size does not match image dimensions\n");
206         exit(1);
207     }
208
209
210     // 4
211
212     // Open Input Band 6 (SWIR)
213     if ((fp8=fopen(argv[8],"rb"))==NULL){
214         printf("Error: Bad InputImageFile: <%s>\n",argv[8]);
215         exit(1);
216     }
217
218     // Allocate Memory for Input Band 6 (SWIR)
219     if ((band_6_no_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
220         printf("Error: Not enough memory available\n");
221         exit(1);
222     }
223
224     // Read Input Band 6 (SWIR)
225     if (fread(band_6_no_cloud,sizeof(float),rows*cols*bands,fp8)!=rows*cols*bands){
226         printf("Error: Input file size does not match image dimensions\n");
227         exit(1);
228     }
229
230     // Allocating memory for Aerosol Green Indexes
231
232     // Allocate Memory for Aerosol_Green_Index_cloud
233     if ((aero_green_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
234         printf("Error: Not enough memory available\n");
235         exit(1);
236     }
237
238
239     // Allocate Memory for Aerosol_Green_Index_no_cloud
240     if ((aero_green_no_cloud=malloc(rows*cols*bands*sizeof(float)))==NULL){
241         printf("Error: Not enough memory available\n");
242         exit(1);
243     }

```

```

244
245 // Allocating memory for output band
246
247 // Create Output File
248 if ((fpout=fopen(argv[9],"wb"))==NULL){
249     exit(1);
250 }
251
252
253 // Alocate Memory for Output Image
254 if ((image_out=malloc(rows*cols*bands*sizeof(float)))==NULL){
255     printf("Error: Not enough memory available\n");
256     exit(1);
257 }
258
259 // Initializing output
260 for (i=0; i<rows; i++) {
261     for (j=0; j<bands; j++) {
262         for (k=0; k<cols; k++) {
263             IMAGE_OUT(i, j, k)= BAND_CLOUD(i, j, k);
264         }
265     }
266 }
267
268
269 // Creating Aerosol Green indexes
270
271 // Working on cloudy image
272 for (i=0; i<rows; i++) {
273     for (j=0; j<bands; j++) {
274         for (k=0; k<cols; k++) {
275             if (BAND_1_CLOUD(i, j, k)!=0){
276                 AERO_GREEN_CLOUD(i, j, k)= (1/4)*((BAND_1_CLOUD(i, j, k)-BAND_3_CLOUD(i, j,
277 k))/(BAND_1_CLOUD(i, j, k)));
278                 AERO_GREEN_CLOUD(i, j, k)= BAND_1_CLOUD(i, j, k)+AERO_GREEN_CLOUD(i, j, k);
279             }
280         }
281     }
282 }
283
284 // Working on cloud free image
285 for (i=0; i<rows; i++) {
286     for (j=0; j<bands; j++) {
287         for (k=0; k<cols; k++) {
288             if (BAND_1_NO_CLOUD(i, j, k)!=0){
289                 AERO_GREEN_NO_CLOUD(i, j, k)= (1/4)*((BAND_1_NO_CLOUD(i, j, k)-
290 BAND_3_NO_CLOUD(i, j, k))/(BAND_1_NO_CLOUD(i, j, k)));
291                 AERO_GREEN_NO_CLOUD(i, j, k)= BAND_1_NO_CLOUD(i, j,
292 k)+AERO_GREEN_NO_CLOUD(i, j, k);

```

```

293     }
294     }
295 }
296 }
297
298 // Multi-date Effect Brightness Correction
299
300 // Create Cloud-Shadow Mask For Statistics
301
302
303 // Alocate Memory for Mask Image
304 if ((mask=malloc(rows*cols*bands*sizeof(unsigned char)))==NULL){
305     printf("Error: Not enough memory available\n");
306     exit(1);
307 }
308
309 // Create Cloud_Shadow mask
310 for (i=0; i<rows; i++) {
311     for (j=0; j<bands; j++) {
312         for (k=0; k<cols; k++) {
313             if ((AERO_GREEN_NO_CLOUD(i, j, k)+0.1<AERO_GREEN_CLOUD(i, j, k) &&
314 AERO_GREEN_CLOUD(i, j, k)>=0.2) || (BAND_6_CLOUD(i, j, k)+0.1<BAND_6_NO_CLOUD(i, j, k) &&
315 BAND_6_CLOUD(i, j, k)<0.1))
316
317                 MASK(i,j,k) =1 ;
318             else
319
320                 MASK(i,j,k) =0 ;
321         }
322     }
323 }
324
325 // Compute new image statistics without cloud and shadows for band_cloud
326 int new_row,new_col,N;
327 float sum_cloud,mean_cloud,std_cloud,temp_cloud;
328 new_row=rows;
329 new_col=cols;
330 mean_cloud=0;
331 sum_cloud=0;
332 std_cloud=0;
333 temp_cloud=0;
334 N=new_row*new_col;
335 for (i=0; i<rows; i++) {
336     for (j=0; j<bands; j++) {
337         for (k=0; k<cols; k++) {
338             if (MASK(i,j,k) ==1) {
339
340                 N=N-1;
341

```

```

342     }
343     else {
344         sum_cloud=sum_cloud+BAND_CLOUD(i, j, k);
345     }
346 }
347 }
348 }
349 }
350
351 mean_cloud=sum_cloud/(N);
352 sum_cloud=0;
353 for (i=0; i<rows; i++) {
354     for (j=0; j<bands; j++) {
355         for (k=0; k<cols; k++) {
356             if (MASK(i,j,k) ==0) {
357                 temp_cloud=BAND_CLOUD(i, j, k)-mean_cloud;
358                 sum_cloud+=sqr(temp_cloud);
359             }
360         }
361     }
362 }
363 std_cloud=sum_cloud/(new_row*new_col-1);
364
365 // Compute image statistics for band_no_cloud
366 float sum,mean,std,temp;
367 mean=0;
368 sum=0;
369 std=0;
370 temp=0;
371
372 for (i=0; i<rows; i++) {
373     for (j=0; j<bands; j++) {
374         for (k=0; k<cols; k++) {
375
376             sum=sum+BAND_NO_CLOUD(i, j, k);
377
378         }
379     }
380 }
381
382 mean=sum/(rows*cols);
383 sum=0;
384
385 for (i=0; i<rows; i++) {
386     for (j=0; j<bands; j++) {
387         for (k=0; k<cols; k++) {
388
389             temp=BAND_NO_CLOUD(i, j, k)-mean;
390             sum+=sqr(temp);

```

```

391 }
392 }
393 }
394
395 std=sum/(rows*cols-1);
396
397 // Apply the Multi-date Effect Brightness Correction
398 for (i=0; i<rows; i++) {
399     for (j=0; j<bands; j++) {
400         for (k=0; k<cols; k++) {
401             BAND_NO_CLOUD(i, j, k)=mean_cloud+(BAND_NO_CLOUD(i, j, k)-mean)*std_cloud/std;
402         }
403     }
404 }
405
406 // Cloud removal
407 for (i=0; i<rows; i++) {
408     for (j=0; j<bands; j++) {
409         for (k=0; k<cols; k++) {
410             if ((AERO_GREEN_NO_CLOUD(i, j, k)+0.1<AERO_GREEN_CLOUD(i, j, k) &&
411 AERO_GREEN_CLOUD(i, j, k)>=0.2) || (BAND_6_CLOUD(i, j, k)+0.1<BAND_6_NO_CLOUD(i, j, k) &&
412 BAND_6_CLOUD(i, j, k)<0.1 ))
413             {
414
415
416                 IMAGE_OUT(i, j, k)= BAND_NO_CLOUD(i, j, k);
417
418                 // Also replace in a 4x4 window
419                 for (l=1; l<5; l++){
420                     if (i+l<rows) {
421                         IMAGE_OUT(i+l, j, k)=BAND_NO_CLOUD(i+l, j, k);
422                     }
423
424                     if (k+l<cols) {
425                         IMAGE_OUT(i, j, k+l)=BAND_NO_CLOUD(i, j, k+l);
426                     }
427
428                     if (i-l>=0) {
429                         IMAGE_OUT(i-l, j, k)=BAND_NO_CLOUD(i-l, j, k);
430                     }
431
432                     if (k-l>=0) {
433                         IMAGE_OUT(i, j, k-l)=BAND_NO_CLOUD(i, j, k-l);
434                     }
435                 }
436             }
437         }
438     }
439 }

```

```
440     }
441 }
442
443 // Write new Image
444 fwrite(image_out,sizeof(float),rows*cols*bands,fpout);
445
446 // Free memory
447 free(band_cloud);
448 free(band_1_cloud);
449 free(band_3_cloud);
450 free(band_6_cloud);
451 free(band_no_cloud);
452 free(band_1_no_cloud);
453 free(band_3_no_cloud);
454 free(band_6_no_cloud);
455 free(aero_green_cloud);
456 free(aero_green_no_cloud);
457 free(image_out);
458
459 return 0;
```

Εξαγωγή Αποτελεσμάτων σε Web Gis με Χρήση Ανοιχτών Προτύπων

```

1 <html
2   <head>
3     <title>Demo Web Gis</title>
4     <meta http-equiv='imagetoolbar' content='no'/>
5     <style type="text/css"> v\.* {behavior:url(#default#VML);}
6     html, body { overflow: hidden; padding: 0; height: 100%; width: 100%; font-family: 'Lucida
7 Grande', Geneva, Arial, Verdana, sans-serif; }
8     body { margin: 10px; background: #fff; }
9     h1 { margin: 0; padding: 6px; border: 0; font-size: 20pt; }
10    #header { height: 43px; padding: 0; background-color: #eee; border: 1px solid #888; }
11    #subheader { height: 12px; text-align: right; font-size: 10px; color: #555; }
12    #map { height: 80%; border: 1px solid #888; }
13    .ollImageLoadError { display: none; }
14    .olControlLayerSwitcher .layersDiv { border-radius: 10px 0 0 10px; }
15    #tool { height: 43px; padding: 0; background-color: #eee; border: 1px solid #888; }
16  </style>
17  <script src="http://www.openlayers.org/api/2.12/OpenLayers.js"></script>
18  <script>
19    var map;
20    var mapBounds = new OpenLayers.Bounds( 21.5367424869, 38.7385061233, 23.0695941369,
21 39.9275851667);
22    var mapMinZoom = 1;
23    var mapMaxZoom = 18;
24    var emptyTileURL = "http://www.maptiler.org/img/none.png";
25    OpenLayers.IMAGE_RELOAD_ATTEMPTS = 3;
26
27    function init(){
28      var options = {
29        div: "map",
30        controls: [],
31        projection: "EPSG:900913",
32        displayProjection: new OpenLayers.Projection("EPSG:4326"),
33        numZoomLevels: 20
34      };
35      map = new OpenLayers.Map(options);
36
37
38      var osm = new OpenLayers.Layer.OSM("OpenStreetMap");
39
40      // create TMS Overlay layer
41      var tmsoverlay = new OpenLayers.Layer.TMS("TMS Overlay", "",
42      {
43        serviceVersion: '.',
44        layername: '.',
45        alpha: true,
46        type: 'png',
47        isBaseLayer: false,

```

```

48         getUrl: getUrl
49     });
50     if (OpenLayers.Util.alphaHack() == false) {
51         tmsoverlay.setOpacity(1);
52     }
53
54     var tmsoverlay2 = new OpenLayers.Layer.TMS("TMS Overlay2", "",
55     {
56         serviceVersion: '.',
57         layername: '.',
58         alpha: true,
59         type: 'png',
60         isBaseLayer: false,
61         getUrl: get_URL
62     });
63     if (OpenLayers.Util.alphaHack() == false) {
64         tmsoverlay2.setOpacity(1);
65     }
66
67     map.addLayers([osm, tmsoverlay, tmsoverlay2]);
68
69     var switcherControl = new OpenLayers.Control.LayerSwitcher();
70     map.addControl(switcherControl);
71     switcherControl.maximizeControl();
72
73     map.zoomToExtent(mapBounds.transform(map.displayProjection, map.projection));
74
75     map.addControls([new OpenLayers.Control.PanZoomBar(),
76         new OpenLayers.Control.Navigation(),
77         new OpenLayers.Control.MousePosition(),
78         new OpenLayers.Control.ArgParser(),
79         new OpenLayers.Control.Attribution()]);
80 }
81
82 function getUrl(bounds) {
83     bounds = this.adjustBounds(bounds);
84     var res = this.getServerResolution();
85     var x = Math.round((bounds.left - this.tileOrigin.lon) / (res * this.tileSize.w));
86     var y = Math.round((bounds.bottom - this.tileOrigin.lat) / (res * this.tileSize.h));
87     var z = this.getServerZoom();
88
89     var path = this.serviceVersion + "/thes/" + this.layername + "/" + z + "/" + x + "/" + y + "." +
90 this.type;
91     var url = this.url;
92     if (OpenLayers.Util.isArray(url)) {
93         url = this.selectUrl(path, url);
94     }
95     if (mapBounds.intersectsBounds(bounds) && (z >= mapMinZoom) && (z <= mapMaxZoom))
96 {

```



```

97         return url + path;
98     } else {
99         return emptyTileURL;
100     }
101 }
102
103 function get_URL(bounds) {
104     bounds = this.adjustBounds(bounds);
105     var res = this.getServerResolution();
106     var x = Math.round((bounds.left - this.tileOrigin.lon) / (res * this.tileSize.w));
107     var y = Math.round((bounds.bottom - this.tileOrigin.lat) / (res * this.tileSize.h));
108     var z = this.getServerZoom();
109
110     var path = this.serviceVersion + "/class/" + this.layername + "/" + z + "/" + x + "/" + y + "." +
111 this.type;
112     var url = this.url;
113     if (OpenLayers.Util.isArray(url)) {
114         url = this.selectUrl(path, url);
115     }
116     if (mapBounds.intersectsBounds(bounds) && (z >= mapMinZoom) && (z <= mapMaxZoom))
117 {
118         return url + path;
119     } else {
120         return emptyTileURL;
121     }
122 }
123
124 function getWindowHeight() {
125     if (self.innerHeight) return self.innerHeight;
126     if (document.documentElement && document.documentElement.clientHeight)
127         return document.documentElement.clientHeight;
128     if (document.body) return document.body.clientHeight;
129     return 0;
130 }
131
132 function getWindowWidth() {
133     if (self.innerWidth) return self.innerWidth;
134     if (document.documentElement && document.documentElement.clientWidth)
135         return document.documentElement.clientWidth;
136     if (document.body) return document.body.clientWidth;
137     return 0;
138 }
139
140 function resize() {
141     var map = document.getElementById("map");
142     var header = document.getElementById("header");
143     var subheader = document.getElementById("subheader");
144     map.style.height = (getWindowHeight()-80) + "px";
145     map.style.width = (getWindowWidth()-20) + "px";

```

```

146     header.style.width = (getWindowWidth()-20) + "px";
147     subheader.style.width = (getWindowWidth()-20) + "px";
148     if (map.updateSize) { map.updateSize(); };
149 }
150
151 onresize=function(){ resize(); };
152
153 function toggleControl(element) {
154     if(element.checked)
155     {
156         boxControl = new OpenLayers.Control();
157         OpenLayers.Util.extend(boxControl, {
158             draw: function() {
159                 this.box = new OpenLayers.Handler.RegularPolygon(boxControl,
160                 {"done": this.notice}, {sides:4, irregular:true, persist:true});
161
162                 boxControl.box.activate();
163             },
164             notice: function(geom) {
165                 alert(geom);
166                 boxControl.box.deactivate();
167                 document.getElementById('lineToggle').checked = false;
168             } });
169         map.addControl(boxControl);
170
171     }
172     if(!element.checked)
173     {
174         map.removeControl(boxControl);
175
176     }
177 }
178 </script>
179 </head>
180 <body onload="init()">
181     <div id="header"><h1>Demo Web Gis</h1></div>
182
183     <div id="map"></div>
184     <div id='tool'>
185         <li>
186             <input type="checkbox" name="type" value="line" id="lineToggle"
187             onchange="toggleControl(this);" />
188             <label for="lineToggle">draw box</label>
189         </li>
190     </body>
191 </html>

```

