



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Αγρονόμων και Τοπογράφων Μηχανικών-Μηχανικών Γεωπληροφορικής

Τομέας Τοπογραφίας

Διπλωματική Εργασία

«Τεχνολογίες Συνελικτικών Αυτοκωδικοποιητών σε Θερμικά
και Οπτικά Δεδομένα Βίντεο για την Αυτόματη Ανίχνευση
Πτώσεων»

Καρόλου Ελένη Ειρήνη
Υπεύθυνος Καθηγητής: Δουλάμης Αναστάσιος

Τριμελής Επιτροπή

Δουλάμης Αναστάσιος
Αναπληρωτής Καθηγητής

Δουλάμης Νικόλαος
Αναπληρωτής Καθηγητής

Ιωαννίδης Χαράλαμπος
Καθηγητής



National Technical University of Athens

School of Rural Surveying and Geoinformatics Engineering

Department of Topography

Diploma Thesis

**«Convolutional Autoencoder technologies in Thermal and
Optical Video Data for Automatic Human Fall Detection»**

Eleni Eirini Karolou

Supervisor: Anastasios Doulamis

Three Member Committee

Anastasios Doulamis
Associate Professor

Charalabos Ioannidis
Professor

Nikolaos Doulamis
Associate Professor

Περίληψη

Τα περιστατικά ανθρώπινων πτώσεων αποτελούν σοβαρά ατυχήματα, όπου η ταχεία ανίχνευσή του περιστατικού αυτού, κρίνεται ζωτικής σημασίας για την ασφαλή απαλλαγή του ατόμου από τον μεγάλο κίνδυνο. Για τον σκοπό αυτό, η χρήση μοντέλων βαθιάς μάθησης ως αυτόματων ανιχνευτών, για τα συγκεκριμένα σενάρια, έχει αποδειχθεί αποτελεσματική. Ωστόσο, προκειμένου το εκπαιδευτικό πλαίσιο να λειτουργεί καλύτερα είναι επιτακτική η ανάγκη χρήσης των σωστών μεθόδων αποτύπωσης, όπως οπτικές και θερμικές μέθοδοι. Διερευνάται η χρήση ενός συνελκτικού αυτοκωδικοποιητή, εκπαιδευμένο πάνω σε θερμικά και οπτικά δεδομένα, ως μηχανισμό για την αυτόματη ανίχνευση ανθρώπινων πτώσεων. Επιπλέον, παρουσιάζεται ένα σύνολο δεδομένων που δημιουργήθηκε για να προσομοιώσει τέτοια γεγονότα και χρησιμοποιήθηκε για την εκπαίδευση και τον έλεγχο του αλγορίθμου.

Λέξεις κλειδιά: Human detection, Convolutional Neural Networks (CNN), deep learning, spatiotemporal autoencoder (spatial AE), LSTM.

Abstract

Incidents of human falls are serious accidents, where the rapid detection of this incident is considered vital for the safe release of the individual from great danger. To this end, the use of deep learning models as automated detectors for these scenarios has proven effective. However, in order for the learning framework to work better, it is imperative to use the right imaging methods, such as optical and thermal methods. There is an investigation about the use of a spatiotemporal autoencoder trained on thermal and visual data, as a mechanism for the automatic detection of human falls. In addition, there is a presentation of a set of data created to simulate such events and used to train and test the algorithm.

Key Words: Human detection, Convolutional Neural Networks (CNN), deep learning, spatiotemporal autoencoder (spatial AE), LSTM.

Ευχαριστίες

Ξεκινώντας, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κύριο Δουλάμη Αναστάσιο που με παρότρυνε να ασχοληθώ με το θέμα της αυτόματης ανίχνευσης πτώσεων και με συμβούλεψε κάθε φορά που αναζητούσα τη βοήθειά του. Επίσης, θα ήθελα να ευχαριστήσω από κοινού, τον κύριο Δουλάμη Αναστάσιο και τον κύριο Δουλάμη Νικόλαο, που μέσα από τα μαθήματά τους και τον τρόπο διδασκαλίας τους με ενέπνευσαν να ασχοληθώ με τις προγραμματιστικές τεχνικές και να κατανοήσω σε βάθος τις λειτουργίες τους και την πληθώρα εφαρμογών τους.

Επιπρόσθετα, ευχαριστώ τον κύριο Χαράλαμπο Ιωαννίδη που μέσα από τις διαλέξεις του κατάφερα να κατανοήσω τις αρχές της επιστήμης της Φωτογραμμετρίας.

Ακόμα, ευχαριστώ τον Μπάκαλο Νικόλαο, υποψήφιο διδάκτορα, που επέβλεπε της διπλωματική μου εργασία και κάθε φορά που αντιμετώπιζα κάποιο πρόβλημα με βοηθούσε.

Τέλος, ευχαριστώ από καρδιάς την οικογένεια μου και τους φίλους μου, που με στηρίζουν καθημερινά σε όλα τα κομμάτια της ζωής μου και πιστεύουν στις δυνατότητές μου.

Περιεχόμενα

Περίληψη	1
Abstract	2
Κατάλογος Συντομογραφιών	7
Περιεχόμενα Εικόνων	9
Περιεχόμενα Πινάκων	10
Περιεχόμενα Γραφημάτων.....	11
Κεφάλαιο 1 ^ο : Εισαγωγή	12
1.1. Η σημασία της εργασίας	12
1.2. Σκοπός της εργασίας.....	13
1.3. Δομή Εργασίας.....	13
Κεφάλαιο 2 ^ο : Βιβλιογραφική Ανασκόπηση.....	14
2.1. Ανίχνευση Ανθρώπινης Δραστηριότητας	14
2.2. Ανίχνευση Πτώσης	14
2.3. Ανίχνευση ασυνήθιστης συμπεριφοράς.....	15
2.3.1. Τεχνικές επιβλεπόμενης ταξινόμησης βαθιάς μάθησης.....	15
2.3.2. Τεχνικές μη επιβλεπόμενης ταξινόμησης	16
2.3.3. Τεχνικές για εύρεση ακραίων σημείων (outlier detection)	17
2.3.4. Τεχνικές βαθιάς μάθησης.....	17
2.4. Όραση Υπολογιστών	18
Κεφάλαιο 3 ^ο : Αρχές Τεχνητής Νοημοσύνης	20
3.1. Νοημοσύνη	20
3.2. Τεχνητή Νοημοσύνη.....	20
3.3. Μηχανική Μάθηση (Machine Learning)	21
3.3.1. Επιβλεπόμενη Μάθηση (Supervised Learning)	21
3.3.2. Μη Επιβλεπόμενη Μάθηση	22
3.3.3. Ημι Επιβλεπόμενη Μάθηση (Semi-Supervised Learning)	23
3.4. Νευρωνικά Δίκτυα (Neural Networks-NN)	23
3.4.1. Perceptron	24
3.4.2. Συνάρτηση Ενεργοποίησης (Activation Function)	26
3.4.3. Συνάρτηση Απώλειας (Loss Function)	27
3.5. Βαθιά Μάθηση (Deep Learning)	28
3.5.1. Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks-CNN)	28

3.5.2. Αυτοκωδικοποιητές (Autoencoders).....	31
3.5.3. Περιγραφή Αρχιτεκτονικής Δικτύου	33
3.5.4. Ο συνελκτικός αυτοκωδικοποιητής στην αυτόματη ανίχνευση πτώσεων	34
Κεφάλαιο 4 ^ο : Υλοποίηση σε περιβάλλον Google Colaboratory σε γλώσσα προγραμματισμού Python....	35
4.1. Λίγα λόγια για την γλώσσα προγραμματισμού Python	35
4.1.1. Διαδραστικό Περιβάλλον Προγραμματισμού Python: Google Colaboratory	36
4.2. Σημαντικές Βιβλιοθήκες για την Υλοποίηση του Αυτοκωδικοποιητή	37
4.3. Υλοποίηση του Αυτοκωδικοποιητή.....	39
4.3.1. Προεπεξεργασία των Δεδομένων.....	39
4.3.2. Εκπαίδευση Δικτύου Αυτοκωδικοποιητή	40
4.3.3. Προβλέψεις του δικτύου Αυτοκωδικοποιητή	44
4.3.4. Έλεγχος του δικτύου Αυτοκωδικοποιητή	44
Κεφάλαιο 5 ^ο : Πειραματική Διαχείριση.....	45
5.1. Περιγραφή των Δεδομένων	45
5.1.1. Οπτικά Δεδομένα (RGB Data).....	45
5.1.2. Θερμικά Δεδομένα (Thermal Data)	46
5.2. Πειραματικές Εκτιμήσεις.....	48
5.2.1. Area under the curve of the Receiver operating characteristic (ROC AUC)	48
5.2.2. Δείκτες αξιολόγησης.....	50
5.2.3. Παρουσίαση Δεικτών Αξιολόγησης για το Δίκτυο του Αυτοκωδικοποιητή	52
5.3. Κατασκευή Συνελκτικού Ταξινομητή Μάκρο-Βραχυπρόθεσμης Μνήμης (LSTM).....	55
5.3.1. Αρχιτεκτονική Συνελκτικού Ταξινομητή	55
5.3.2. Προεπεξεργασία Δεδομένων του Συνελκτικού Ταξινομητή	57
5.3.3. Κατασκευή Μοντέλου Ταξινομητή	58
5.3.4. Εκπαίδευση Δικτύου Ταξινομητή.....	59
5.3.5. Έλεγχος του Δικτύου Ταξινομητή LSTM.....	61
5.3.6. Αξιολόγηση του του Δικτύου Ταξινομητή LSTM.....	61
Κεφάλαιο 6 ^ο : Συμπεράσματα.....	66
Βιβλιογραφία	68

Κατάλογος Συντομογραφιών

MOB	Man Overboard
ConvAE	Convolutional Autoencoder
LSTM	Long-Short Term Memory
ROI	Region Of Interest
HMM	Hidden Markov Models
GMM	Gaussian Mixture Models
SOI	Segments of Interest
ICL	Incremental Coding Length
GPR	Gaussian Progress Regression
STIPs	Spatio temporal Interest Points
BSD	Behavior Specific Dictionary
HOG	Histogram of Oriented Gradients
HOF	Histogram of Oriented Flows
AMDN	Appearance an Motion Deep Network
SVM	Support Vector Machines
BGS	Background Subtraction
MF	Median Filtering
FD	Frame differencing
KDE	Kernel Density Estimation
AMC	Absorbing Markov Chain
HOT	Histogram of Oriented Tracklets
LDA	Latent Dirichlet Allocation
ML	Machine Learning
DL	Deep Learning
NN	Neural Networks
ANN	Artificial Neural Networks
ReLU	Rectified Linear Unit

tanh	Hyperbolic Tangent Function
MSE	Mean Squared Error
CNN	Convolutional Neural Networks
FNN	Feedforward Neural Networks
RNN	Recurrent Neural Networks
GC	Garbage Collection
IDE	Integrated Development Environment
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
CPU	Central Processing Unit
RAM	Random Access Memory
API	Application Programming Interface
ROC	Receiver Operating characteristic Curve
AUC	Area Under Curve
RGB	Red Green Blue
Adam	Adaptive Movement
AdaGrad	Adaptive Gradient Algorithm
RMSProp	Root Mean Square Propagation
TPR	True Positive Rate
FPR	False Positive Rate
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives

Περιεχόμενα Εικόνων

Figure 1: Τεχνητή Νοημοσύνη	20
Figure 2: Απλό perceptron	24
Figure 3: Παράδειγμα διαχωρισμού Perceptron	25
Figure 4: Γραφική παράσταση συναρτήσεων tanh και Relu	26
Figure 5: Πράξη Συνέλιξης.....	29
Figure 6: Παράδειγμα Υποδειγματοληψίας Μέγιστης Τιμής.....	30
Figure 7: Παράδειγμα της απλής αρχιτεκτονικής ενός αυτοκωδικοποιητή.....	31
Figure 8: Γενική Αρχιτεκτονική Αυτοκωδικοποιητή	32
Figure 9: Προτεινόμενη Αρχιτεκτονική Δικτύου.....	33
Figure 10: Δείγματα εικόνων γωνία λήψης 45° και οριζόντιας γωνίας λήψης για οπτικά δεδομένα	46
Figure 11: Δείγματα εικόνων γωνίας λήψης 45° και οριζόντιας γωνίας λήψης για θερμικά δεδομένα.....	47
Figure 12: Γραφήματα AUC για τα θερμικά δεδομένα	49
Figure 13: Γραφική Παράσταση AUC για τα οπτικά δεδομένα	49
Figure 14: Κατανόηση των τιμών του Πίνακα Σύγκρισης.....	50
Figure 15: Γραφήματα AUC για τα Θερμικά Δεδομένα.....	61
Figure 16: Γραφήματα AUC για τα Οπτικά Δεδομένα.....	62

Περιεχόμενα Πινάκων

Table 1: Χαρακτηριστικά του Google Colab.....	36
Table 2: Συγκεντρωτικός Πίνακας Παραμέτρων Εκπαίδευσης.....	43
Table 3:Κατανομή των Οπτικών Δεδομένων	46
Table 4:Κατανομή Θερμικών Δεδομένων	47
Table 5:Πίνακας Σύγκρισης.....	50
Table 6:Αξιολόγηση Εκπαίδευσης δικτύου για τα Δεδομένα του Συνελκτικού Αυτοκωδικοποιητή.....	54
Table 7: Συγκεντρωτικός Πίνακας Διαμόρφωσης Δεδομένων	57
Table 8: Συγκεντρωτικός Πίνακας Παραμέτρων Εκπαίδευσης του Συνελκτικού Ταξινομητή LSTM	60
Table 9:Αξιολόγηση Δικτύου Συνελκτικού Ταξινομητή για τα Σετ Δεδομένων	65

Περιεχόμενα Γραφημάτων

Γράφημα 1: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Οριζόντιας Γωνίας Λήψης.....	52
Γράφημα 2: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Γωνίας Λήψης 45°	52
Γράφημα 3: Γράφημα Μελέτης για τα Οπτικά Δεδομένα Οριζόντιας Γωνίας Λήψης.....	53
Γράφημα 4: Γράφημα Μελέτης για τα Οπτικά Δεδομένα Γωνίας Λήψης 45°	53
Γράφημα 5: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Οριζόντιας Γωνίας Λήψης.....	62
Γράφημα 6: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Γωνίας Λήψης 45°	63
Γράφημα 7: Γράφημα Μελέτης για τα RGB Δεδομένα Οριζόντιας Γωνίας Λήψης	63
Γράφημα 8: Γράφημα Μελέτης για τα RGB Δεδομένα Γωνίας Λήψης 45°	64

Κεφάλαιο 1^ο: Εισαγωγή

1.1. Η σημασία της εργασίας

Η εργασία αυτή πραγματεύεται μία κατάσταση έκτακτης ανάγκης κατά την οποία ένας άνθρωπος πέφτει από το πλοίο (Man Overboard- MOB), στη θάλασσα και χρήζει άμεσης βοήθειας. Τα περιστατικά αυτά θα πρέπει να ερευνηθούν και να τεθούν σε προτεραιότητα καθώς ο χρόνος διάσωσης είναι αρκετά περιορισμένος και κρίσιμος, καθώς τα άτομα τίθενται σε υψηλούς κινδύνους, όπως για παράδειγμα, πνιγμό, υποθερμία και τραυματισμούς [1].

Πιο συγκεκριμένα, από τη χρονιά 2000, διακόσιοι ογδόντα τέσσερεις (284) άνθρωποι έχουν πέσει από κρουαζιερόπλοια και σαράντα ένα (41) από μεγάλα πλοία. Κάθε μήνα, περίπου δύο (2) άνθρωποι βρίσκονται σε αυτή την κατάσταση και μόνον το 17% έως το 25% έχουν διασωθεί [2]. Το γεγονός πως το άτομο μένει για μία ώρα μέσα στο νερό, με 4.4 °C, έχει ως αποτέλεσμα να ρίχνει τη θερμοκρασία του σώματος στους 30 °C[3]. Για τον λόγο αυτό, υπάρχει μικρή πιθανότητα επιβίωσης [1].

Αναφέρεται πως η έκτακτη κατάσταση που περιγράφεται, θα μπορούσε να χαρακτηριστεί και ως πρόβλημα ασυνήθιστης συμπεριφοράς (abnormal behavior), είτε ως ανίχνευση πτώσης (fall detection). Το πρόβλημα της ασυνήθιστης συμπεριφοράς, περιλαμβάνει την συνήθη κατάσταση, κατά την οποία δεν υπάρχει κίνδυνος και την ασυνήθιστη, κατά την οποία υπάρχει η πτώση. Το ζήτημα αυτό, κατά κύριο λόγο, προσεγγίζεται με μεθόδους μη επιβλεπόμενης ταξινόμησης, αλλά και με επιβλεπόμενη ταξινόμηση.

Γενικότερα, για την παρακολούθηση του φαινομένου αυτού, χρησιμοποιούνται συστήματα οπτικής παρακολούθησης τα οποία είναι προγραμματισμένα να ελέγχουν στατικά, μία καθορισμένη περίμετρο[1]. Καθώς όμως, ο χρόνος ανταπόκρισης και διάσωσης είναι πολύ σημαντικός, είναι αναγκαίο να υπάρξει και άλλη προσέγγιση, κατά την οποία οι άνθρωποι σε κίνδυνο θα εντοπίζονται αυτόματα. Στην κάτωθι εργασία, προτείνεται ένα πρόγραμμα νευρωνικού δικτύου μη επιβλεπόμενης ταξινόμησης (unsupervised learning) το οποίο έχει ως είσοδό του θερμικά και έγχρωμα βίντεο και είναι ικανό να κάνει αυτόματο διαχωρισμό μεταξύ μίας κανονικής και μίας κρίσιμης κατάστασης, στην προκείμενη περίπτωση, την ανθρώπινη πτώση. Ο αλγόριθμος που προτείνεται είναι αυτός του συνελκτικού αυτοκωδικοποιητή (Convolutional Autoencoder). Ο αυτοκωδικοποιητής (autoencoder) αυτός χρησιμοποιεί Μακρο-Βραχυ Πρόθεσημη Μονάδα Μνήμης (Long-Short Term Memory, LSTM) ως λύση για την αποθήκευση πληροφοριών σε παρατεταμένο χρόνο [4].

1.2.Σκοπός της εργασίας

Σκοπός της εργασίας είναι να δημιουργηθεί ένα πρόγραμμα μη επιβλεπόμενης ταξινόμησης το οποίο θα έχει την ικανότητα να εντοπίσει ένα άτομο που εκτελεί πτώση, ώστε να υπάρξει αντιμετώπιση για την έγκαιρη διάσωσή του. Ακόμα, το πρόγραμμα αυτό πιθανώς να είναι ικανό, να δώσει λύσεις και σε άλλα αντίστοιχα προβλήματα όπου μπορεί να υπάρχει κίνδυνος ανθρώπινης πτώσης, όπως σε ένα εργοτάξιο, όπου γίνονται διάφορα τεχνικά έργα υποδομής, όπως για παράδειγμα δρόμοι, γέφυρες, κτίρια κ.λπ. Επιπρόσθετα, θα μπορούσε να βοηθήσει στην διάσωση επιζώντων από συντρίμια, που μπορεί να έχει προκαλέσει μία φυσική καταστροφή, όπως ο σεισμός. Μπορεί ακόμα να συνεισφέρει σε ατυχήματα στο οδικό δίκτυο, σε περίπτωση σύγκρουσης, για την έγκαιρη περίθαλψη των ατόμων, όπως επίσης και σε σιδηροδρομικά μέσα, σε περίπτωση πτώσης ατόμου από την αποβάθρα στις σιδηροδρομικές γραμμές, είτε εγκλωβισμός του μεταξύ συρμού και αποβάθρας.

1.3.Δομή Εργασίας

Στο επόμενο κεφάλαιο, γίνεται βιβλιογραφική ανασκόπηση σε εργασίες που αντιμετωπίζουν την κατάσταση έκτακτης ανάγκης ή παρόμοια προβλήματα όρασης υπολογιστών με διάφορους τρόπους.

Κατά το τρίτο κεφάλαιο, παρουσιάζονται οι βασικές αρχές της Τεχνητής Νοημοσύνης, γίνονται αναφορές σε τεχνικές Μηχανικής Μάθησης και Βαθιάς Μάθησης. Και τέλος, παρουσιάζεται οι τεχνικές των αλγορίθμων του Αυτοκωδικοποιητή, καθώς γίνεται και εκτενής ανάλυση της αρχιτεκτονικής του δικτύου.

Στο τέταρτο κεφάλαιο, παρουσιάζεται η υλοποίηση του δικτύου μέσα από την γλώσσα προγραμματισμού Python. Παρουσιάζεται η εκπαίδευση και οι προβλέψεις των σφαλμάτων του δικτύου. Ακόμα, γίνεται αναφορά σε σημαντικές βιβλιοθήκες της γλώσσας προγραμματισμού που χρησιμοποιούνται για την υλοποίηση του μοντέλου.

Το πέμπτο κεφάλαιο αποτελείται από μία εκτενή παρουσίαση των δεδομένων που χρησιμοποιούνται, τα πειραματικά αποτελέσματα του Συνελκτικού Αυτοκωδικοποιητή, καθώς επίσης παρουσιάζεται, για σκοπούς, σύγκρισης, μέθοδος επιβλεπόμενης μάθησης με τα ίδια δεδομένα.

Τέλος, στο έκτο κεφάλαιο διατυπώνονται συμπεράσματα σχετικά με τις αποδόσεις του μοντέλου του Συνελκτικού Αυτοκωδικοποιητή, πραγματοποιούνται συγκρίσεις και αναφέρονται μελλοντικές προσθήκες για το δίκτυο.

Κεφάλαιο 2^ο: Βιβλιογραφική Ανασκόπηση

2.1. Ανίχνευση Ανθρώπινης Δραστηριότητας

Γενικά, αρκετές μέθοδοι ανίχνευσης ανθρώπων, είτε ανίχνευσης δραστηριότητας (event/ activity recognition) και έχουν παρουσιαστεί και δίνουν έμφαση κυρίως στη σημαντικότητα των συστημάτων παρακολούθησης σε πραγματικό χρόνο [5], [6]. Πιο συγκεκριμένα, μία πολύ δημοφιλής εφαρμογή είναι η παρακολούθηση των γραμμών παραγωγής σε βιομηχανίες και εργοστάσια. Η σημαντικότητα της εφαρμογής αυτής έγκειται σε θέματα ασφαλείας του προσωπικού, μείωση κόστους, καθώς και ποιοτικό έλεγχο κατά την παραγωγική διαδικασία [5]. Η [5] χωρίζεται σε τρία μέρη για την υλοποίησή της. Το πρώτο μέρος είναι αφιερωμένο στην εύρεση των περιοχών ενδιαφέροντος (Regions of Interest -ROIs) μέσα από το βίντεο, απομονώνοντας αυτές τις περιοχές είναι δυνατόν να εξεταστεί διαφορετική δραστηριότητα σε κάθε περιοχή. Το δεύτερο μέρος αποτελεί την εξέταση της δραστηριότητας και αν αυτή αποτελεί μέρος της παραγωγικής διαδικασίας, για τον σκοπό της παρακολούθησης της γραμμής παραγωγής, γίνεται εφαρμογή της μεθοδολογίας των Μαρκοβιανών Μοντέλων (Hidden Markov Models – HMM). Τέλος, χρησιμοποιείται μία τεχνική αντιστοίχισης συμβολοσειρών για να επιβεβαιωθεί η εγκυρότητα των επιβλεπόμενων ακολουθιών ή να διορθωθούν τα λάθη της ταξινόμησης.

Στην [5], προτείνεται μία προσέγγιση αξιολογητικής διόρθωσης (Evaluative Rectification) με σκοπό να διορθώσει δυναμικά λαθεμένες ταξινομήσεις και να ενισχύσει την απόδοση του μοντέλου και γενικά τα στατιστικά του. Για την υλοποίηση της [5] χρησιμοποιούνται τα Μαρκοβιανά Μοντέλα. Κατά το κομμάτι της ταξινόμησης τα Μαρκοβιανά Μοντέλα, αποτελούν μία αρκετά δημοφιλή προσέγγιση καθώς μπορούν να μοντελοποιήσουν αποτελεσματικά στοχαστικές χρονοσειρές σε διάφορες χρονικές κλίμακες.

2.2. Ανίχνευση Πτώσης

Στο κεφάλαιο αυτό, γίνεται μία επισκόπηση εργασιών που εξετάζουν την ανθρώπινη πτώση και πως αυτή μπορεί να αντιμετωπισθεί με διάφορες μεθόδους. Πιο συγκεκριμένα, στην [7] αναφέρεται η σημαντικότητα της αναβάθμισης των οικιακών συστημάτων παρακολούθησης. Η πρόταση της συγκεκριμένης εργασίας αναφέρεται σε αλγόριθμο αφαίρεσης του υποβάθρου της εικόνας σε πραγματικό χρόνο, βασισμένο στην πληροφορία της κίνησης στη σκηνή και στην ένταση των εικονοστοιχείων (pixel), αλγόριθμο ικανό να λειτουργεί σε δυναμική αλλαγή των οπτικών συνθηκών έτσι ώστε να εντοπίζεται η πτώση. Στην [7] δημιουργείται αλγόριθμος ημι επιβλεπόμενης ταξινόμησης ο οποίος συλλέγει πληροφορίες από την οικιακή κάμερα και σε συνδυασμό με τις συντεταγμένες τις και τον πραγματικό τρισδιάστατο χώρο είναι δυνατό να γίνεται ανίχνευση πτώσης σε πραγματικό χρόνο. Ακόμα αναφέρεται πως για λόγους προσωπικών δεδομένων το σύστημα αυτό, είναι σχεδιασμένο ώστε να μπει σε κατάσταση ανάγκης και να δίνει τη δυνατότητα εγγραφής βίντεο μόνο αφού λάβει χώρα το συμβάν. Η ανίχνευση αυτή είναι συνυφασμένη με την ταχύτητα της κάθετης κίνησης και της αλλαγές στη στάση του σώματος.

Μία άλλη προσέγγιση που αφορά οικιακά ζητήματα είναι αυτή της εργασίας [8]. Κατά την εργασία αυτή πραγματοποιείται παρακολούθηση βίντεο για ανίχνευση πτώσης, βασισμένη στην παραμόρφωση του ανθρώπινου σώματος. Η όραση υπολογιστών παρέχει λύση όσο αφορά την ανάλυση της ανθρώπινης συμπεριφοράς και τον εντοπισμό ασυνήθιστων συμβάντων, όπως οι πτώσεις. Στην εργασία αυτή, η ανίχνευση της πτώσης γίνεται από βίντεο παρακολούθησης αναλύοντας την παραμόρφωση του ανθρώπινου σώματος μέσα από μεθόδους ανάλυσης σχήματος. Έπειτα, οι πτώσεις διαφοροποιούνται από άλλες δραστηριότητες όπως το να κάθεται κανείς χωρίς προσοχή, ή το σκύψιμο. Η διαφοροποίηση γίνεται με Γκαουσιανά μοντέλα (Gaussian mixture models-GMM).

Η εργασία [9] απασχολείται με ένα σύστημα ανίχνευσης πτώσης με βάση τη στάση του σώματος για την παρακολούθηση ενός ηλικιωμένου ατόμου σε ένα έξυπνο οικιακό περιβάλλον. Προτείνει ένα σύστημα όραση υπολογιστών το οποίο χρησιμοποιεί μία και μοναδική κάμερα για την παρακολούθηση ενός ηλικιωμένου, ο οποίος διαμένει μόνος του. Αρχικά, εφαρμόζεται ένας αλγόριθμος αποκοπής έτσι ώστε να εξαχθεί το προσκήνιο, δηλαδή την ανθρώπινη φιγούρα. Έπειτα, γίνεται και άλλη επεξεργασία, έτσι ώστε να υπάρχουν καλύτερα αποτελέσματα. Από τα στοιχεία που έχουν εξαχθεί, αντλούνται δεδομένα από το προσαρμοσμένο ελλειψοειδές και το ιστόγραμμα προβολής, τα οποία χρησιμοποιούνται για σκοπούς ταξινόμησης. Έπειτα, τα δεδομένα τροφοδοτούνται στον κατευθυνόμενο άκυκλο γράφο (Directed Acyclic Graph Support Vector Machine), το οποίο έχει εκπαιδευτεί από δεδομένα που περιέχουν στοιχεία που έχουν εξαχθεί από διαφορετικές στάσεις του σώματος σε διαφορετικούς προσανατολισμούς. Τέλος, η εξαγόμενη ανθρώπινη φιγούρα ταξινομείται ως μία από τις τέσσερις διαφορετικές στάσεις (κύρτωση, κατάκλιση, κάθισμα και όρθια στάση). Τα αποτελέσματα της ταξινόμησης σε συνδυασμό με τις ανιχνεύσιμες πληροφορίες του πατώματος, χρησιμοποιούνται για να βρεθεί αν ή αν δεν υπάρχει πτώση.

2.3. Ανίχνευση ασυνήθιστης συμπεριφοράς

Η [10] ενσωματώνει αυτοκωδικοποιητές και τεχνικές επιβλεπόμενης ταξινόμησης για την εύρεση των ανωμαλιών. Η εργασία αυτή στοχεύει στην σύνδεση της ευρετικής unmasking διαδικασίας και των πολλαπλών ταξινομήσεων δύο δειγμάτων στην στατιστική μηχανική μάθηση.

Μία άλλη προσέγγιση είναι αυτή της [11], όπου αναπτύσσεται ένα νέο συγχωνευμένο μοντέλο διασποράς (new joint sparsity model) το οποίο επιτρέπει τον εντοπισμό ανωμαλιών ακόμα και σε πολλά αντικείμενα. Ακόμα, συστήνει τη μη γραμμικότητα σε αυτό, δηλαδή τον πυρήνα. Το μοντέλο της διασποράς είναι ικανό να διακρίνει τις κλάσεις και κατ' επέκταση τα μη συνηθισμένα γεγονότα.

2.3.1. Τεχνικές επιβλεπόμενης ταξινόμησης βαθιάς μάθησης

Μία εφαρμογή της επιβλεπόμενης ταξινόμησης είναι αυτή που παρουσιάζεται στην [12]. Η εφαρμογή της εστιάζεται στη διαχείριση ροής κυκλοφορίας και πιο συγκεκριμένα, στον εντοπισμό ασυνήθιστης συμπεριφοράς μέσα από έξυπνα συστήματα μεταφοράς. Το άρθρο [11] προτείνει έναν αλγόριθμο ανάκτησης συμβάντων από μεγάλα βίντεο και βασίζεται στην τμηματοποίηση υπερπλασιών (superframe segmentation). Εντοπίζοντας το εύρος της κίνησης ενός μεγάλου βίντεο, πολλές εικόνες που είναι περιττές, μπορούν να αφαιρεθούν, μειώνοντας έτσι τον αριθμό των εικόνων που χρειάζονται για να υπολογισθούν. Έπειτα, εφαρμόζοντας τον αλγόριθμο για την τμηματοποίηση υπερπλασιών βασισμένο στην συγχώνευση χαρακτηριστικών, το απομένον βίντεο χωρίζεται σε πεδία ενδιαφέροντος (Segments of Interest-SOIs) και στη συνέχεια χρησιμοποιείται ένα εκπαιδευμένο σημασιολογικό μοντέλο ώστε να ταυριάζει τα πεδία ενδιαφέροντος σύμφωνα με τα ζητούμενα αντικείμενα.

Μία ακόμα εφαρμογή αποτελεί η έξυπνη παρακολούθηση των δημόσιων χώρων, που προτείνεται στην [13]. Η εργασία αυτή, βασίζεται στον εντοπισμό ασυνήθιστης συμπεριφοράς που χρησιμοποιεί δυαδικά δεδομένα για να κωδικοποιήσει την πληροφορία της κίνησης. Αρχικά, υπολογίζονται οι χρονικές κλίμακες και το παρασκήνιο των εικόνων, έπειτα, εντοπίζονται τα σημεία ενδιαφέροντος μέσω του αλγορίθμου FAST [13]. Στη συνέχεια, εξάγονται τα δυαδικά χαρακτηριστικά και χρησιμοποιούνται Γκαουσιανά μοντέλα (Gaussian mixture models-GMMs) για να μοντελοποιηθούν όλα τα δυαδικά στοιχεία τα οποία λαμβάνονται από τη χωρητικότητα του προσκήνιου. Τέλος, εφαρμόζεται ένας συμπερασματικός μηχανισμός που χρησιμοποιεί Γκαουσιανά μοντέλα, ώστε να βρεθούν τα ασυνήθιστα γεγονότα.

2.3.2. Τεχνικές μη επιβλεπόμενης ταξινόμησης

Στο υποκεφάλαιο αυτό, συναντώνται μέθοδοι μη επιβλεπόμενης ταξινόμησης. Πιο συγκεκριμένα, ξεκινώντας με την εργασία [14], αναφέρεται πως για τον εντοπισμό της ασυνήθιστης συμπεριφοράς δημιουργούνται αμφίδρομα συγκεντρωτικά δίκτυα πολλαπλών κλίμακων (Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection – BMAN). Το προτεινόμενο BMAN εκπαιδεύεται από κανονικά γεγονότα και εντοπίζει τις αποκλίσεις από τα μη κανονικά γεγονότα. Το BMAN αποτελείται από δύο μέρη, το πρώτο αφορά την δια-πλαισιακή πρόβλεψη (inter-frame predictor) και το δεύτερο τον συνδυασμένο εντοπισμό εμφάνισης και κίνησης (appearance- motion joint detector). Ο inter-frame predictor έχει σχεδιαστεί έτσι ώστε να κωδικοποιεί τα κανονικά δεδομένα, ο οποίος στη συνέχεια παράγει ένα δια-πλαίσιο χρησιμοποιώντας αμφίδρομες πολλαπλές κλίμακες βασισμένες στην προσοχή. Με τη συγκέντρωση των δεδομένων είναι δυνατόν να επιτύχει η ευρωστία για παραλλαγές της κλίμακας ενός αντικειμένου και σύνθετες κινήσεις του, σε κανονικά δεδομένα. Τα ασυνήθιστα γεγονότα, εντοπίζονται από τον appearance-motion joint detector, όπου τα χαρακτηριστικά της εμφάνισης και της κίνησης λαμβάνονται υπόψιν. Η εργασία αυτή, έχει εφαρμογές στην παρακολούθηση βιομηχανικών περιβαλλόντων.

Στην [15], οι εκπαιδευτικές ακολουθίες δεν είναι διαθέσιμες και οι ανωμαλίες υπολογίζονται ανεξαρτήτως της χρονικής διάταξης. Ακόμα, η εργασία αυτή δεν χρησιμοποιεί επίβλεψη και δεν απαιτούνται ξεχωριστές εκπαιδευτικές διαδικασίες. Η προσέγγιση αφορά τον απευθείας υπολογισμό της διάκρισης των εικόνων αναφορικά με το γενικό πλαίσιο που αφορά το βίντεο. Ουσιαστικά, εκτελείται μία ανίχνευση αλλαγών (change detection) σε μία ακολουθία δεδομένων από βίντεο, με σκοπό να βρεθούν ποιες εικόνες διαφοροποιούνται από τις προηγούμενες τους, ώστε να γίνει ο εντοπισμός των ανωμαλιών σε σχέση με τα κανονικά δεδομένα. Με σκοπό να είναι οι διακρίσεις ανεξάρτητες από τον χρόνο, τα δεδομένα εισέρχονται με τυχαία σειρά (shuffle). Ακόμα, απλοί ταξινομητές χρησιμοποιούνται για την αποφυγή υπερπροσαρμογής.

Η [16], προσεγγίζει το πρόβλημα του εντοπισμού ασυνήθιστης συμπεριφοράς με μη επιβλεπόμενη ταξινόμηση. Χρησιμοποιεί λεξικά χαρακτηριστικών με βάση τα τοπικά χωροχρονικά δεδομένα χρησιμοποιώντας αντικειμενικό κώδικα διασποράς (sparse coding objective). Η ανωμαλία ενός γεγονότος εξαρτάται από τη συχνότητα που κάθε στοιχείο ανακατασκευάζει όλα τα γεγονότα και από τον απόλυτο συντελεστή. Το ICL (Incremental Coding Length) είναι το μέτρο απόκτησης εντροπίας κάθε χαρακτηριστικού. Δεν χρειάζεται παραμέτρους, είναι υπολογιστικά ικανό και χρησιμοποιείται για τον εντοπισμό ξεχόντων χαρακτηριστικών σε εικόνες.

Μία ακόμη προσέγγιση αποτελεί το Unmasking , το οποίο αναφέρει η [17]. Στην εργασία αυτή, προτείνεται εντοπισμός ασυνήθιστων γεγονότων χωρίς εκπαιδευτικές διαδικασίες. Η τεχνική αυτή, έχει χρησιμοποιηθεί στο παρελθόν για την επιβεβαίωση συγγραφέων σε έγγραφα κειμένου. Η υλοποίηση της τεχνικής αυτής αφορά της επανειλημμένη εκπαίδευση ενός δυαδικού ταξινομητή, για τον διαχωρισμό δύο συνεχών βίντεο, ενώ σε κάθε βήμα αφαιρούνται τα διακριτά χαρακτηριστικά. Τέλος, την καλύτερη ακρίβεια, την παρουσιάζουν τα ασυνήθιστα γεγονότα, έναντι των κανονικών.

Η [18] προτείνει μία μέθοδο ιεραρχικής ταξινόμησης με βάση την ανάλυση της τροχιάς για την εύρεση ασυνήθιστων γεγονότων. Η μέθοδος είναι βασισμένη στην μη επιβλεπόμενη ταξινόμηση των τροχιών των αντικειμένων, οι οποίες μοντελοποιούνται με βάση τα Μαρκοβιανά μοντέλα (hidden Markov models -HMM). Η μέθοδος της δυναμικής ιεράρχησης ενσωματώνεται στους αλγόριθμους ταξινόμησης με βάση την τροχιά, με σκοπό να αποφευχθεί η υπερπροσαρμογή.

2.3.3. Τεχνικές για εύρεση ακραίων σημείων (outlier detection)

Για την εύρεση ακραίων σημείων, οι εργασίες των [19], [20], [20], βασίζονται στην εκμάθηση πληθώρας υπο-γεγονότων, μέσα από μία διαδικασία εκπαίδευσης και τα γεγονότα εκείνα που δεν βρίσκονται στον διαμερισμένο υποχώρο, σημειώνονται ως ασυνήθιστα. Πιο συγκεκριμένα στην [19], παρουσιάζεται ο εντοπισμός τοπικών αλλά και παγκόσμιων ανωμαλιών μέσα από μία ιεραρχική αναπαράσταση στοιχείων και μίας διαδικασίας Γκαουσιανής διαδικασίας παλινδρόμησης (Gaussian process regression -GPR). Η εργασία αυτή, δείχνει ενδιαφέρον όσο αφορά την ασυνήθιστη συμπεριφορά σε παγκόσμιο επίπεδο, όπου αυτή ενέχει πολλαπλά συνηθισμένα γεγονότα τα οποία αλληλοεπιδρούν με ασυνήθιστους τρόπους, όπως ένα τροχαίο ατύχημα. Σκοπός είναι να χρησιμοποιηθούν τα γειτονικά χωροχρονικά σημεία ενδιαφέροντος (STIPs-spatio-temporal interest points) ώστε να γίνει εντοπισμός της ασυνήθιστης συμπεριφοράς.

Η [20], αναφέρεται στην μη επιβλεπόμενη ταξινόμηση συγκεκριμένης συμπεριφοράς λεξικών εκμάθησης για εντοπισμό μη φυσιολογικών γεγονότων. Αποτελείται από δύο μέρη, το πρώτο μέρος αναφέρεται στην εκμάθηση λεξικών συγκεκριμένης συμπεριφοράς (Behavior-Specific Dictionary (BSD) Learning), στα οποία συμπεριλαμβάνονται άτομα που έχουν ένα συγκεκριμένο πρότυπο συμπεριφοράς. Στο δεύτερο μέρος, γίνεται μία βελτίωση των λεξικών αυτών και βρίσκονται οι συμπεριφορές των ατόμων που εμπεριέχονται λαθεμένα στο κάθε λεξικό.

2.3.4. Τεχνικές βαθιάς μάθησης

Κατά τις τεχνικές βαθιάς μάθησης τα μοντέλα εκπαιδεύονται, με κανονικά δεδομένα, έτσι ώστε τα μη κανονικά να δίνονται ως διαφορά των αρχικών εικόνων σε σχέση με τις παραγόμενες εικόνες.

Στην [21] παρουσιάζεται μία προσέγγιση η οποία αποτελείται από δύο συνελκτικούς αυτοκωδικοποιητές (Convolutional Autoencoders- ConvAE), καθώς είναι λειτουργικοί με ελάχιστη ή και καθόλου επίβλεψη. Αρχικά αξιοποιούνται τα συμβατικά χωροχρονικά τοπικά δεδομένα και ύστερα τροφοδοτούνται σε έναν πλήρως συνελκτικό αυτοκωδικοποιητή. Έπειτα κατασκευάζεται ένας πλήρως συνελκτικός feed-forward autoencoder με σκοπό να μάθει και τα τοπικά δεδομένα αλλά και τους ταξινομητές. Ο autoencoder, τροφοδοτείται με τα κανονικά (normal) δεδομένα και σκοπός είναι να ανασχηματίσει τα κανονικά δεδομένα με μικρό σφάλμα και τα μη κανονικά με μεγαλύτερο. Στον autoencoder εισάγονται δεδομένα κίνησης που αποτελούνται από ιστογράμματα προσανατολισμένων διαβαθμίσεων (Histograms of Oriented Gradients- HOG) και από Ιστογράμματα Οπτικών Ροών (Histograms of Optical Flows -HOF).

Στην [22], χρησιμοποιούνται autoencoders, με σκοπό την εκμάθηση στοιχείων και την ανοικοδόμηση των αρχικών εικόνων. Συγκεκριμένα στην [22], παρουσιάζεται ένα δίκτυο βαθιάς μάθησης που αφορά την εμφάνιση και την κίνηση (Appearance and Motion DeepNet -AMDN), το οποίο χρησιμοποιεί νευρωνικά δίκτυα βαθιάς μάθησης με σκοπό να μαθαίνει αυτόματα τις αναπαραστάσεις των χαρακτηριστικών. Με σκοπό να εκμεταλλευτούν οι πληροφορίες των μοτίβων της κίνησης, αλλά και της εμφάνισης, προτείνεται μια διαδικασία η οποία, συνδυάζει παλαιότερες και νεότερες μεθόδους συγχώνευσης (fusion). Ειδικότερα, συνδυάζει τους denoising autoencoders, με σκοπό την εκμάθηση των χαρακτηριστικών της εμφάνισης και της κίνησης και ύστερα εφαρμόζονται πολλαπλές διανυσματικές μηχανές μίας κατηγορίας (multiple one-class SVM models) με σκοπό την πρόβλεψη ασυνήθιστης συμπεριφοράς για κάθε ένα από τα στοιχεία που εισάγονται.

Κατά την [23] προτείνεται μία υβριδική μέθοδος που συνδυάζει τον LSTM (Long Short Term Memory) encoder-decoder(κωδικοποιητή-αποκωδικοποιητή) με τον Convolutional (συνελκτικό) autoencoder

(ConvAE) , με σκοπό την εξαγωγή των χώρο-χρονικών δεδομένων αλλά και τη βελτίωση της ικανότητας παρέκτασης του αντίστοιχου αποκωδικοποιητή με την συντομότερη ανταπόκριση. Η αρχιτεκτονική του υβριδικού μοντέλου που χρησιμοποιείται χωρίζεται σε δύο μέρη. Στο πρώτο γίνεται εφαρμογή του ConvAE για την εκμάθηση της χωρικής δομής για κάθε εικόνα. Στο δεύτερο μέρος χρησιμοποιείται ο LSTM encoder-decoder με τον Conv- LSTM για την εκμάθηση τη την κίνηση των χαρακτηριστικών της κίνησης ενός χρονικού βίντεο για τους χάρτες χωρικών χαρακτηριστικών κάθε χρονικού βήματος.

2.4. Όραση Υπολογιστών

Στο υπό-κεφάλαιο αυτό, γίνεται αναφορά των εργασιών που απασχολούνται με θερμικά και οπτικά δεδομένα.

Ξεκινώντας, η [24] προτείνει μία αφαίρεση του παρασκηνίου η οποία μοντελοποιεί τις θερμικές αντιδράσεις κάθε εικονοστοιχείου σε ένα μείγμα Γκαουσιανών με άγνωστο αριθμό παραμέτρων. Ακολουθεί μια προσέγγιση σύμφωνα με το κανόνα του Bayes και κατά αυτόν τον τρόπο αποφεύγονται η υπερπροσαρμογή (overfitting) και η υποπροσαρμογή (underfitting). Η πυκνότητα του κάθε εικονοστοιχείου υπολογίζεται από έναν αναβαθμισμένο μηχανισμό, ο οποίος επιτρέπει στο σύστημα να προσαρμόζεται στις δυναμικές αλλαγές των συνθηκών λειτουργίας. Αναφέρεται πως χρησιμοποιούνται θερμικοί δεδομένα, καθώς οι αισθητήρες αυτοί δεν έχουν περιορισμό μεταξύ ημέρας και νύχτας καθώς δεν επηρεάζονται από τις αλλαγές του φωτισμού.

Η [25] ασχολείται με οπτικές τεχνικές που δίνουν έμφαση στην μοντελοποίηση του παρασκηνίου και του στόχου. Η [24] συστήνει ένα σύστημα πολλαπλών καμερών ώστε να αποφευχθεί το πρόβλημα που προκαλείται από την εξάρτηση της οπτικής της κάμερας. Ακόμα, εξετάζεται η αποδοτικότητα των Μαρκοβιανών μοντέλων (Markov models) καθώς και μοντέλα Student's t-endowed για ανοχή των άκρων. Προτείνεται ένα σύστημα βασισμένο ε μηχανισμούς νευρωνικών δικτύων που στοχεύει στη δυναμική διόρθωση εσφαλμένων ταξινομήσεων σε εικονικές ακολουθίες και με αυτόν τον τρόπο, βελτιώνονται τα συνολικά ποσοστά της αναγνώρισης.

Κατά την [26] χρησιμοποιούνται τεχνικές αφαίρεσης του παρασκηνίου (BackGround Subtraction-BGS), με σκοπό να βρεθεί κινούμενο αντικείμενο. Οι τεχνικές αυτές χρησιμοποιούνται για να εντοπιστεί το παρασκήνιο, όπου υπάρχει και κίνηση. Οι μέθοδοι που εξετάζονται χωρίζονται σε τρεις κατηγορίες, τις βασικές οι οποίες περιέχουν τη διαφορά μεταξύ των εικόνων (Frame differencing -FD) και την εφαρμογή του Μέσου φίλτρου (Median filtering MF) ,τις παραμετρικές ενέχουν Γκαουσιανά μοντέλα (Simple Gaussian, Mixture of Gaussians, Gamma method) και τέλος, τις μη παραμετρικές που βασίζονται σε προσέγγιση μέσω ιστογράμματος (Histogram-based approach -Hb) και την Εκτίμηση Πυκνότητας του Πυρήνα (Kernel Density Estimation- KDE).

Η [27], έχει εφαρμογή στον εντοπισμό δραστηριότητας. Πιο συγκεκριμένα χρησιμοποιείται η Απορροφητική Αλυσίδα του Μάρκοφ (Absorbing Markov Chain -AMC) στην κατάτμηση υπέρ-εικονοστοιχείων (superpixel segmentation), η οποία χρησιμοποιεί τα superpixels που προσδιορίζονται από δύο διαδοχικά πλαίσια, όπου τα superpixels του παρασκηνίου του προηγούμενου πλαισίου αντιστοιχούν στην απορρόφηση κορυφών, ενώ όλα τα υπόλοιπα δημιουργούν, διαδοχικά, καινούργια.

Στην [28] γίνεται ενασχόληση με την ταυτοποίηση ασυνήθιστων και ύποπτων συμπεριφορών. Η υλοποίησή της είναι βασισμένη στην χρήση των Μαρκοβιανών μοντέλων (hidden Markov models) για διαδικτυακή αναγνώριση εργασιών, καθώς επίσης χρησιμοποιείται και γνώση που βασίζεται στο σχέδιο βελτιστοποίησης κατά Hopfield. Με σκοπό να διευκολυνθεί η ανάλυση εκτός σύνδεσης , εξάγονται περιλήψεις των ήδη καταγεγραμμένων γεγονότων. Ακόμα, εξάγεται η ενέργεια της κίνησης

χρησιμοποιώντας νόρμες κατά τον Zernike, ψάχνοντας και το τοπικό μέγιστο και ελάχιστο που υποδηλώνουν διακριτά οπτικά γεγονότα, δηλαδή εικόνες ενδιαφέροντος.

Η [29] ασχολείται με τη δυναμική του πλήθους. Χρησιμοποιεί Ιστόγραμμα Προσανατολισμένων Ισοδύναμων Μετρήσεων (Histogram of Oriented Tracklets- HOT) για να αναγνωρίζει ασυνήθιστες συμπεριφορές σε πολυσύχναστα μέρη. Χωρίζει τις ακολουθίες των βίντεο σε χωροχρονικά cuboids, από τα οποία συλλέγονται στατιστικά. Δίνεται βάση στα δισδιάστατα ιστογράμματα να οποία κωδικοποιούν μοτίβα κίνησης για κάθε ένα από τα cuboids. Οι εικόνες ταξινομούνται σε συνηθισμένες και ασυνήθιστες σύμφωνα με μοντέλα Latent Dirichlet Allocation (LDA).

Η [30], ακολουθεί μία διαφορετική τεχνική σύμφωνα με τις προηγούμενες εργασίες. Είναι βασισμένη στην ανάλυση τροχιάς (trajectory analysis), για τις εφαρμογές επιτήρησης. Εστιάζει στην ομαδοποίηση, την εύρεση ασυνήθιστης συμπεριφοράς και την παραγωγή σύνοψης.

Η [31] δίνει μία σφαιρική εικόνα σχετικά με τις πιο σημαντικές προσεγγίσεις βαθιάς μάθησης που χρησιμοποιούνται για την όραση υπολογιστών. Αναφέρει για τα συνελκτικά νευρωνικά δίκτυα (Convolutional Neural Networks), τις μηχανές του Boltzmann (Deep Boltzmann Machines), τα Δίκτυα Βαθιάς Πειποιθήσεως (Deep Belief Networks) και τους Αυτοκωδικοποιητές (Stacked Denoising Autoencoders). Αναφέρει τη δομή τους, τα πλεονεκτήματα και τα μειονεκτήματά τους, καθώς επίσης και εφαρμογές που μπορούν να χρησιμοποιηθούν, όσο αφορά την όραση υπολογιστών. Μερικές από τις εφαρμογές είναι, εντοπισμός αντικειμένου, αναγνώριση προσώπου, αναγνώριση κίνησης και δραστηριότητας και εκτίμηση της στάσης του σώματος.

Κεφάλαιο 3^ο: Αρχές Τεχνητής Νοημοσύνης

3.1. Νοημοσύνη [32]

Σύμφωνα με τον Douglas Hofstadter, η νοημοσύνη ανταποκρίνεται σε μία πληθώρα συμπεριφορών, όπως κανείς να αναγνωρίζει και να ιεραρχεί τα διάφορα δεδομένα με βάση της σπουδαιότητά τους, να βρίσκει ομοιότητες μεταξύ καταστάσεων οι οποίες φαντάζουν διαφορετικές, να βρίσκει το άτομο διαφορές μεταξύ καταστάσεων οι οποίες μοιάζουν παρόμοιες. Ακόμα, η νοημοσύνη είναι να ανταποκρίνεται κανείς σε καταστάσεις με ελαστικότητα και να κατανοεί τα ασαφή μηνύματα μέσα από τα συμπραζόμενα. Οι ικανότητες αυτές, αποκτώνται εύκολα από ανθρώπους και βασίζονται στην κοινή λογική που έχει κάθε άτομο.

Με λίγα λόγια, η ικανότητα του ανθρώπου να μπορεί να επεξεργαστεί πληροφορίες και μέσα από αυτές να λαμβάνει κάποια απόφαση, ονομάζεται νοημοσύνη.

3.2. Τεχνητή Νοημοσύνη

Τεχνητή νοημοσύνη είναι ο τομέας της επιστήμης υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων, τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες. Εμφανίζονται χαρακτηριστικά που αποδίδονται στην ανθρώπινη συμπεριφορά, όπως η επίλυση προβλημάτων, η αντίληψη, η μάθηση και η εξαγωγή συμπερασμάτων.

Η τεχνητή νοημοσύνη αποτελείται από βασικά υποσύνολα όπως η Μηχανική Μάθηση (Machine Learning-ML) και η Βαθιά Μάθηση (Deep Learning-DL), τα οποία αποτελούνται επίσης από βασικά κεφάλαια που παρουσιάζονται παρακάτω.

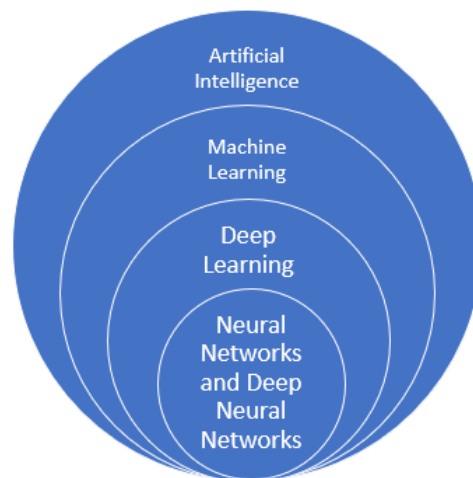


Figure 1: Τεχνητή Νοημοσύνη

3.3. Μηχανική Μάθηση (Machine Learning) [33]

Η Μηχανική Μάθηση αποτελεί έναν κλάδο της Τεχνητής Νοημοσύνης και κατέχει σημαντικό ρόλο στην επιστημονική έρευνα τα τελευταία χρόνια. Η δημιουργία προτύπων από ένα σύνολο δεδομένων, από ένα υπολογιστικό σύστημα, ονομάζεται μηχανική μάθηση. Διερευνά την μελέτη και κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από δεδομένα και να είναι ικανοί να κάνουν προβλέψεις μέσα από αυτά.[34]. Υπάρχουν διάφοροι ορισμοί για την Μηχανική Μάθηση ανά τα χρόνια. Ο Carbonell (1987), διευκρινίζει τον όρο ως «Η μελέτη υπολογιστικών μεθόδων για την απόκτηση νέας γνώσης, νέων δεξιοτήτων και νέων τρόπων οργάνωσης της υπάρχουσας γνώσης». Ο Mitchell (1997) διατύπωσε τη φράση «Ένα πρόγραμμα υπολογιστή θεωρείται ότι μαθαίνει από την εμπειρία E σε σχέση με μία κατηγορία εργασιών T και μία μετρική απόδοσης P , αν η απόδοση του σε εργασίες της T , όπως μετριοούνται από την P , βελτιώνονται με την εμπειρία E ». Και τέλος, οι Witten & Frank (2000) είπαν «Κάτι μαθαίνει όταν αλλάζει τη συμπεριφορά του κατά τέτοιο τρόπο ώστε να αποδίδει καλύτερα στο μέλλον».

Η Μηχανική Μάθηση κυρίως χωρίζεται σε τρεις κατηγορίες, την επιβλεπόμενη ταξινόμηση, την μη επιβλεπόμενη ταξινόμηση και την ημι-επιβλεπόμενη ταξινόμηση. Οι κατηγορίες αυτές αναλύονται παρακάτω.

3.3.1. Επιβλεπόμενη Μάθηση (Supervised Learning)

Η επιβλεπόμενη μάθηση είναι ο πρώτος τύπος της μηχανικής μάθησης, κατά την οποία χρησιμοποιούνται δεδομένα γνωστής ετικέτας (labelled data) με σκοπό να εκπαιδευτεί ο αλγόριθμος. Στην επιβλεπόμενη μάθηση, η είσοδος και η έξοδος των δεδομένων είναι γνωστές. Το σύνολο εισόδου αποτελείται από δεδομένα X , ενώ το σύνολο εξόδου από δεδομένα Y . Ο αλγόριθμος μαθαίνει συγκρίνοντας τα πραγματικά γεγονότα με τις σωστές εξόδους, ώστε να βρει τα σφάλματα. Έπειτα, τροποποιεί αναλόγως το μοντέλο. Τα αρχικά δεδομένα, χωρίζονται σε δύο μέρη, το πρώτο μέρος αφορά τα δεδομένα εκπαίδευσης (train data) και το δεύτερο τα δεδομένα ελέγχου (test data). Η επιβλεπόμενη ταξινόμηση χρησιμοποιεί τα πρότυπα δεδομένων με σκοπό να μπορεί να προβλέψει τα επιπλέον δεδομένα σύμφωνα με τις σωστές ετικέτες.

3.3.1.1. Ταξινόμηση (Classification)

Οι τύποι των επιβλεπόμενων μαθήσεων χωρίζονται κυρίως σε δύο τεχνικές. Την Ταξινόμηση (Classification) και την Παλινδρόμηση (Regression).

Η Ταξινόμηση, είναι ένας τύπος Επιβλεπόμενης Μάθησης όπου τα δεδομένα γνωστής ετικέτας μπορούν να χρησιμοποιηθούν για προβλέψεις μη συνεχούς μορφής (non-continuous form). Η έξοδος, δεν είναι πάντα συνεχής και το γράφημα είναι μη γραμμικό. Στην τεχνική αυτή, ο αλγόριθμος μαθαίνει από τα δεδομένα εισόδου με τα οποία τροφοδοτείται, και τα χρησιμοποιεί με σκοπό να ταξινομήσει νέες παρατηρήσεις, δεδομένα. Τα δεδομένα μπορεί να είναι δύο κλάσεων, ή και πολλών και να γίνει διαχωρισμός μεταξύ αυτών. Υπάρχουν πολλοί αλγόριθμοι ταξινόμησης, οι οποίοι εκτελούν διαφορετικούς σκοπούς. Η ταξινόμηση, μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές, εκ των οποίων, πολύ σημαντική, αποτελεί η αναγνώριση αντικειμένων. Μερικοί από τους πιο σημαντικούς είναι ο αλγόριθμος λογιστικής παλινδρόμησης/ ταξινόμησης (Logistic Regression/Classification), ο αλγόριθμος του εγγύτερου γείτονα K (K-Nearest Neighbors), οι Μηχανές Διανυσματικές Υποστήριξης (Support Vector Machines- SVM), οι Μηχανές Διανυσματικής Υποστήριξης Πυρήνα (Kernel Support Vector Machines-Kernel SVM), ο αλγόριθμος Naive Bayes και τέλος έχουμε τα Δέντρα Απόφασης (Decision Tree Classification).

3.3.1.2. Παλινδρόμηση (Regression)

Η Παλινδρόμηση είναι μία μορφή επιβλεπόμενης μάθησης όπου τα δεδομένα γνωστής ετικέτας χρησιμοποιούνται για να κάνουν προβλέψεις συνεχούς μορφής (continuous form). Ο αλγόριθμος προσπαθεί να γραμμικοποιήσει τις τιμές εισόδου, ουσιαστικά να μεταφράσει τη συμπεριφορά τους μέσω μίας γραμμικής εξίσωσης. Η παλινδρόμηση ερευνά τη σχέση μεταξύ των ανεξάρτητων (δεδομένα εισόδου) και των εξαρτημένων (δεδομένα εξόδου) μεταβλητών. Μερικά παραδείγματα εφαρμογών της παλινδρόμησης αποτελούν η πρόβλεψη του καιρού και η εφαρμογή της για την βελτιστοποίηση κάποιας διαδικασίας (process optimization). Υπάρχουν πολλοί αλγόριθμοι παλινδρόμησης που χρησιμοποιούνται για διαφορετικές εφαρμογές. Μερικοί από τους πιο βασικούς είναι η Απλή Γραμμική Παλινδρόμηση (Simple Linear Regression), η Πολλαπλή Γραμμική Παλινδρόμηση (Multiple Linear Regression), η Πολυωνυμική Παλινδρόμηση (Polynomial Regression), οι Μηχανές Υποστήριξης Διανυσματικής Παλινδρόμησης (Support Vector Regression), η Παλινδρόμηση Ridge, η Παλινδρόμηση Lasso, η Παλινδρόμηση ElasticNet, Η Μπευζιανή Παλινδρόμηση (Bayesian Regression) και τέλος, το Decision Tree Regression.

3.3.2. Μη Επιβλεπόμενη Μάθηση

Η μη Επιβλεπόμενη μάθηση αποτελεί μια κατηγορία της Μηχανικής Μάθησης και εκεί είναι βασισμένη κατά κύριο λόγο και η παρούσα εργασία. Κατά τη μη επιβλεπόμενη μάθηση, ο αλγόριθμος τροφοδοτείται με δεδομένα που δεν έχουν ετικέτα. Σκοπός της είναι να εξερευνήσει τα δεδομένα και να βρει μία δομή μέσα από αυτά. Κατά την διαδικασία αυτή, τα δεδομένα εισέρχονται κατευθείαν στον αλγόριθμο και τις περισσότερες φορές δεν χρειάζεται κάποια προεπεξεργασία. Ακόμα, δεν είναι γνωστά τα δεδομένα εξόδου. Επίσης, στο γενικότερο πλαίσιο της μη επιβλεπόμενης μάθησης, συνήθως το ενδιαφέρον επικεντρώνεται στη μάθηση της πιθανοτικής κατανομής που δημιουργήσε το σύνολο δεδομένων, είτε αναλυτικά, όπως στον υπολογισμό πυκνότητας, είτε έμμεσα, για καταστάσεις όπως σύνθεση (synthesis) ή μετουσίωση (denoising). Κάποιοι από τους αλγόριθμους μη επιβλεπόμενης μάθησης επικεντρώνονται στην συσταδοποίηση (clustering), η οποία αποτελείται από τη διαίρεση των δεδομένων σε παρόμοια παραδείγματα. Τέλος, η μη επιβλεπόμενη μάθηση βρίσκει εφαρμογές και σε ζητήματα μείωσης διάστασης.

3.3.2.1. Συσταδοποίηση (Clustering)

Η Συσταδοποίηση αποτελεί κλάδο της μη επιβλεπόμενης μάθησης κατά την οποία, χρησιμοποιούνται δεδομένα χωρίς ετικέτα και ομαδοποιεί οντότητες με βάση τα κοινά χαρακτηριστικά τους, ύστερα, τα ομαδοποιημένα δεδομένα χρησιμοποιούνται για να φτιαχτούν συστάδες. Ο στόχος αυτής της τεχνικής είναι να βρει ομοιότητες ανάμεσα στα δεδομένα και να τα ομαδοποιήσει και έπειτα, να βρει σε ποια συστάδα ανήκουν τα νέα δεδομένα που θα εισαχθούν. Υπάρχουν πολλοί αλγόριθμοι συσταδοποίησης, οι οποίοι χρησιμοποιούνται για διάφορες εφαρμογές. Μερικοί από αυτούς είναι , η Συσταδοποίηση με βάση τον K μέσο (K-Means Clustering) και η Ιεραρχική Συσταδοποίηση (Hierarchical Clustering).

3.3.2.2. Μείωση Διάστασης (Dimensionality Reduction)

Η μείωση της διάστασης αναφέρεται στη μείωση των δεδομένων από την είσοδο. Η τεχνική αυτή χρησιμοποιείται για την αφαίρεση χαρακτηριστικών που δεν χρειάζονται από τα δεδομένα. Βασίζεται στη διαδικασία τροποποίησης μίας συλλογής δεδομένων που έχουν μεγάλες διαστάσεις σε μικρότερες, διατηρώντας την ουσιαστική πληροφορία που χρειάζεται. Η τεχνική αυτή χρησιμοποιείται στη μηχανική μάθηση, για την απόκτηση καλύτερων χαρακτηριστικών. Μερικοί αλγόριθμοι μείωσης διάστασης είναι η Ανάλυση Κύριων Στοιχείων (Principal Component Analysis), η Γραμμική Ανάλυση Διακρίσεων (Linear Discriminant Analysis) και τέλος, η Ανάλυση Κύριων Στοιχείων με βάση τον Πυρήνα (Kernel Principal Component Analysis).

3.3.3. Ημι Επιβλεπόμενη Μάθηση (Semi-Supervised Learning)

Η Ημι Επιβλεπόμενη μάθηση, είναι ένας κλάδος της Μηχανικής Μάθησης, στον οποίο χρησιμοποιούνται και τα δύο είδη δεδομένων, δηλαδή με ετικέτα και χωρίς. Χρησιμοποιεί μικρό ποσοστό των δεδομένων εισόδου που έχουν ετικέτα και σε σημαντικό μέρος του, δεδομένα που δεν έχουν ετικέτα. Αποτελεί ένα υβρίδιο Επιβλεπόμενης και Μη Επιβλεπόμενης Μάθησης και χρησιμοποιείται για τους ίδιους σκοπούς, όπως η Επιβλεπόμενη Μάθηση με τη διαφορά τους να βρίσκεται στον τύπο δεδομένων που χρησιμοποιούν. Η τεχνική αυτή είναι χρήσιμη, καθώς η διαδικασία επισήμανσης σε μεγάλη ποσότητα δεδομένων είναι χρονοβόρα και ακριβή.

3.4. Νευρωνικά Δίκτυα (Neural Networks-NN)

Νευρωνικό Δίκτυο ονομάζεται ένα κύκλωμα διασυνδεδεμένων νευρώνων, η αρχή του έχει αναπτυχθεί από το γεγονός πως ο ανθρώπινος εγκέφαλος υπολογίζει καταστάσεις με διαφορετικό τρόπο από ότι ένας συμβατικός ηλεκτρονικός υπολογιστής. Ο εγκέφαλος αποτελεί ένα σύνθετο και μη γραμμικό μηχανισμό. Αποτελείται από νευρώνες και είναι ικανός να ανταποκριθεί σε καταστάσεις και υπολογισμούς πολύ πιο γρήγορα από τον πιο ικανό ηλεκτρονικό υπολογιστή στις μέρες μας. Με λίγα λόγια, ένας νευρωνικό δίκτυο αποτελεί έναν παράλληλα κατανεμημένο επεξεργαστή, ο οποίος αποτελείται από απλή επεξεργασία μονάδων και έχει την τάση για αποθήκευση βιωματικών γνώσεων καθιστώντας το κατάλληλο για χρήση. Μοιάζει με τον ανθρώπινο εγκέφαλο σύμφωνα με δύο απόψεις. Πρώτον, η γνώση είναι απαραίτητη για το δίκτυο μέσα από μία εκπαιδευτική διαδικασία από το περιβάλλον του. Δεύτερον, οι ισχύεις του διανευρώνα (Interneuron), γνωστές και ως συναπτικά βάρη, χρησιμοποιούνται για να αποθηκεύσουν την απαραίτητη πληροφορία. Τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks-ANN), έχουν σχεδιαστεί με σκοπό να μοντελοποιήσουν τον τρόπο λειτουργίας των βιολογικών νευρώνων. Αποτελούνται από τεχνητούς νευρώνες, πλήρως συνδεδεμένους μεταξύ τους και δέχονται και επεξεργάζονται δεδομένα μέσα από μία διαδικασία μάθησης. Η υλοποίηση των Τεχνητών Νευρωνικών Δικτύων γίνεται με την ανάπτυξη κατάλληλων αλγορίθμων που εκτελούνται σε ηλεκτρονικό υπολογιστή. Μερικά από τα οφέλη των Νευρωνικών Δικτύων είναι η Μη Γραμμικότητα (Nonlinearity), ένα τεχνητό νευρωνικό δίκτυο μπορεί να είναι, είτε γραμμικό, είτε μη γραμμικό. Ακόμα, υπάρχει η χαρτογράφηση εισόδου και εξόδου (Input–Output Mapping), η οποία απαντάται σε μεθόδους επιβλεπόμενης μάθησης, η Προσαρμογή (Adaptivity), η Αποδεικτική Απάντηση (Evidential Response), Πληροφόρηση σύμφωνα με το γενικό πλαίσιο των δεδομένων (Contextual Information), Ανοχή σε Σφάλματα (Fault Tolerance), VLSI Implementability (very-large-scale-integrated – VLSI), Ομοιομορφία Ανάλυσης Σχεδιασμού (Uniformity of Analysis and Design) και τέλος Νευροβιολογική Ανατομία (Neurobiological Analogy), καθώς των νευρωνικό δίκτυο βασίζεται στον ανθρώπινο εγκέφαλο.

3.4.1. Perceptron [35]

Το perceptron κατέχει μία ιδιαίτερη θέση στην ιστορική ανάπτυξη των νευρωνικών δικτύων. Αποτελεί το πρώτο νευρωνικό δίκτυο το οποίο εφευρέθηκε από το Rosenblatt, ψυχολόγο που εμπνεύστηκε από μηχανικούς, φυσικούς και μαθηματικούς που αφιέρωσαν την ερευνητική τους προσπάθεια σε διαφορετικές πτυχές των νευρωνικών δικτύων κατά τις δεκαετίες 1960 και 1970. Αξιοθαύμαστο αποτελεί το γεγονός πως το perceptron είναι τόσο έγκυρο σήμερα, όσο και κατά τη πρώτη δημοσίευση του Rosenblatt το 1958. Το perceptron, αποτελεί την πιο απλή μορφή νευρωνικού δικτύου και εκτελεί κυρίως εργασίες ταξινόμησης προτύπων που είναι γραμμικά διαχωρίσιμα.

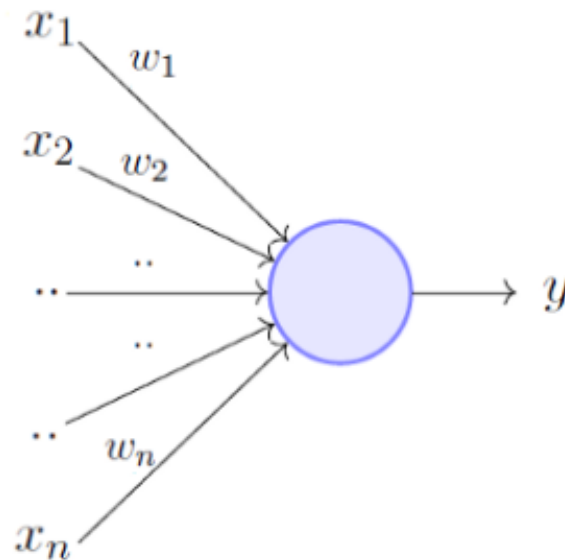


Figure 2: Απλό perceptron

Κυρίως αποτελείται από έναν νευρώνα με ρυθμιζόμενα συναπτικά βάρη και πόλωση (adjustable synaptic weights and bias). Ο Rosenblatt, απέδειξε πως αν τα πρότυπα (διανύσματα) που χρησιμοποιούνται για την εκπαίδευση του perceptron προέρχονται από δύο γραμμικά διαχωρίσιμες κατηγορίες, τότε ο αλγόριθμος συγκλίνει και τοποθετεί την απόφαση σε μορφή hyperplane μεταξύ δύο κλάσεων. Η απόδειξη της σύγκλισης του αλγορίθμου είναι γνωστή ως Θεώρημα Σύγκλισης Perceptron (perceptron convergence theorem). Καθώς το perceptron είναι κατασκευασμένο σύμφωνα με έναν νευρώνα, μπορεί να πραγματοποιήσει ταξινόμηση μόνο ανάμεσα σε δύο κλάσεις. Αν όμως επεκταθεί η έξοδος του δικτύου ώστε να συμπεριληφθούν περισσότεροι νευρώνες, είναι δυνατόν να γίνει ταξινόμηση σε περισσότερες κλάσεις, πάντοτε όμως οι κλάσεις πρέπει να είναι γραμμικά διαχωρίσιμες για να λειτουργήσει.

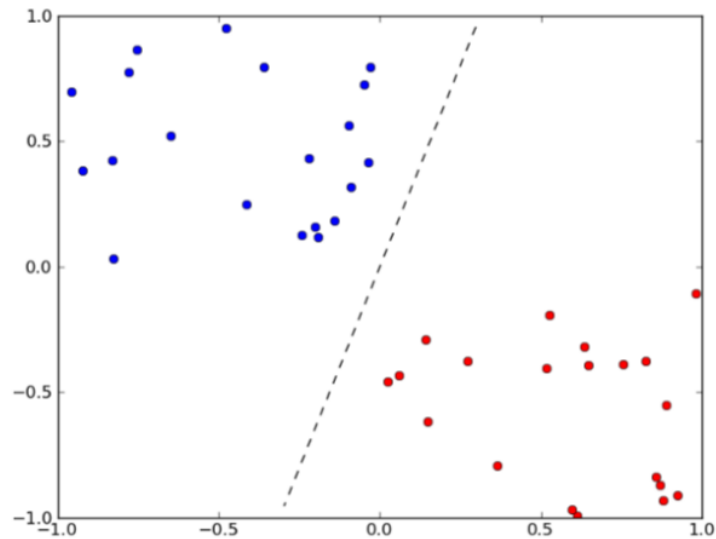


Figure 3: Παράδειγμα διαχωρισμού Perceptron

Στο perceptron για κάθε τιμή εισόδου x_i , υπάρχει βάρος W_i , όπου υπολογίζεται το εσωτερικό γινόμενο τους και χρησιμοποιείται ως είσοδος του νευρώνα. Στην τελική τιμή εισόδου, προστίθεται η τιμή b (bias), η οποία μετατοπίζει το όριο της απόφασης και δεν εξαρτάται από κάποια τιμή εισόδου. Έπειτα από την παραπάνω διαδικασία υπολογίζεται η τιμή εξόδου y .

$$y = x_i W_i + b$$

3.4.2. Συνάρτηση Ενεργοποίησης (Activation Function) [36]

Γενικά, στα νευρωνικά δίκτυα, η έξοδος του νευρώνα προκύπτει από εφαρμογή της συνάρτησης ενεργοποίησης στην συνολική του έξοδο y . Η συνάρτηση ενεργοποίησης χρησιμοποιείται στα τεχνητά νευρωνικά δίκτυα (ANN) με σκοπό να βοηθήσει το εκάστοτε δίκτυο να μάθει τα πολύπλοκα πρότυπα των δεδομένων. Ουσιαστικά, η συνάρτηση αυτή παίρνει το σήμα εξόδου από τον προηγούμενο νευρώνα και το μετατρέπει, σε μία μορφή ώστε να μπορεί να χρησιμοποιηθεί ως είσοδο στον επόμενο νευρώνα. Οι συναρτήσεις αυτές, χρησιμοποιούνται κυρίως, με σκοπό να βοηθήσουν την διατήρηση της τιμής εξόδου από τον νευρώνα σε ένα ορισμένο όριο, σύμφωνα με τις απαιτήσεις του σχεδιαστή του αλγορίθμου. Η είσοδος της συνάρτησης ενεργοποίησης αποτελείται από το εσωτερικό γινόμενο $x_i W_i$ με την πρόσθεση της τιμής b . Η τιμή που αναφέρθηκε, αν δεν είναι περιορισμένη εντός συγκεκριμένου ορίου, μπορεί να γίνει η αιτία σε υπολογιστικά θέματα. Οι συναρτήσεις ενεργοποίησης, είναι είτε γραμμικές, είτε μη γραμμικές. Οι μη γραμμικές προτιμώνται σε σχέση με τις γραμμικές καθώς αντικατοπτρίζουν πραγματικά δεδομένα τα οποία είναι πάντοτε μη γραμμικά.

Αξίζει να αναφερθεί, πως στην παρούσα εργασία χρησιμοποιούνται οι συναρτήσεις ενεργοποίησης, η Ανορθωτική Γραμμική Συνάρτηση (Rectified Linear activation Unit-ReLU) και η Συνάρτηση Υπερβολικής Εφαπτομένης (Hyperbolic Tangent Function –tanh). Η ReLU, γενικώς χρησιμοποιείται ως συνάρτηση ενεργοποίησης για τα κρυφά επίπεδα (Hidden Layers) στη βαθιά μάθηση. Έχει ισχυρά βιολογικά και μαθηματικά θεμέλια. Η συνάρτηση αυτή μετατρέπει τις αρνητικές τιμές σε μηδέν, ενώ τις θετικές τιμές τις διατηρεί. Αποτελεί συνήθη επιλογή κατά τη διαδικασία της μάθησης καθώς έχει σταθερή παράγωγο για τις θετικές τιμές [37].

Όσο αφορά τη Συνάρτηση Υπερβολικής Εφαπτομένης, είναι μία συνάρτηση που χρησιμοποιείται επίσης, αρκετά συχνά σε αλγορίθμους βαθιάς μάθησης. Είναι μία ομαλή συνάρτηση με κέντρο το μηδέν, όπου το εύρος της κυμαίνεται από τιμές μείον ένα έως ένα (-1,1). Η συνάρτηση αυτή, μοιάζει με τη σιγμοειδή (sigmoid), προτιμάται όμως περισσότερο καθώς δίνει καλύτερες επιδόσεις κατά την εκπαίδευση σε πολυεπίπεδα νευρωνικά δίκτυα. Η συνάρτηση tanh βοηθάει σε προβλήματα οπισθοδιάδοσης (back-propagation) [38].

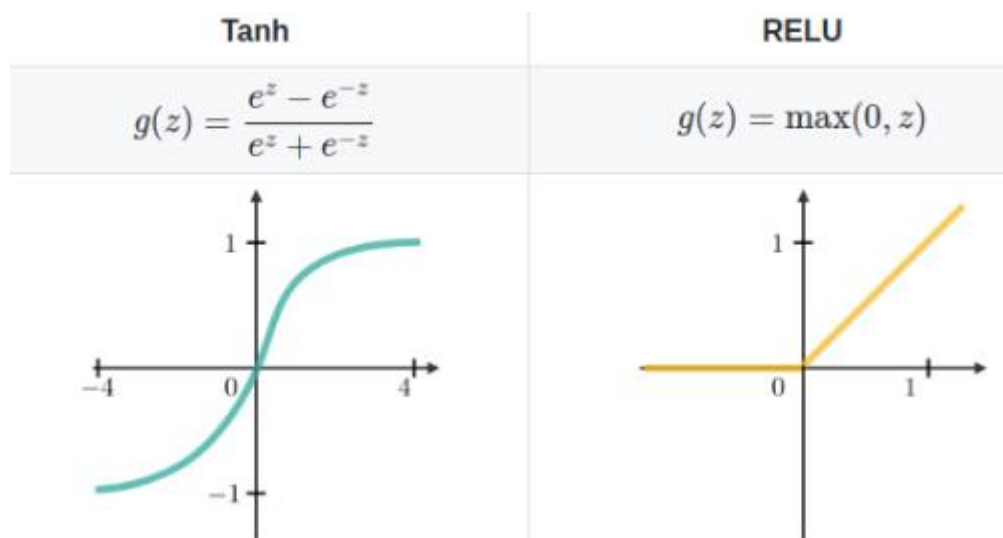


Figure 4:Γραφική παράσταση συναρτήσεων tanh και Relu

3.4.3. Συνάρτηση Απώλειας (Loss Function)

Κατά τη μαθηματική βελτιστοποίηση και τη θεωρία των αποφάσεων, η συνάρτηση απώλειας είναι μία συνάρτηση που χαρτογράφει ένα γεγονός ή τις τιμές ενός ή περισσότερων μεταβλητών σε πραγματικούς αριθμούς που αντιπροσωπεύουν το κόστος που επιφέρει μία κατάσταση, στην συγκεκριμένη περίπτωση, την απώλεια που έχει το δίκτυο. Η συνάρτηση απωλειών είναι ιδιαίτερα σημαντική για τα νευρωνικά δίκτυα, καθώς προβλέπει το σφάλμα για το εκάστοτε δίκτυο. Με άλλα λόγια, η συνάρτηση αυτή υπολογίζει την ακρίβεια του μοντέλου του νευρωνικού δικτύου. Αν η συνάρτηση απώλειας είναι υψηλή, το μοντέλο δεν έχει καλή απόδοση. Μία μικρή τιμή της συνάρτησης απώλειας σημαίνει και μία καλή απόδοση του μοντέλου. Γενικά υπάρχουν τέτοιες συναρτήσεις που προσφέρονται από τις βιβλιοθήκες Keras και Tensorflow, ανάλογα με το θέμα που ασχολείται ο αναλυτής. Στην προκείμενη περίπτωση, χρησιμοποιείται η συνάρτηση του Μέσου Τετραγωνικού Σφάλματος (Mean Squared Error- MSE). Η συνάρτηση αυτή υπολογίζει την τετραγωνική διαφορά μεταξύ των προβλέψεων του μοντέλου και των πραγματικών δεδομένων[39].

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Equation 1: Η συνάρτηση απώλειας MSE

3.5. Βαθιά Μάθηση (Deep Learning)

Ένα ακόμα κεφάλαιο της Τεχνητής Νοημοσύνης αποτελεί η Βαθιά Μάθηση. Αποτελεί έναν τύπο Μηχανικής Μάθησης, εμπνευσμένο από τον ανθρώπινο εγκέφαλο. Οι αλγόριθμοι βαθιάς μάθησης προσπαθούν ολοένα και περισσότερο να εξάγουν συμπεράσματα που είναι παρόμοια με εκείνα του ανθρώπου. Οι τεχνικές βαθιάς μάθησης όπως τα νευρωνικά δίκτυα βαθιάς μάθησης, τα νευρωνικά δίκτυα βαθιάς πίστωσης (deep belief networks), τα βαθιά δίκτυα ενισχυτικής μάθησης (deep reinforcement learning), τα επαναλαμβανόμενα νευρωνικά δίκτυα (recurrent neural networks) και τα συνελκτικά νευρωνικά δίκτυα (convolutional neural networks), έχουν εφαρμογές σε πολλούς τομείς όπως η όραση υπολογιστών, αναγνώριση προτύπων, αναγνώριση λόγου, βιοπληροφορική, ανάλυση ιατρικών εικόνων, καθώς και σε πολλές ακόμα εφαρμογές [40].

Η Βαθιά μάθηση είναι μία διαδικασία που είναι ικανή όχι μόνο να μάθει την σχέση ανάμεσα σε δύο ή περισσότερες μεταβλητές αλλά να κατανοήσει και τι ελέγχει αυτή τη σχέση και το νόημά της[41].

Το πλεονέκτημα της βαθιάς μάθησης έναντι της μηχανικής μάθησης έγκειται στο γεγονός πως η βαθιά μάθηση δεν είναι απαραίτητη η εξαγωγή χαρακτηριστικών (feature extraction), δηλαδή δεν χρειάζεται προεπεξεργασία και οι αλγόριθμοί της μπορούν να εφαρμοστούν κατευθείαν σε αρχικά δεδομένα, όπως εικόνες, αρχεία τύπου csv και άλλα. Ακόμα, η βαθιά μάθηση μπορεί να χρησιμοποιήσει δεδομένα μεγάλου εύρους (Big Data) [42].

3.5.1. Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks-CNN)

Το Συνελκτικό Νευρωνικό Δίκτυο, αποτελεί αλγόριθμο βαθιάς μάθησης, ο οποίος έχει ως κύριο στόχο την επεξεργασία εικόνων. Οι εικόνες θεωρούνται σαν δισδιάστατο κάναβο εικονοστοιχείων (2-D grid of pixels). Το όνομα των συνελκτικών νευρωνικών δικτύων υποδηλώνει δίκτυα που χρησιμοποιούν μία μαθηματική διεργασία, την συνέλιξη, η οποία αποτελεί μία γραμμική πράξη. Τα συνελκτικά Νευρωνικά Δίκτυα διαχωρίζονται ως προς τα απλά νευρωνικά, καθώς χρησιμοποιούν την πράξη συνέλιξης έναντι του γενικού πολλαπλασιασμού πινάκων σε τουλάχιστον ένα από τα επίπεδά του[43]. Τα συνελκτικά νευρωνικά δίκτυα συνδυάζουν τρεις αρχιτεκτονικές ιδέες με σκοπό να εξασφαλίσουν σε ένα βαθμό τη μετατόπιση και παραμόρφωση. Χρησιμοποιούν τοπικά αντίστοιχα πεδία (local respective fields), μοιραζόμενα βάρη (shared weights) και κάποιες φορές χρονική υποδειγματοληψία (temporal subsampling). Εκτός από την εφαρμογή τους σε δεδομένα εικόνων, έχουν τη δυνατότητα να μειώνουν τον όγκο των δεδομένων, διατηρώντας μόνο την απαραίτητη πληροφορία.

Τα συνελκτικά Νευρωνικά Δίκτυα έχουν πολλές εφαρμογές στον εντοπισμό συμβάντων, που πραγματεύεται και η παρούσα εργασία, στην αυτοματοποίηση μηχανών, στην όραση υπολογιστών, σε ιατρικές εικόνες και σε πολλά ακόμα χωρία.

3.5.1.1. Πράξη Συνέλιξης (Convolutional Operation)

Η πράξη συνέλιξης αποτελεί μία πράξη μεταξύ της εκάστοτε εικόνας εισόδου (τανυστής-tensor) με ένα φίλτρο πυρήνα (kernel) ίδιου βάθους. Το αποτέλεσμα αυτής της πράξης πολλές φορές αναφέρεται ως χάρτης χαρακτηριστικών (feature map), επομένως η πράξη συνέλιξης είναι υπεύθυνη για την εξαγωγή χαρακτηριστικών[33]. Πιο συγκεκριμένα για την πράξη συνέλιξης, το φίλτρο μετακινείται πάνω στην εικόνα σύμφωνα με κάποιο χωρικό βήμα (stride) που ορίζεται, έως ότου καλύψει όλο το πλάτος. Καθώς οι εικόνες δεν αποτελούν γραμμικά δεδομένα είναι απαραίτητη μετά από κάθε πράξη συνέλιξης μία συνάρτηση ενεργοποίησης.

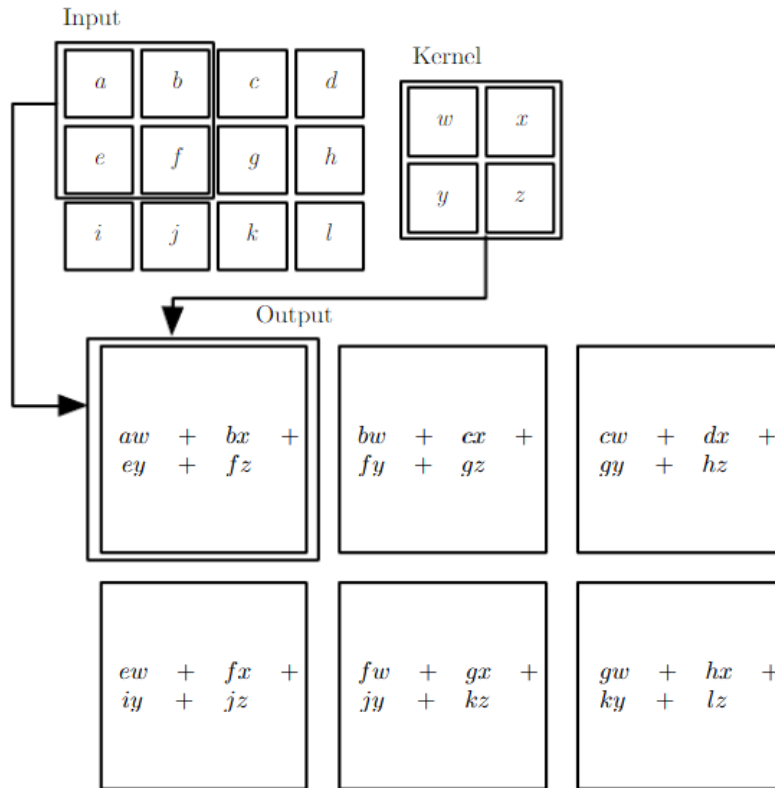


Figure 5: Πράξη Συνέλιξης

3.5.1.2. Επίπεδο Υποδειματοληψίας (Pooling Layer)

Ένα συνηθισμένο επίπεδο συνελκτικού νευρωνικού δικτύου αποτελείται από τρία στάδια. Κατά το πρώτο στάδιο, τα επίπεδα πραγματοποιούν πολλαπλές παράλληλες πράξεις συνέλιξης στα δεδομένα με σκοπό να δημιουργηθεί μία σειρά από γραμμικές ενεργοποιήσεις. Στο δεύτερο στάδιο κάθε γραμμική ενεργοποίηση περνάει από μία μη γραμμική συνάρτηση ενεργοποίησης με σκοπό την εξάλειψη της γραμμικής συμπεριφοράς, το στάδιο αυτό ονομάζεται στάδιο ανίχνευσης (detector stage). Τέλος, στο τρίτο στάδιο, χρησιμοποιείται μία η διαδικασία της υποδειματοληψίας. Στόχος του σταδίου αυτού, αποτελεί η μείωση των διαστάσεων των δεδομένων, κρατώντας μόνο την απαραίτητη πληροφορία τους[33].

Το επίπεδο υποδειματοληψίας είναι υπεύθυνο για την μείωση των συνελκτικών χαρακτηριστικών που προκύπτουν. Με την μείωση των διαστάσεων, κατορθώνει να υπάρχει μικρότερη υπολογιστική ισχύ για την επεξεργασία των δεδομένων. Επίσης, το επίπεδο αυτό είναι χρήσιμο στην εξαγωγή κυρίαρχων χαρακτηριστικών που είναι αμετάβλητα όσο αφορά την περιστροφή και τη θέση, επομένως διατηρείται μία διαδικασία αποτελεσματικής εκπαίδευσης του μοντέλου. Υπάρχουν δύο τύποι υποδειματοληψίας, αυτή της μέγιστης τιμής (Max Pooling) και εκείνη του μέσου όρου (Average Pooling). Η Max Pooling διαδικασία, είναι εκείνη που έχει επιλεγεί για την παρούσα εργασία, επιστρέφει την μέγιστη τιμή του τμήματος της εικόνας που έχει εφαρμοστεί το φίλτρο (Kernel). Στην περίπτωση της Average Pooling διαδικασίας επιστρέφεται ο μέσος όρος της εικόνας που έχει εφαρμοστεί το φίλτρο. Ακόμα, η Max Pooling

προτιμάται καθώς έχει τη δυνατότητα να αφαιρέσει τον θόρυβο της εικόνας (Noise Suppressant), μαζί με τη μείωση των διαστάσεων. [44]

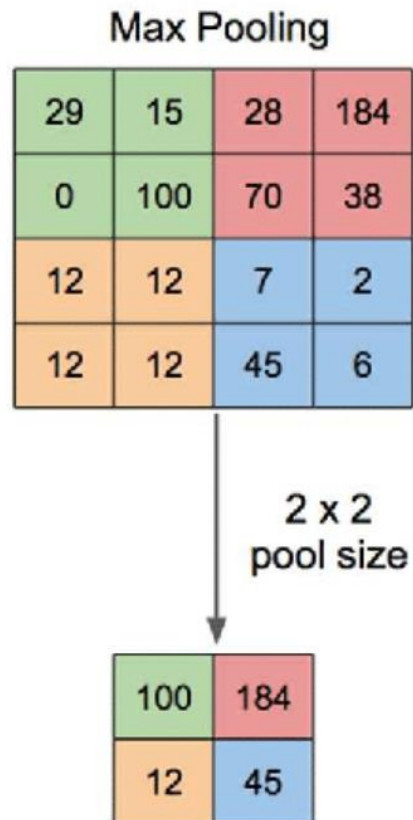


Figure 6: Παράδειγμα Υποδειγματοληψίας Μέγιστης Τιμής

3.5.2. Αυτοκωδικοποιητές (Autoencoders)

Ο αυτοκωδικοποιητής (autoencoder) είναι ένα νευρωνικό δίκτυο μη επιβλεπόμενης μάθησης που εκπαιδεύεται έτσι ώστε τα δεδομένα εισόδου να είναι ίδια με αυτά της εξόδου. Εσωτερικά, έχει ένα κρυφό επίπεδο h που περιγράφει έναν κώδικα, που χρησιμοποιείται για να απεικονίσει την είσοδο. Το δίκτυο μπορεί να θεωρηθεί πως αποτελείται από δύο κομμάτια, τη συνάρτηση του κωδικοποιητή (encoder) $h = f(x)$ και του αποκωδικοποιητή (decoder) που παράγει τον ανασχηματισμό $r = g(h)$. Ο αυτοκωδικοποιητής για να είναι λειτουργικός, πρέπει να σχεδιαστεί με τέτοιο τρόπο ώστε να αντιγράφει επακριβώς την είσοδο που αντικατοπτρίζει τα δεδομένα εκπαίδευσης[33].

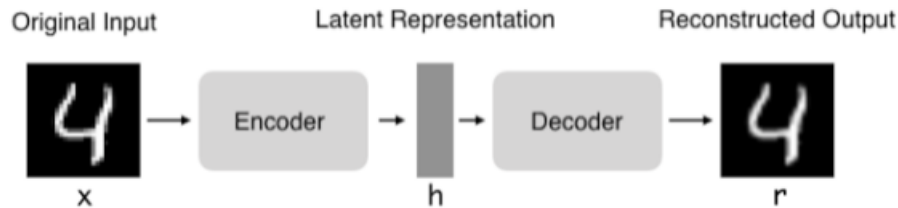


Figure 7: Παράδειγμα της απλής αρχιτεκτονικής ενός αυτοκωδικοποιητή

Η ιδέα των αυτοκωδικοποιητών συμπληρώνει δεκαετίες στην ιστορία των νευρωνικών δικτύων. Παραδοσιακά, οι αυτοκωδικοποιητές χρησιμοποιούνταν για μείωση της διάστασης ή για εκμάθηση χαρακτηριστικών. Πρόσφατα, οι θεωρητικές συνδέσεις μεταξύ των αυτοκωδικοποιητών και των λανθανόντων μεταβλητών μοντέλων (latent variable models), έχουν φέρει τους αυτοκωδικοποιητές στο προσκήνιο των generative models. Οι αυτοκωδικοποιητές θα μπορούσαν να θεωρηθούν και μία εξειδικευμένη περίπτωση των νευρωνικών δικτύων εμπρόσθιας τροφοδοσίας (feedforward neural networks-FNN).

Οι αυτοκωδικοποιητές έχουν την ικανότητα να ανακαλύπτουν συσχετίσεις μεταξύ των δεδομένων και για τον λόγο αυτό είναι ιδιαίτερα χρήσιμοι στην ανίχνευση ασυνήθιστης συμπεριφοράς.

Οι αυτοκωδικοποιητές διαθέτουν τέσσερα κύρια μέρη. Πρώτο μέρος είναι ο encoder, όπου το μοντέλο μαθαίνει να μειώνει τις διαστάσεις των δεδομένων εισόδου και να συμπιέζει τα δεδομένα σε μία κωδικοποιημένη αναπαράσταση. Ύστερα, υπάρχει το Bottleneck επίπεδο, το οποίο περιέχει την συμπιεσμένη κωδικοποιημένη αναπαράσταση των αρχικών δεδομένων. Εκεί βρίσκεται και η μικρότερη διάσταση που μπορούν να πάρουν τα δεδομένα. Τρίτο μέρος αποτελεί ο decoder, όπου το μοντέλο μαθαίνει πως να ανασχηματίζει τα δεδομένα από την κωδικοποιημένη αναπαράσταση, ώστε να είναι όσο παρόμοια, όσο είναι δυνατόν, με την αρχική τους μορφή. Τέλος, είναι η απώλεια ανασχηματισμού (Reconstruction Loss)[45]. Η μέθοδος αυτή μετράει την απόδοση του αποκωδικοποιητή και πόσο κοντά βρίσκονται τα δεδομένα εξόδου σε σχέση με τα αρχικά δεδομένα. Γενικά, η αρχιτεκτονική του αυτοκωδικοποιητή βρίσκει ομοιότητες με τις αρχιτεκτονικές των δικτύων εμπρόσθιας τροφοδοσίας, τα δίκτυα βραχυπρόθεσμης μνήμης (Long Short Term Memory-LSTM) καθώς και τα νευρωνικά δίκτυα.

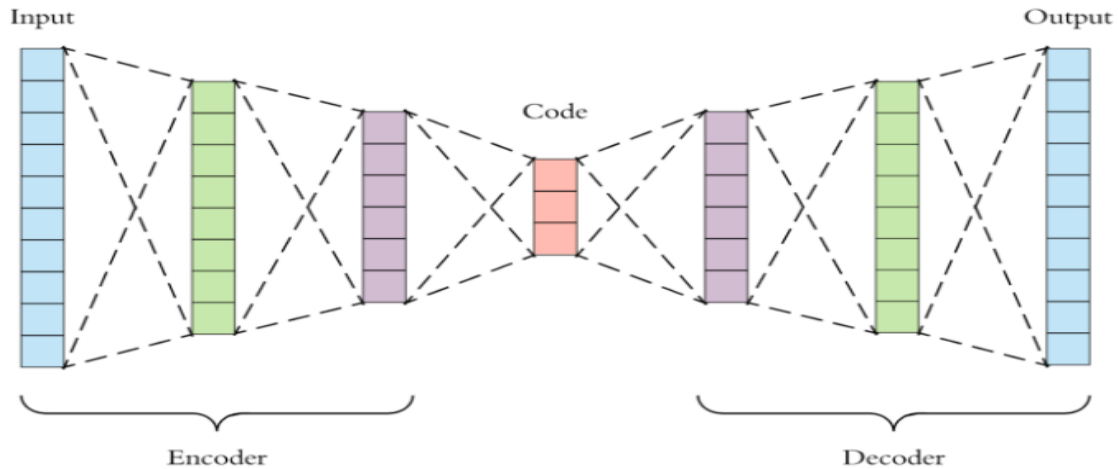


Figure 8: Γενική Αρχιτεκτονική Αυτοκωδικοποιητή

Οι αυτοκωδικοποιητές μπορούν να χρησιμοποιηθούν σε πληθώρα εφαρμογών αναλόγως με το πρόβλημα που καλούνται να αντιμετωπίσουν. Μπορούν να χρησιμοποιηθούν για αναγνώριση προσώπου, εντοπισμό χαρακτηριστικών, εντοπισμό ασυνήθιστης συμπεριφοράς, μείωση διάστασης, συμπίεση εικόνας, αφαίρεση θορύβου από εικόνα, ανάλυση κύριων συνιστωσών (Principal component Analysis-PCA), εξαγωγή χαρακτηριστικών, δημιουργία εικόνας, πρόβλεψη ακολουθιών, σύστημα συστάσεων, αλλά και πολλές εφαρμογές ακόμα [46].

Υπάρχουν πολλοί αλγόριθμοι αυτοκωδικοποιητών που χρησιμοποιούνται ευρέως, μερικοί από αυτούς παρουσιάζονται παρακάτω. Αρχικά, υπάρχει ο Denoising Autoencoder (DAE), ο οποίος χρησιμοποιείται για αφαίρεση θορύβου σε εικόνες. Έπειτα, υπάρχει ο Variational autoencoder (VAE) ο οποίος αποτελεί μία νέα και πιο πολύπλοκη μορφή αυτοκωδικοποιητή και μαθαίνει τις παραμέτρους κατανομής των δεδομένων εισόδου, έναντι της μάθησης αυθαίρετης συνάρτησης που χρησιμοποιείται γενικότερα. Ακόμα, υπάρχει ο Sparse Autoencoder (SAE), ο οποίος μαθαίνει αναπαραστάσεις ενθαρρύνοντας τη διασπορά η οποία βελτιώνει την απόδοση σε εργασίες ταξινόμησης. Επίσης, υπάρχει ο Convolutional Autoencoder, που χρησιμοποιείται σε εφαρμογές που υπάρχουν δεδομένα εικόνων (3D vectors), ή οπτικά δεδομένα γενικότερα, όπως βίντεο, ο Spatiotemporal Autoencoder και ο LSTM Autoencoder. Στην εργασία, παρουσιάζεται ο Spatiotemporal Convolutional LSTM Autoencoder που αποτελεί ένα συνονθύλευμα των τριών τελευταίων αλγορίθμων.

3.5.2.1. Spatiotemporal Convolutional LSTM Autoencoder

Οι Spatiotemporal Convolutional Autoencoders, χρησιμοποιούνται με σκοπό να μοντελοποιήσουν δεδομένα μεγάλης διάστασης σε τεχνικές μη επιβλεπόμενης μάθησης. Στόχος είναι να εξαχθούν χωροχρονικές αναπαραστάσεις έτσι ώστε να διαχωριστούν τα κανονικά και μη κανονικά γεγονότα στις ακολουθίες των βίντεο. Ακόμα χρησιμοποιούνται επίπεδα LSTM με σκοπό την εκμάθηση των μοτίβων των κανονικών δεδομένων από τα βίντεο που χρησιμοποιούνται ως είσοδος.

Οι αλγόριθμοι LSTM αποτελούν μία εξειδικευμένη μορφή αλγορίθμων Επαναλαμβανόμενων Νευρωνικών Δικτύων (Recurrent Neural Networks –RNN), οι οποίοι είναι ικανοί να μαθαίνουν βραχυπρόθεσμες εξαρτήσεις, συστήθηκαν από τους Hochreiter & Schmidhuber (1997) και πλέον χρησιμοποιούνται ευρέως για την αντιμετώπιση πληθώρας προβλημάτων. Οι αλγόριθμοι LSTM έχουν σχεδιαστεί έτσι ώστε να αποφύγουν το πρόβλημα της βραχυπρόθεσμης μνήμης και θυμούνται πληροφορίες μεγάλων χρονικών περιόδων[47].

3.5.3. Περιγραφή Αρχιτεκτονικής Δικτύου

Όπως αναφέρθηκε προηγουμένως, η παρούσα εργασία προσπαθεί να λύσει το πρόβλημα της ανίχνευσης πτώσεων με την εφαρμογή συνελκτικού αυτοκωδικοποιητή. Η αρχιτεκτονική του αυτοκωδικοποιητή φαίνεται στο παρακάτω σχήμα.

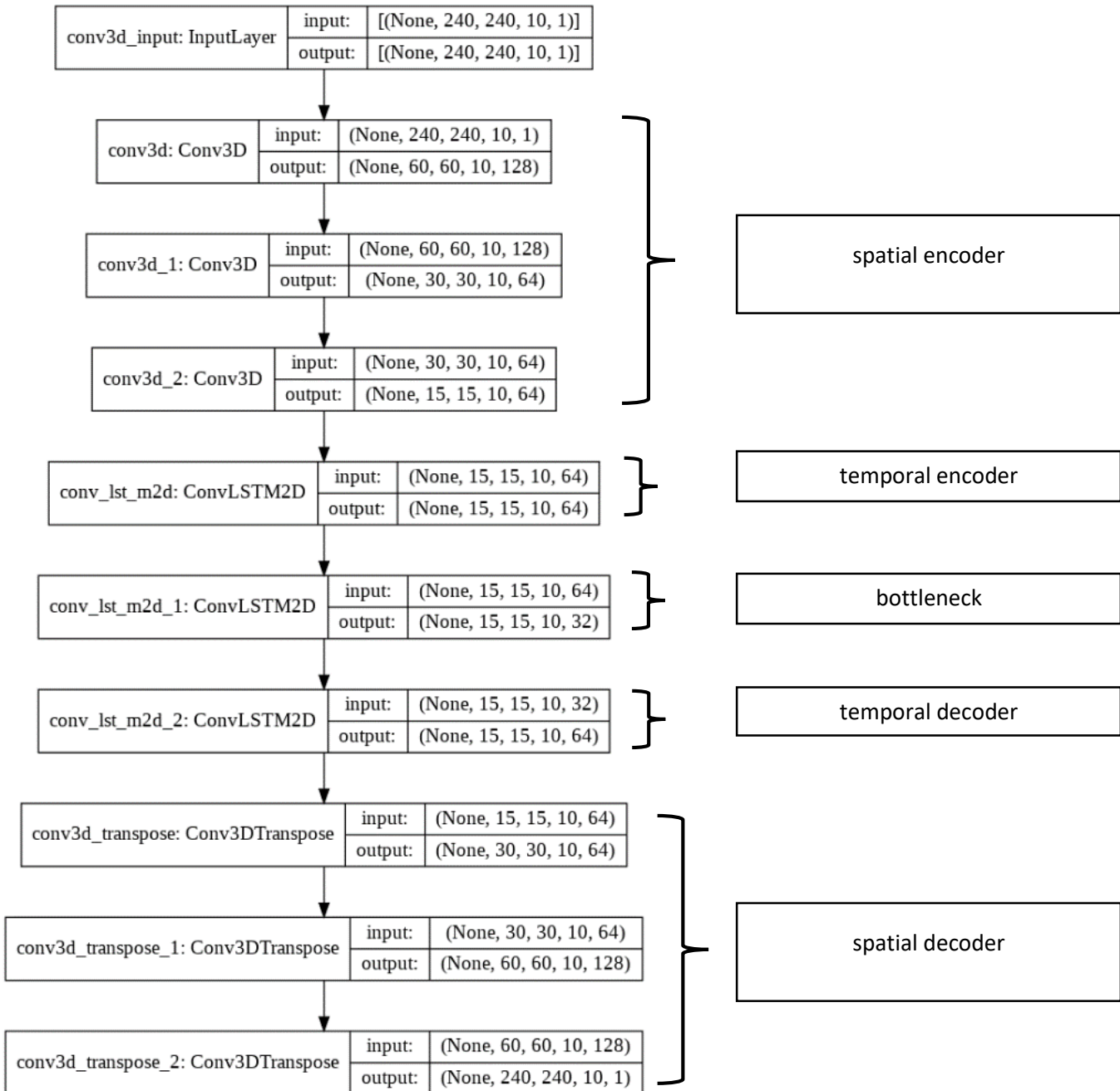


Figure 9: Προτεινόμενη Αρχιτεκτονική Δικτύου

Ξεκινώντας, υπάρχει το επίπεδο εισόδου το οποίο δέχεται εικόνες με τη μορφή πενταδιάστατου τανυστή (5D tensor). Έπειτα, πραγματοποιούνται τρεις επαναλαμβανόμενες πράξεις συνέλιξης με σκοπό να μειωθεί η διάσταση των δεδομένων, εφαρμόζοντας τη συνάρτηση ενεργοποίησης \tanh . Εν συνεχεία, ακολουθούν τα επίπεδα LSTM καθώς παρέχουν σταθερότητα και ευρωστία στη διατήρηση βραχυπρόθεσμων εξαρτήσεων. Τέλος, εφαρμόζονται μεταθετικές συνέλιξεις (transposed convolutions), με σκοπό να πραγματοποιηθεί μετασχηματισμός προς την αντίθετη κατεύθυνση από μία κανονική συνέλιξη, έτσι ώστε να τα δεδομένα εξόδου, να πάρουν την μορφή των αρχικών δεδομένων, διατηρώντας ένα μοτίβο συνδεσιμότητας που είναι συμβατό με την εν λόγω συνέλιξη. Αναφέρεται πως η αρχιτεκτονική του δικτύου θα έχει δεχθεί μεγαλύτερη ανάλυση κατά το τέταρτο κεφάλαιο της εργασίας.

3.5.4. Ο συνελκτικός αυτοκωδικοποιητής στην αυτόματη ανίχνευση πτώσεων

Η χρήση των αυτοματοποιημένων συστημάτων που εντοπίζουν ασυνήθιστη συμπεριφορά, όπως η πτώση, είναι αρκετά επιθυμητή και οδηγεί σε καλύτερη και ευρεία παρακολούθηση ασφάλειας. Γενικά, η διαδικασία αυτή αποτελεί μία πρόκληση για τους ερευνητές. Υπάρχει μεγάλη ζήτηση για την ανάπτυξη των συστημάτων αυτών με σκοπό να είναι γρήγορος ο εντοπισμός, αλλά και ακριβής.

Ο συνελκτικός αυτοκωδικοποιητής εκπαιδεύεται με μη επιβλεπόμενο τρόπο και εκεί έγκειται και η χρησιμότητά του. Σε σχέση με μεθόδους επιβλεπόμενης μάθησης, η μέθοδος του αυτοκωδικοποιητή που έχει δημιουργηθεί δεν χρειάζεται δεδομένα γνωστής ετικέτας, καθώς σε άλλη περίπτωση τα δεδομένα που περιέχουν συμβάντα ασυνήθιστης συμπεριφοράς θα έπρεπε να τιλοφορηθούν και αυτή η εργασία είναι χρονοβόρα απαιτεί αρκετό εργατικό δυναμικό[48].

Κεφάλαιο 4^ο: Υλοποίηση σε περιβάλλον Google Colaboratory σε γλώσσα προγραμματισμού Python

4.1. Λίγα λόγια για την γλώσσα προγραμματισμού Python

Η Python είναι μία αξιολημείωτα δυναμική γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως και για πολλές εφαρμογές. Έχει πολύ καθαρή και ευανάγνωστη σύνταξη, ισχυρές ικανότητες ενδοσκόπησης, διαισθητικό αντικειμενικό προσανατολισμό (intuitive object orientation), φυσική έκφραση του διαδικαστικού κώδικα, αρκετά πακέτα υποστήριξης (βιβλιοθήκες), χειρισμό σφαλμάτων βάσει εξαίρεσης, υψηλού επιπέδου δυναμικού τύπου δεδομένα, εκτεταμένες βιβλιοθήκες που καλύπτουν σχεδόν κάθε πιθανή εφαρμογή, επεκτάσεις που εύκολα γράφονται σε άλλες γλώσσες προγραμματισμού (C, C++, Java, Jython, .NET, IronPython) και τέλος, είναι ενσωματωμένη σε εφαρμογές με επιφάνεια κειμένου. Η Python, είναι κατάλληλη για αρχάριους και έμπειρους προγραμματιστές. Η φιλοσοφία σχεδιασμού της Python, έχει μία αντικειμενοστραφή προσέγγιση με σκοπό να βοηθήσει τους προγραμματιστές να γράφουν καθαρούς και λογικούς κώδικες[49].

Η γλώσσα αυτή, ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστραφές (object-oriented programming) προγραμματιστικό υπόδειγμα (programming paradigm). Ακόμα, υποστηρίζει τη συλλογή απορριμμάτων (garbage collection-GC), χρησιμοποιώντας καταμέτρηση αναφοράς (reference counting).

Η Python ξεκίνησε να ερευνάται στα τέλη του 1980 από τον Guido van Rossum στο Centrum Wiskunde & Informatica (CWI) στην Ολλανδία [50] και η πρώτη έκδοση της Python 0.9.0. εμφανίστηκε το 1991 [51]. Η Python 2.0. εκδίδεται στις 16 Οκτωβρίου του 2000 με πολλαπλές νέες δυνατότητες συμπεριλαμβανομένου ενός cycle detecting garbage collector με σκοπό την υποστήριξη του Unicode[52]. Στις 3 Δεκεμβρίου του 2008, εκδίδεται η Python 3, η οποία είναι η πρώτη γλώσσα προγραμματισμού που σπάει την προς τα πίσω συμβατότητα με προηγούμενες εκδόσεις, έτσι ώστε να διορθωθούν κάποια λάθη προγενέστερων εκδόσεων [52].

Η Python αποτελεί μία εύκολη στην κατανόηση της γλώσσα προγραμματισμού, κυρίως λόγω της εύκολης ανάγνωσής της. Χρησιμοποιεί αγγλικούς χαρακτήρες και δεν χρησιμοποιεί τονισμούς. Σε σχέση με άλλες γλώσσες προγραμματισμού, δεν χρησιμοποιεί αγκύλες και τα ερωτηματικά μετά τις δηλώσεις επιτρέπονται αλλά είναι σπάνια. Σύμφωνα με τον Guido Van Rossum « Το όμορφο είναι καλύτερο από το άσχημο»

Η γλώσσα αυτή είναι ανοιχτού λογισμικού και διατίθεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL (Γενική Άδεια Δημόσιας Χρήσης GNU) [54].

Για να εκτελεστούν διαδραστικά (interactively) μεμονωμένες εντολές ή προγράμματα είναι απαραίτητη η εγκατάσταση του διεργμηνευτή της Python, ο οποίος είναι ελεύθερα διαθέσιμος για «κατέβασμα» (download) από τον επίσημο ιστότοπό της (www.python.org). Για Microsoft Windows υπάρχουν εκδόσεις των 32 ή 64 bits. Στα λειτουργικά συστήματα Linux και Mac OS X συνηθίζεται να είναι προεγκατεστημένη, πιθανότατα όμως να είναι μια παλαιότερη έκδοσή της[54].

Για την υλοποίηση της γλώσσας προγραμματισμού, είναι απαραίτητος ένας κειμενογράφος, είτε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE), το οποίο αποτελεί λογισμικό για την ανάπτυξη των εφαρμογών. Η Python, συνοδεύεται από ένα απλό περιβάλλον ανάπτυξης με την ονομασία IDLE (Interactive DeveLopment Environment). Το περιβάλλον αυτό είναι γραμμένο σε Python από τον Guido Van Rossum, χρησιμοποιεί τη βιβλιοθήκη γραφικών Tkinter, οπότε μπορεί να εκτελεσθεί σε περιβάλλον Linux, Windows και Mac OS X. Δίνει τη δυνατότητα να χρησιμοποιείται

διαδραστικά ο διερμηνευτής της γλώσσας, να γραφτούν και να επεξεργαστούν και να εκτελεστούν προγράμματα, καθώς επίσης εκτελείται η διαδικασία της αποσφαλμάτωσης (debugging). [54]. Υπάρχει πληθώρα περιβάλλοντων ανάπτυξης που διατίθενται για κάθε τύπο λειτουργικού συστήματος (Windows, Linux, Mac OS X και άλλα), μερικά από αυτά είναι το PyCharm, Spyder, IDLE, Microsoft Visual Studio, PyDev, Eclipse και άλλα.

4.1.1. Διαδραστικό Περιβάλλον Προγραμματισμού Python: Google Colaboratory

Στην παρούσα εργασία, για την εκπόνησή της, χρησιμοποιείται το διαδραστικό περιβάλλον προγραμματισμού (Interactive Python Programming Environment), Google Colaboratory (Google Colab). Το Google Colab, αποτελεί ένα προϊόν της Google Research που επιτρέπει στους προγραμματιστές να γράφουν και να εκτελούν κώδικες γλώσσας προγραμματισμού Python, μέσω του προγράμματος περιήγησής τους. Το περιβάλλον αυτό αποτελεί εξαιρετικό εργαλείο για εργασίες που απαιτούν βαθιά μάθηση. Ουσιαστικά, είναι μία εξειδικευμένη μορφή του Jupyter notebook, που δεν χρειάζεται κάποια εγκατάσταση. Το Google Colab, αποτελεί ένα κατά βάση ανοιχτό περιβάλλον προγραμματισμού, προσφέροντας υπολογιστικούς πόρους, όπως GPU και TPU, οι οποίες αποτελούν μονάδες επεξεργασίας γραφικών και μειώνουν σημαντικά τον χρόνο που απαιτείται για την εκτέλεση ενός προγράμματος. Επιπροσθέτως, το Google Colab έχει ήδη εγκατεστημένες σχεδόν όλες τις απαραίτητες βιβλιοθήκες, οι οποίες είναι διαθέσιμες ανά πάσα στιγμή. Ένα ιδιαίτερο χαρακτηριστικό του προγράμματος είναι πως πραγματοποιεί αυτόματη αποθήκευση στο Cloud και έτσι δεν υπάρχει κίνδυνος να καταστραφεί ο αλγόριθμος που έχει ήδη υλοποιηθεί. Ακόμα, το Google Colab, επιτρέπει τη συνεργασία μεταξύ των προγραμματιστών για μία κοινή εργασία. [55].

Η παρούσα εργασία χρησιμοποιεί την έκδοση 3.7.12 της γλώσσας προγραμματισμού Python, καθώς επίσης υλοποιεί την μονάδα επεξεργασίας γραφικών GPU Tesla K80. Στον παρακάτω πίνακα παρουσιάζονται κάποια σημαντικά χαρακτηριστικά του προγράμματος.

Table 1: Χαρακτηριστικά του Google Colab

Hardware specs	CPU	GPU
Όνομα	Intel(R) Xeon(R)	Tesla K80
Μνήμη	12 GB	12 GB
Διαθέσιμη RAM	12 GB	12 GB

4.2. Σημαντικές Βιβλιοθήκες για την Υλοποίηση του Αυτοκωδικοποιητή

Η εργασία αυτή, υλοποιήθηκε με τη γλώσσα προγραμματισμού Python, καθώς είναι ανοιχτού κώδικα, δηλαδή δεν επιφέρει κόστος λειτουργίας και είναι διαθέσιμη για όλους. Ακόμα, σημαντική συμβολή της είναι οι βιβλιοθήκες που διαθέτει που δίνουν εύκολη δυνατότητα στον προγραμματισμό νευρωνικών δικτύων. Οι κύριες βοηθητικές βιβλιοθήκες είναι η TensorFlow και η εφαρμογή προγραμματισμού διεπαφής (application programming interface –API), αυτού, το Keras.

Η TensorFlow αποτελεί ένα περιβάλλον (interface) έκφρασης της Μηχανικής Μάθησης και εφαρμογή εκτέλεσης τέτοιου είδους αλγορίθμων. Το σύστημα αυτό, είναι ευέλικτο και μπορεί να χρησιμοποιηθεί για να εκφράσει πληθώρα αλγορίθμων, συμπεριλαμβανομένων αλγορίθμων εκπαίδευσης για νευρωνικά δίκτυα βαθιάς μάθησης. Ακόμα, έχει χρησιμοποιηθεί για διαχείριση ερευνών και για αξιοποίηση συστημάτων μηχανικής μάθησης σε πολλούς κλάδους της επιστήμης υπολογιστών, καθώς και σε άλλους τομείς όπως η αναγνώριση του λόγου, όραση υπολογιστών, εντοπισμός πληροφοριών, ρομποτική, γεωγραφικά συστήματα πληροφοριών, καθώς και σε πολλούς ακόμα κλάδους[56].

Η TensorFlow αναπτύχθηκε από ερευνητές και μηχανικούς που εργάζονταν στην ομάδα Google Brain, με τη βοήθεια του οργανισμού Google Machine Intelligence Research αξιοποιώντας δίκτυα μηχανικής και βαθιάς μάθησης[57].

Η βιβλιοθήκη Keras, είναι μία βιβλιοθήκη ανοιχτού κώδικα που παρέχεται από την Python για προγραμματισμό σε Τεχνητά Νευρωνικά Δίκτυα (ANN) και αποτελεί μέρος της πλατφόρμας μηχανικής μάθησης TensorFlow[58]. Η Keras, περιέχει διάφορες εφαρμογές που χρησιμοποιούνται συνήθως στα νευρωνικά δίκτυα, όπως επίπεδα (layers), συναρτήσεις ενεργοποίησης, βελτιστοποιητές (optimizers), καθώς και πολλά εργαλεία που διευκολύνουν την εργασία με δεδομένα εικόνας και κειμένου, ώστε να απλοποιηθούν οι απαραίτητοι κώδικες για την εκτέλεση νευρωνικών δικτύων βαθιάς μάθησης. Η βιβλιοθήκη αυτή, υποστηρίζει επίπεδα χρησιμότητας όπως το dropout, batch normalization και pooling[59].

Η Keras, είναι απλή χωρίς να είναι απλοϊκή. Μειώνει το γνωστικό φορτίο με σκοπό να αφήσει τον εκάστοτε προγραμματιστή να ασχοληθεί με προβλήματα μεγαλύτερης βαρύτητας. Είναι ευέλικτη και υιοθετεί τη αρχή της προοδευτικής αποκάλυψης της πολυπλοκότητας (principle of progressive disclosure of complexity), ουσιαστικά οι μη απλές εργασίες πρέπει να είναι εύκολες και οι εργασίες προχωρημένου επιπέδου να είναι δυνατόν να επιλυθούν με ένα γνωστικό μονοπάτι που ο προγραμματιστής έχει ήδη μάθει. Τέλος, η βιβλιοθήκη αυτή είναι ισχυρή και παρέχει δυνατότητες και αποδόσεις βιομηχανικών τάξεων συμπεριλαμβανομένων την NASA και το Youtube[58].

Τρία θεμελιώδη επίπεδα της Keras που χρησιμοποιούνται στην εργασία είναι το Conv3D layer, ConvLSTM2D layer και τέλος, το Conv3DTranspose layer

Άλλες βιβλιοθήκες που βοήθησαν στην εκπόνηση του δικτύου είναι η NumPy, η OpenCV, η matplotlib, καθώς και η scikit-learn.

Η βιβλιοθήκη NumPy έχει τη δυνατότητα να φτιάξει έναν αριθμητικό πίνακα αντικειμένου (Numeric array object), στόχος της είναι να αποτελέσει έναν ακρογωνιαίο λίθο για ένα χρήσιμο περιβάλλον στον επιστημονικό υπολογισμό. Η NumPy, διαθέτει υποστήριξη για μεγάλα πολυδιάστατα διανύσματα και πίνακες, μαζί με μία μεγάλη συλλογή μαθηματικών συναρτήσεων υψηλού επιπέδου για τη λειτουργία αυτών. Η βιβλιοθήκη αυτή διαθέτει υποδομές που μπορούν να στηρίξουν βασικές τεχνικές γραμμικής άλγεβρας και μετασχηματισμούς Fourier, καθώς και γεννήτριες τυχαίων αριθμών.

Δημιουργήθηκε το 2005 από τον Travis E. Oliphant και αποτελεί πακέτο επηρεασμένο από την προγενέστερη βιβλιοθήκη Numeric[60]. Η NumPy, είναι ικανή να αποθηκεύει και να χρησιμοποιεί δεδομένα, όπως εικόνες. Οι εικόνες με πολλά κανάλια αποτελούν τρισδιάστατα διανύσματα, επομένως μπορούν εύκολα να αποθηκευτούν και να χρησιμοποιηθούν μέσω της μορφής NumPy και μπορεί να γίνει οποιαδήποτε πράξη σε αυτές[60]. Στην παρούσα εργασία, χρησιμοποιείται η βιβλιοθήκη αυτή με σκοπό οι εικόνες που εξάγονται από τις ακολουθίες βίντεο να αποθηκεύονται σε πενταδιάστατο ταυνοστή, διατηρώντας όλες τις πληροφορίες, αλλά και παρέχοντας ευκολία πράξεων με τις εικόνες.

Η OpenCV (Open Source Computer Vision Library), αποτελεί μία βιβλιοθήκη ανοιχτού κώδικα προγραμματιστικών συναρτήσεων. Αρχικά δημιουργήθηκε από την Intel και αργότερα ενισχύθηκε από τον Willow Garage[61]. Είναι διαθέσιμη από το έτος 2000 υπό της άδεια ελεύθερου λογισμικού BSD. Η βιβλιοθήκη αυτή, παρέχει εργαλεία που χρειάζονται σε προβλήματα όρασης υπολογιστών. Παρέχει συναρτήσεις για επεξεργασία εικόνας και αλγορίθμους για προβλήματα όπως η αναγνώριση προσώπου, η ανίχνευση πεζών, αντιστοίχιση χαρακτηριστικών και άλλα. Κατά το έτος 2010, μία νέα μονάδα που παρέχει επιτάχυνση GPU προστέθηκε στην OpenCV, όπου η μονάδα αυτή καλύπτει μεγάλο εύρος της λειτουργικότητας της βιβλιοθήκης[62]. Η OpenCV, έχει αρκετές περιοχές εφαρμογής οι οποίες περιλαμβάνουν, εργαλείοι για δισδιάστατα και τρισδιάστατα χαρακτηριστικά, Egomotion estimation, αναγνώριση προσώπου, αναγνώριση χειρονομίας, αλληλεπίδραση μεταξύ ανθρώπου και υπολογιστή, κατανόηση της κίνησης, εντοπισμός αντικειμένων, τμηματοποίηση και αναγνώριση, επαυξημένη πραγματικότητα και άλλα. Για την υποστήριξη των παραπάνω τομέων, η βιβλιοθήκη περιλαμβάνει μία στατιστική βιβλιοθήκη μηχανικής μάθησης που περιέχει ενίσχυση (Boosting), δέντρα αποφάσεων, αλγόριθμο μεγιστοποίησης προσδοκίας (Expectation-maximization algorithm), αλγόριθμο εγγύτερου γείτονα, ταξινομητή Naive Bayes, τεχνητά νευρωνικά δίκτυα, δίκτυα βαθιάς μάθησης, Support vector machine (SVM) και άλλα [61].

Η βιβλιοθήκη OpenCV χρησιμοποιείται στην εργασία με σκοπό να διαβάσει τις ακολουθίες των βίντεο και να εξάγει τις εικόνες μέσα από αυτές, αλλά και για να πραγματοποιήσει αλλαγή μεγέθους στις παραγόμενες εικόνες.

Η βιβλιοθήκη Matplotlib είναι μία βιβλιοθήκη της Python που έχει σχεδιαστεί για την παραγωγή διαγραμμάτων. Παρέχει αντικειμενοστραφή API, για ενσωματωμένα γραφήματα σε εφαρμογές χρησιμοποιώντας τις εργαλείοι γενικής χρήσης του GUI. Η Matplotlib δημιουργήθηκε από τον John D. Hunter και έκτοτε έχει ενεργή κοινότητα ανάπτυξης και είναι διαθέσιμη με την άδεια ανοιχτού λογισμικού BSD [63]. Στην εργασία, χρησιμοποιείται με σκοπό να αποδοθούν τα απαραίτητα γραφήματα.

Τέλος, η βιβλιοθήκη scikit-learn (sklearn), αποτελεί μία βιβλιοθήκη μηχανικής μάθησης ανοιχτού λογισμικού, για τη γλώσσα προγραμματισμού Python. Διαθέτει διάφορους αλγορίθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης συμπεριλαμβανομένων των διανυσμάτων μηχανικής υποστήριξης (SVM), του αλγόριθμου κ μέσου (k-means), του DBSCAN. Έχει σχεδιαστεί να λειτουργεί και με αριθμητικές και επιστημονικές βιβλιοθήκες, όπως η NumPy[64].

Η βιβλιοθήκη scikit-learn χρησιμοποιείται με σκοπό την αξιολόγηση των αποτελεσμάτων των δικτύων. Συγκεκριμένα, με τη βοήθειά της υπολογίζεται η καμπύλη ROC-AUC (Receiver Operating characteristic Curve- Area Under Curve), ο πίνακας σύγχυσης (Confusion Matrix) καθώς και οι δείκτες αξιολόγησης accuracy, precision score, recall score και f1 score.

4.3. Υλοποίηση του Αυτοκωδικοποιητή

Στο υποκεφάλιο αυτό, παρουσιάζονται αναλυτικά οι τεχνικές μη επιβλεπόμενης μάθησης που χρησιμοποιούνται για την εκπαίδευση και τον έλεγχο του δικτύου του Αυτοκωδικοποιητή, μέσα από την ανάλυση της αρχιτεκτονικής του, καθώς και οι προεπεξεργασία των δεδομένων. Ακόμα, γίνεται εκτενής ανάλυση των συναρτήσεων που χρησιμοποιούνται, μέσα από τις διάφορες βιβλιοθήκες.

4.3.1. Προεπεξεργασία των Δεδομένων

Τα δεδομένα που χρησιμοποιούνται στην παρούσα εργασία αποτελούνται από μία σειρά ακολουθιών βίντεο που εξετάζονται σε επόμενο κεφάλαιο εκτενέστερα.

Με σκοπό τα δεδομένα να είναι σε θέση να εισέλθουν στο δίκτυο είναι απαραίτητο να γίνει μία επεξεργασία αυτών. Αρχικά, κατασκευάζεται αλγόριθμος με σκοπό την εξαγωγή εικόνων από τις ακολουθίες βίντεο. Με τη βοήθεια της βιβλιοθήκης OpenCv (cv2) είναι δυνατόν να διαβαστούν οι ακολουθίες βίντεο καθώς και να εξαχθούν οι εικόνες τους, με τις συναρτήσεις `vidcap.read` και `imwrite` αντιστοίχως.

Υστερα, οι εικόνες που έχουν εξαχθεί, διαβάζονται από τον αλγόριθμο και στη συνέχεια γίνεται αλλαγή μεγέθους αυτών (`resize`) με σκοπό να μειωθούν οι διαστάσεις τους και να διατηρηθεί η απαραίτητη πληροφορία τους. Πιο συγκεκριμένα, οι διαστάσεις τους πλέον είναι $240 \times 240 \times 3$ (όπου τρία είναι τα κανάλια τους- RGB).

```
img_resized = cv2.resize(img, (IMG_HEIGHT, IMG_WIDTH))
```

Έπειτα, οι εικόνες αυτές δέχονται μία ακόμα επεξεργασία αλλαγής των καναλιών τους, με σκοπό να βρεθούν στην κλίμακα του γκρι (`grayscale`) και να έχουν στην τιμή των καναλιών τους τον αριθμό 1. Η διαδικασία αυτή πραγματοποιείται σύμφωνα με το παρακάτω απόσπασμα του αλγορίθμου.

```
img_gray = 0.2989*img_resized[:,:,0]+0.5870*img_resized[:,:,1]+0.1140*img_resized[:,:,2]
```

Τέλος, οι επεξεργασμένες πλέον, εικόνες αποθηκεύονται σε έναν μαθηματικό πίνακα `npz`, χρησιμοποιώντας της βιβλιοθήκη `NumPy` και έπειτα ο πίνακας αυτός μετατρέπεται στο πενταδιάστατο τανυστή (5D tensor), που χρησιμοποιείται για την εισαγωγή των δεδομένων στο δίκτυο.

```
X_train = X_train.reshape(-1,IMG_HEIGHT,IMG_WIDTH,10,1)
```

Τελευταία διαδικασία της προεπεξεργασίας αποτελεί η ομαλοποίηση των δεδομένων (`data normalization`). Η διαδικασία αυτή πραγματοποιείται για τους εξής σκοπούς. Αρχικά, το νευρωνικό δίκτυο βαθιάς μάθησης μπορεί να εκπαιδευτεί πιο γρήγορα. Ακόμα, είναι πιο υψηλό το ποσοστό εκμάθησης (`learning rate`), αυξάνοντας έτσι την ταχύτητα εκπαίδευσης του δικτύου, έχοντας λιγότερες αποκλίσεις. Τέλος, γίνεται πιο εύκολη η αρχικοποίηση του βάρους (`Weight initialization`). Με την ομαλοποίηση των δεδομένων μειώνεται η ευαισθησία των αρχικών βαρών εκίνησης.

```
X_train = X_train.astype('float64') / 255.
```

4.3.2. Εκπαίδευση Δικτύου Αυτοκωδικοποιητή

Για την εκπαίδευση του δικτύου χρησιμοποιούνται μόνο δεδομένα τα οποία δεν έχουν πτώση (normal data). Οι εικόνες που περιέχουν πτώση (abnormal data) χρησιμοποιούνται σε επόμενο στάδιο για τον έλεγχο του δικτύου.

Πρώτα πρώτα, καθορίζεται το μέγεθος εισόδου των εικόνων για την αρχικοποίηση παραμέτρων (input_shape). Η είσοδος του δικτύου αποτελείται από έναν πενταδιάστατο τανυστή (5D tensor), καθώς χρησιμοποιούνται δεδομένα βίντεο. Από τα δεδομένα βίντεο αποφασίζεται η δειγματοληψία να είναι δέκα (10) εικόνες ανά δευτερόλεπτο, επομένως ο αριθμός 10 είναι μία από τις πέντε διαστάσεις του τανυστή. Ως πρώτη διάσταση ορίζονται οι συστάδες των εικόνων (batch size), ως δεύτερη και τρίτη οι διαστάσεις της εικόνας. Ως τέταρτη διάσταση, ορίζεται ο αριθμός 10 ($10 \frac{\text{frames}}{\text{second}}$), και τέλος ως πέμπτη διάσταση ο αριθμός καναλιών της εικόνας.

Input = (batch_size,IMG_HEIGHT,IMG_WIDTH,10, IMG_CHANNELS)

Μετά την αρχικοποίηση των δεδομένων εφαρμόζεται το επίπεδο τρισδιάστατης συνέλιξης (3D Convolution Layer- Conv3D). Αυτό το επίπεδο δημιουργεί έναν πυρήνα συνέλιξης ο οποίος συνελίσσεται με το επίπεδο εισόδου και παράγει τις εξόδους του τανυστή. Για την εκτέλεση αυτού του επιπέδου χρειάζονται κάποια ορίσματα (Arguments). Τα επιχειρήματα αυτά περιλαμβάνουν τα φίλτρα (filters), το μέγεθος πυρήνα (kernel_size), τα βήματα (strides), τη διαδικασία padding και τέλος, τη συνάρτηση ενεργοποίησης (activation).

Τα φίλτρα αποτελούνται από ακέραιο αριθμό και καθορίζουν το μέγεθος των φίλτρων εξόδου της συνέλιξης. Το μέγεθος πυρήνα, αποτελείται από μία λίστα τριών ακεραίων αριθμών που καθορίζουν το βάθος, το ύψος και το πλάτος του παραθύρου της τρισδιάστατης συνέλιξης. Επίσης, τα strides αποτελούνται από μία λίστα τριών ακεραίων που καθορίζουν τα βήματα της συνέλιξης και την χωρική διάσταση. Επίσης, η μέθοδος padding ορίζει την τιμή 0 σαν ένα πλαίσιο σε κάθε εικόνα προκειμένου να διατηρούνται σταθερά το ύψος και το πλάτος των εικόνων, όπως κατά την είσοδο. Τέλος, χρειάζεται η συνάρτηση ενεργοποίησης, για κάθε μία από τις συνέλιξεις, η οποία είναι η tanh.

Σε αυτό το επίπεδο εφαρμόζονται τρεις διαδοχικές τρισδιάστατες συνέλιξεις, καλώντας από τη βιβλιοθήκη Keras τη συνάρτηση Conv3D που σκοπό έχουν τη μείωση των διαστάσεων των δεδομένων. Όπως φαίνεται από το απόσπασμα του αλγορίθμου παρακάτω, το ξεκίνημα πραγματοποιείται με μεγαλύτερα μεγέθη φίλτρων, πυρήνα και βημάτων, τα οποία διαδοχικά μειώνονται και μένουν σταθερά. Αρχικά, εφαρμόζονται 128 φίλτρα με μέγεθος πυρήνα $11 \times 11 \times 1$ και βήματα (4,4,1).

```
model.add(Conv3D(filters=128,kernel_size=(11,11,1),strides=(4,4,1),padding='same',  
input_shape=(240,240,10,1),activation='tanh'))
```

```
model.add(Conv3D(filters=64,kernel_size=(5,5,1),strides=(2,2,1),padding='same',activation='tanh'))
```

```
model.add(Conv3D(filters=64,kernel_size=(3,3,1),strides=(2,2,1),padding='same',activation='tanh'))
```

Έπειτα, εφαρμόζεται το επίπεδο δισδιάστατης βραχυπρόθεσμης συνέλιξης (2D Convolutional LSTM layer), το οποίο καλείται από την Keras με την συνάρτηση ConvLSTM2D. Το επίπεδο αυτό είναι παρόμοιο με το απλό επίπεδο LSTM, αλλά οι μετασχηματισμοί εισόδου και οι συνεχείς μετασχηματισμοί εισόδου είναι συνελκτικοί. Τα ορίσματα του επιπέδου αυτού αποτελούνται από τα φίλτρα, το μέγεθος πυρήνα, τα βήματα και τη διαδικασία padding και εισέρχονται τρία νέα ορίσματα, η μέθοδος dropout και recurrent dropout, καθώς και οι ακολουθίες επιστροφής (return sequences). Η μέθοδος dropout αναθέτει ως 0 την έξοδο κάθε κρυφού νευρώνα αδρανοποιώντας ένα ποσοστό των εικονοστοιχείων (pixel) της εικόνας. Η

μέθοδος αυτή χρησιμοποιείται για τη μείωση των σφαλμάτων εκπαίδευσης. Έπειτα, εφαρμόζεται η μέθοδος recurrent dropout, όπου οι τιμές της κυμαίνονται μεταξύ 0 και 1. Αποτελεί κλάσμα των μονάδων προς πτώση για τον γραμμικό μετασχηματισμό του συνεχούς (recurrent) σταδίου. Τέλος, οι ακολουθίες επιστροφής καθορίζουν αν θα επιστρέψει η τελευταία έξοδος στην ακολουθία εξόδου ή στην πλήρη ακολουθία. Το επίπεδο διαστάσεων βραχυπρόθεσμης συνέλιξης εφαρμόζεται τρεις φορές και αξίζει να σημειωθεί το εξής. Στο παρακάτω απόσπασμα κατά την εφαρμογή της δεύτερης βραχυπρόθεσμης συνέλιξης υπάρχουν οι πιο μικρές διαστάσεις του αλγορίθμου εκπαίδευσης. Αυτό συμβαίνει καθώς η συνέλιξη αυτή αποτελεί το bottleneck του αυτοκωδικοποιητή που αναγκάζει μια συμπιεσμένη αναπαράσταση των δεδομένων εισόδου.

```
model.add(ConvLSTM2D(filters=64,kernel_size=(3,3),strides=1,padding='same',dropout=0.4,  
                    recurrent_dropout=0.3,return_sequences=True))  
model.add(ConvLSTM2D(filters=32,kernel_size=(3,3),strides=1,padding='same',dropout=0.3,  
                    return_sequences=True))  
model.add(ConvLSTM2D(filters=64,kernel_size=(3,3),strides=1,return_sequences=True,  
                    padding='same',dropout=0.5))
```

Τελευταίο κομμάτι της αρχιτεκτονικής του δικτύου αποτελεί το επίπεδο τρισδιάστατης μεταθετικής συνέλιξης (Transposed 3D convolution layer), σκοπός του επιπέδου αυτού είναι τα δεδομένα που έχουν προκύψει από τις παραπάνω συνέλιξεις να φτάσουν όσο κοντά γίνεται στην αρχική τους μορφή. Ουσιαστικά, πραγματοποιείται μία διαδικασία αντίστροφη από το πρώτο επίπεδο συνέλιξης και αυτό είναι εύκολα αντιληπτό από το παρακάτω απόσπασμα του αλγορίθμου.

```
model.add(Conv3DTranspose(filters=64,kernel_size=(3,3,1),strides=(2,2,1),padding='same',  
                        activation='tanh'))  
model.add(Conv3DTranspose(filters=128,kernel_size=(5,5,1),strides=(2,2,1),padding='same',  
                        ,activation='tanh'))  
model.add(Conv3DTranspose(filters=1,kernel_size=(1,11,11),strides=(4,4,1),padding='same',  
                        activation='tanh'))
```

Έπειτα από την κατασκευή του δικτύου, γίνεται η μεταγλώττιση (compile) του δικτύου. Η μεταγλώττιση χρησιμοποιεί τα ορίσματα που αφορούν το βελτιστοποιητή (optimizer), την συνάρτηση απώλειας (loss function) και τη μετρική συνάρτηση (metrics). Στην παρούσα εργασία χρησιμοποιείται ο βελτιστοποιητής Adam (Adaptive movement), ο οποίος προσαρμόζει αυτόματα έναν αριθμό εκμάθησης για κάθε μεταβλητή εισόδου για την λειτουργία και εξομαλύνει περαιτέρω τη διαδικασία αναζήτησης, χρησιμοποιώντας μία εκθετικά φθίνουσα μέση τιμή, ώστε να ενημερώνει τις μεταβλητές. Ο Adam optimizer χρησιμοποιείται καθώς συνδυάζει τα πλεονεκτήματα δύο στοχαστικών αλγορίθμων μείωσης κλίσης του Adaptive Gradient Algorithm (AdaGrad) και του Root Mean Square Propagation (RMSProp). Ο ρυθμός εκμάθησης (learning

rate) του Adam είναι το 0.001 [65]. Η συνάρτηση απώλειας που χρησιμοποιείται είναι εκείνη του Μέσου Τετραγωνικού Σφάλματος (Mean Squared Error- MSE). Τελευταία παράμετρο αποτελεί η μετρητική συνάρτηση μετρητική συνάρτηση, η οποία χρησιμοποιείται για να αξιολογήσει την απόδοση του μοντέλου. Οι μετρητικές συναρτήσεις, είναι παρόμοιες με τις συναρτήσεις απώλειας με ειδοποιό διαφορά πως το αποτέλεσμα της μετρητικής συνάρτησης δεν χρησιμοποιείται κατά τη διαδικασία της εκπαίδευσης παρά μόνο για αξιολόγηση του μοντέλου. Στην συγκεκριμένη περίπτωση, ως μετρητική συνάρτηση χρησιμοποιείται ο δείκτης accuracy και mse. Ωστόσο, αξίζει να αναφερθεί πως η μετρητική συνάρτηση accuracy δεν έχει κάποιο νόημα για τον αυτοκωδικοποιητή, είναι χρήσιμη μόνο για τους ταξινομητές.

```
model.compile(optimizer='adam',loss='mean_squared_error',metrics=['accuracy',  
                    'mean_squared_error'])
```

Σε τελευταίο στάδιο, ορίζεται η συνάρτηση προσαρμογής (fit), η οποία εκπαιδεύει το μοντέλο για ένα συγκεκριμένο αριθμό εποχών, στην συγκεκριμένη περίπτωση για 100 εποχές. Στην συνάρτηση αυτή, ορίζονται τα δεδομένα εκπαίδευσης (X_train, Y_train), καθώς και πόσα από αυτά θα χρησιμοποιηθούν για αξιολόγηση, το μέγεθος της επιθυμητής συστάδας εικόνων που θα παίρνει το δίκτυο κάθε φορά (batch_size), πιο συγκεκριμένα ανά δέκα εικόνες, ο αριθμός των επιθυμητών εποχών καθώς και τα callbacks.

```
results = model.fit(X_train, Y_train, validation_split=0.1, batch_size= batch_size, epochs= epochs,  
                    callbacks=[earlystopper, checkpointer])
```

Τα callbacks, αποτελούν κριτήρια που διακόπτουν την προσαρμογή σε περίπτωση που οι παράμετροι σταματήσουν να βελτιώνονται, συγκεκριμένα η συνάρτηση απωλειών για τα δεδομένα αξιολόγησης (validation loss). Αποτελούνται από τις συναρτήσεις Earlystopping και checkpointer, οι οποίες καλούνται από τη βιβλιοθήκη Keras. Η συνάρτηση Earlystopping είναι υπεύθυνη για την διακοπή της προσαρμογής αν για 5 συνεχόμενες εποχές (patience) δεν βελτιώνονται τα αποτελέσματα της εκπαίδευσης, ενώ η ModelCheckpoint είναι υπεύθυνη για την αποθήκευση των καλύτερων βαρών του μοντέλου.

```
earlystopper = keras.callbacks.EarlyStopping(patience=5, verbose=1)
```

```
checkpointer = keras.callbacks.ModelCheckpoint('abnormal_model_thermal_side_camera.h5',  
                    verbose=1, save_best_only=True)
```

Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας με όλους τις παραμέτρους που χρειάζονται για την εκπαίδευση του δικτύου.

Table 2: Συγκεντρωτικός Πίνακας Παραμέτρων Εκπαίδευσης

Συγκεντρωτικός πίνακας παραμέτρων εκπαίδευσης	
Συστάδες εικόνων	10
Ύψος εικόνων	240
Πλάτος εικόνων	240
Βελτιστοποιητής	Adam
Συνάρτηση Ενεργοποίησης	tanh
Συνάρτηση Απωλειών	MSE
Μετρητική Συνάρτηση	MSE
Αριθμός εποχών	100
Earlystopping (patience)	5
ModelCheckpoint	Save best only
Total Parameters	1,142,081
Trainable Parameters	1,142,081

4.3.3. Προβλέψεις του δικτύου Αυτοκωδικοποιητή

Στο σημείο αυτό γίνονται οι προβλέψεις απώλειας του δικτύου, οι οποίες αποτελούνται από την εύρεση του σφάλματος ανακατασκευής (reconstruction error) των normal δεδομένων. Ουσιαστικά, βρίσκονται οι απώλειες του μέσου τετραγωνικού σφάλματος των δεδομένων εκπαίδευσης. Αφού βρεθούν οι προβλέψεις για κάθε εικόνα, υπολογίζεται ο μέσος όρος αυτών, με σκοπό να γίνει πρόβλεψη για όλο το δίκτυο.

```
reconstructions = model.predict(x)
```

```
train_loss = tf.keras.losses.mse(reconstructions, x)
```

```
loss_total_train=np.mean(train_loss)
```

Ακόμα, μέσα από τη μέση τιμή των απωλειών υπολογίζεται ένα κατώφλι, το οποίο χρειάζεται σε επόμενο κεφάλαιο για τον καθορισμό της ύπαρξης πτώσης στις ακολουθίες βίντεο ή όχι. Το κατώφλι αυτό, υπολογίζεται μέσα από τη συνάρτηση τυπικής απόκλισης (standard deviation), της βιβλιοθήκης NumPy.

```
threshold = np.std(train_loss)
```

4.3.4. Έλεγχος του δικτύου Αυτοκωδικοποιητή

Κατά τον έλεγχο του δικτύου του δικτύου, ο αλγόριθμος τροφοδοτείται μόνο με δεδομένα στα οποία υπάρχουν πτώσεις (abnormal data). Σκοπός της διαδικασίας αυτής είναι οι προβλέψεις του δικτύου να είναι πολύ μεγαλύτερες από εκείνες των δεδομένων εκπαίδευσης, με σκοπό να γίνει ο εντοπισμός των ανωμαλιών, δηλαδή της πτώσης.

Η διαδικασία που ακολουθείται είναι ίδια με εκείνη των προβλέψεων του δικτύου με την ειδοποιό διαφορά, πως σε αυτό το κομμάτι χρησιμοποιούνται δεδομένα ελέγχου, που περιέχουν κατά κύριο λόγο πτώσεις.

Κατά τον έλεγχο, οι απώλειες των abnormal δεδομένων, παρουσιάζονται να είναι πολύ μεγαλύτερες σε σχέση με τις απώλειες των normal δεδομένων. Αυτή η αύξηση του σφάλματος, είναι επιθυμητή, καθώς πλέον είναι αντιληπτός ο διαχωρισμός ανάμεσα στα δεδομένα πτώσεων και σε αυτά που δεν έχουν πτώσεις.

```
reconstructions = model.predict(x)
```

```
test_loss = tf.keras.losses.mse(reconstructions, x)
```

```
loss_total_test=np.mean(test_loss)
```

Κεφάλαιο 5^ο: Πειραματική Διαχείριση

Στο κεφάλαιο αυτό γίνεται αναλυτική παρουσίαση των δεδομένων και των πειραματικών αποτελεσμάτων της μεθόδου του Αυτοκωδικοποιητή. Επιπροσθέτως, παρουσιάζεται μία διαφορετική μέθοδος, επιβλεπόμενης ταξινόμησης με τα ίδια δεδομένα. Η μέθοδος αυτή κατασκευάζεται με σκοπό να συγκριθεί με την κύρια μέθοδο της εργασίας, αυτή του Αυτοκωδικοποιητή, ώστε να βρεθεί η πιο κατάλληλη διαδικασία σε σχέση με τα δεδομένα και την αντιμετώπιση του προβλήματος.

5.1. Περιγραφή των Δεδομένων

Με σκοπό να εκπαιδευτεί και να αξιολογηθεί η προτεινόμενη μέθοδος, διεξήχθη μία πτώση ενός ανδρείκελου (ομοιώματος), με διαστάσεις ανθρώπου, από το μπαλκόνι ενός αρκετά ψηλού κτηρίου. Πιο συγκεκριμένα, το ανδρείκελο ζύγιζε τριάντα κιλά και εκτελούσε πτώση από, κατά προσέγγιση, είκοσι μέτρα ύψος, το οποίο είναι περίπου ισοδύναμο με δύο δευτερόλεπτα ελεύθερης πτώσης.

Για τις ανάγκες του πειράματος, πραγματοποιήθηκαν 320 δοκιμαστικές πτώσεις του ανδρείκελου, με σκοπό την προσομοίωση της πτώσης. Επιπροσθέτως, καταγράφηκαν αρκετά βίντεο χωρίς την πτώση του ανδρείκελου, που αποτελούν τα κανονικά δεδομένα.

Τα πειράματα, έλαβαν χώρα στην ευρύτερη περιοχή του Ολυμπιακού Γυμναστηρίου Άρσης Βαρών Νίκαιας και διήρκεσαν πέντε ημέρες. Καθώς η χρονική διάρκεια των δοκιμαστικών πτώσεων συνεχιζόταν όλη τη διάρκεια της ημέρας, από τις 9:00 π.μ. έως τις 5:00 μ.μ., τα επίκτητα βίντεο έχουν αλλαγές όσο αφορά τις συνθήκες φωτισμού. Επίσης, τα πειράματα διεξήχθησαν κάτω από πληθώρα καιρικών συνθηκών, όπως λιακάδα, συννεφιά, βροχή, με άνεμο, ζέστη και κρύο, επομένως παρήχθησαν περισσότερες διακυμάνσεις στο παρασκήνιο των γεγονότων.

Τα δεδομένα βίντεο συγκεντρώθηκαν με ανάλυση 1080×1920 pixels και χρησιμοποιήθηκε η φωτογραφική κάμερα GoPro Hero 7 Silver. Η κάμερα έχει οριστεί να γυρνάει βίντεο με υψηλό αριθμό καρτέ (εικόνες), συγκεκριμένα πενήντα εικόνες το δευτερόλεπτο, για να εξασφαλίσει την απόκτηση επαρκών δεδομένων που αφορούν το κρίσιμο συμβάν.

Οι αισθητήρες των καμερών τοποθετούνται σε τέσσερα διαφορετικά σημεία του κτηρίου, με σκοπό την απόκτηση δεδομένων που ποικίλλουν όσο αφορά το παρασκήνιο, τον φωτισμό τη γωνία λήψης και της απόσταση. Στην παρούσα εργασία τα δεδομένα χωρίζονται με βάση τη γωνία λήψης και πιο συγκεκριμένα αν η γωνία λήψης είναι οριζόντια ή γωνία λήψης 45°.

Ο αλγόριθμος εκπαιδεύεται με οπτικά (RGB) και θερμικά (thermal) δεδομένα με βάση τις γωνίες λήψης που αναφέρονται παραπάνω.

5.1.1. Οπτικά Δεδομένα (RGB Data)

Για τα οπτικά δεδομένα, πάρθηκαν συνολικά τριάντα έξι βίντεο, τα οποία χωρίστηκαν σύμφωνα με δύο μεταβλητές. Η πρώτη μεταβλητή είναι η γωνία λήψης της κάμερας (45° ή οριζόντια) και η δεύτερη μεταβλητή είναι αν υπάρχει πτώση στο βίντεο ή όχι. Πιο συγκεκριμένα, τα δεδομένα διαμορφώνονται ως εξής. Πρώτα, κατά τη γωνία λήψης 45°, υπάρχουν τρία (3) βίντεο με πτώση και πέντε (5) βίντεο χωρίς πτώση. Έπειτα, για την οριζόντια γωνία λήψης υπάρχουν δεκατρία (13) βίντεο με πτώση και δεκαπέντε (15) χωρίς πτώση.

Table 3: Κατανομή των Οπτικών Δεδομένων

Οπτικά Δεδομένα	Γωνία Λήψης 45°	Οριζόντια Γωνία Λήψης
Χωρίς πτώση	3	13
Με Πτώση	5	15

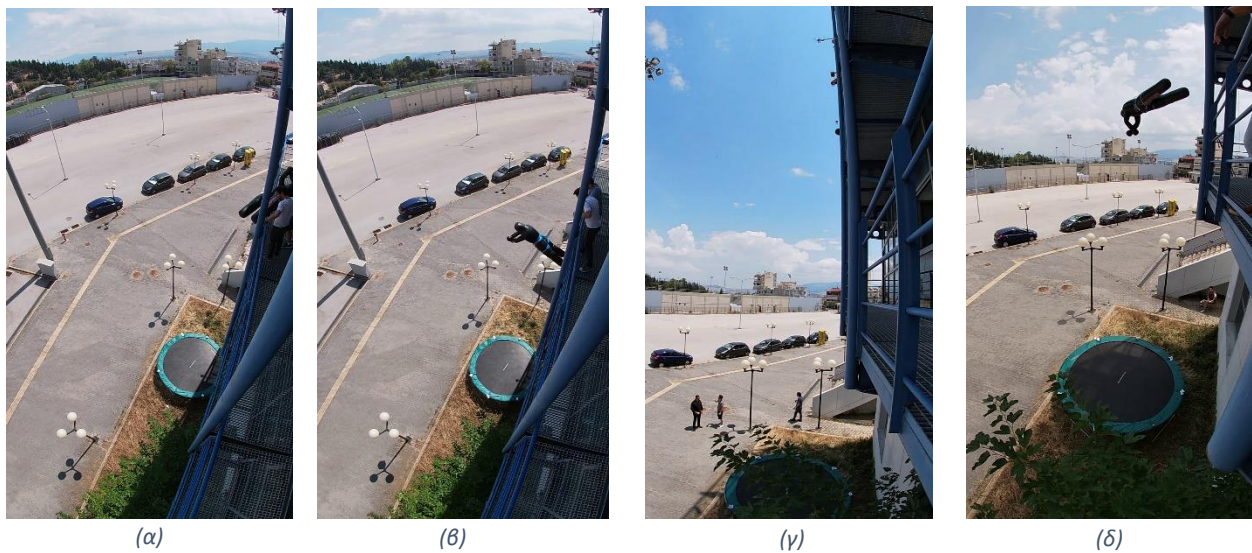


Figure 10: Δείγματα εικόνων γωνία λήψης 45° και οριζόντιας γωνίας λήψης για οπτικά δεδομένα

Στις εικόνες 10 (α) και (β) φαίνονται δείγματα που έχουν διεξαχθεί από την γωνία λήψης 45° της κάμερας, συγκεκριμένα στην (α) δεν εκτελείται πτώση, αντίθετα στην (β) υπάρχει πτώση. Οι εικόνες 10 (γ) και (δ) ανήκουν στα δείγματα της οριζόντιας λήψης της κάμερας όπου στην (γ) δεν υπάρχει πτώση, ενώ η (δ) διακρίνεται από την πτώση του ανδρείκελου.

5.1.2. Θερμικά Δεδομένα (Thermal Data)

Τα θερμικά δεδομένα αποτελούνται από τριάντα έξι (36) βίντεο, τα οποία ομαδοποιήθηκαν σύμφωνα με τη γωνία λήψης του αισθητήρα και στη συνέχεια σύμφωνα με το περιεχόμενό τους, δηλαδή αν εμπεριέχουν πτώση, είτε δεν εμπεριέχουν. Όσο αφορά την γωνία λήψης 45° της κάμερας, η ομάδα αυτή διαθέτει συνολικά οκτώ (8) βίντεο, εκ των οποίων τα τρία (3) έχουν πτώση, ενώ τα πέντε (5) δεν έχουν. Ακόμα, η ομάδα της οριζόντιας γωνίας λήψης, διαθέτει είκοσι οκτώ (28) βίντεο, εκ των οποίων στα δεκατρία (13) παρουσιάζεται πτώση, αλλά στα δεκαπέντε (15) από αυτά, δεν παρουσιάζεται.

Table 4: Κατανομή Θερμικών Δεδομένων

Θερμικά Δεδομένα	Γωνία Λήψης 45°	Οριζόντια Γωνία Λήψης
Χωρίς πτώση	3	13
Με Πτώση	5	15

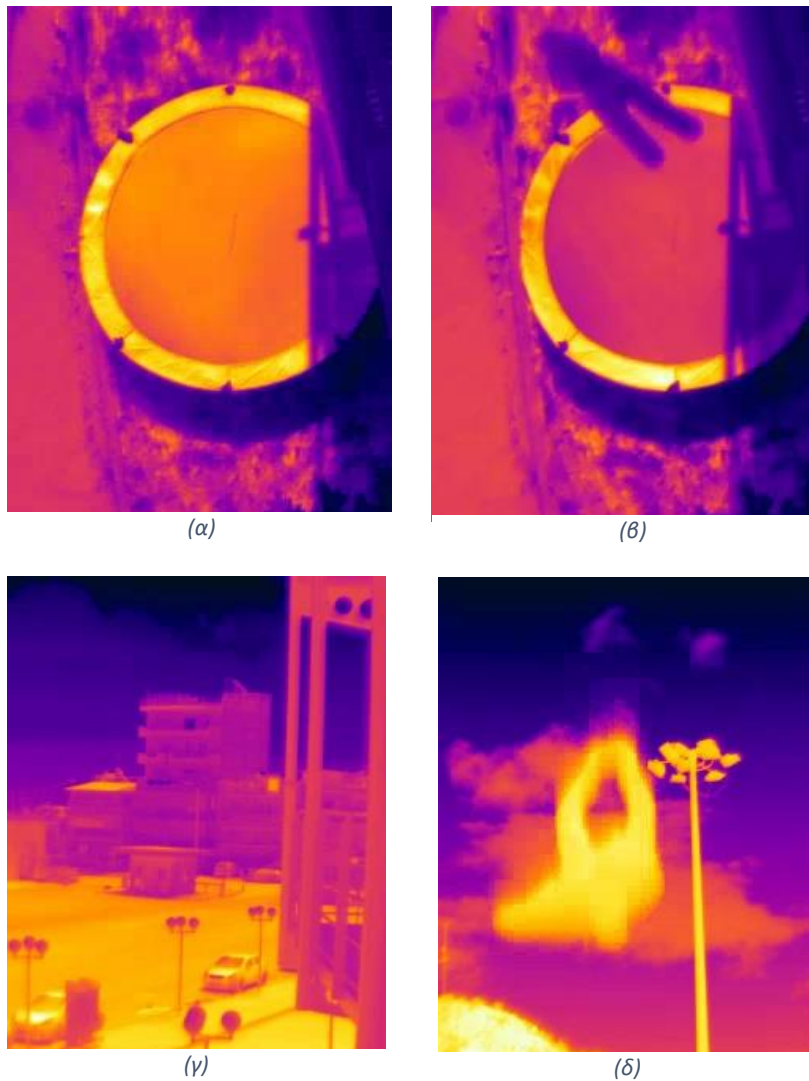


Figure 11: Δείγματα εικόνων γωνίας λήψης 45° και οριζόντιας γωνίας λήψης για θερμικά

Στις εικόνες 11 (α) και (β) φαίνονται δείγματα που έχουν διεξαχθεί από την γωνία λήψης 45° του αισθητήρα, συγκεκριμένα στην (α) δεν εκτελείται πτώση, αντίθετα στην (β) υπάρχει πτώση. Οι εικόνες 11 (γ) και (δ) ανήκουν στα δείγματα της οριζόντιας λήψης του αισθητήρα, όπου στην (γ) δεν υπάρχει πτώση, ενώ η (δ) διακρίνεται από την πτώση του ανδρείκελου.

5.2. Πειραματικές Εκτιμήσεις

Για τις ανάγκες της εργασίας και σύμφωνα με τον παραπάνω διαχωρισμό των δεδομένων, δημιουργούνται τέσσερις αλγόριθμοι συνελκτικών Αυτοκωδικοποιητών, ανάλογα με τις γωνίες λήψης της κάμερας και τη φύση των δεδομένων (θερμικά, οπτικά). Επιπροσθέτως, κατασκευάζεται ένας ταξινομητής με τεχνικές επιβλεπόμενης μάθησης, με σκοπό να συγκριθεί με την κύρια μέθοδο της εργασίας (του συνελκτικού αυτοκωδικοποιητή) και να αποδοθεί η πιο αρμοστή μέθοδος, όσο αφορά το πρόβλημα που καλείται να αντιμετωπίσει.

5.2.1. Area under the curve of the Receiver operating characteristic (ROC AUC)[66]

Η περιοχή κάτω από την καμπύλη των χαρακτηριστικών λειτουργιών του δέκτη (Area under the curve of the Receiver operating characteristic–ROC AUC), χρησιμοποιείται ως μετρητική αξιολόγηση απόδοσης της προτεινόμενης μεθόδου και των αντιστοίχων. Η AUC υπολογίζεται σε σχέση με τις παρατηρήσεις αληθινής βάσης σε επίπεδο εικόνων και αποτελεί μία κοινή μέτρηση για πολλές μεθόδους ανίχνευσης μη φυσιολογικών συμβάντων. Υπολογίζει την ικανότητα του αλγορίθμου μάθησης να ξεχωρίσει φυσιολογικά από μη φυσιολογικά γεγονότα και κάνει μία σύνοψη της καμπύλης ROC του συστήματος, όπου αποτελεί την πιθανοτική καμπύλη που επισημαίνει την ανύψωση μίας πραγματικής ειδοποίησης (true positive) και μίας ψευδούς ειδοποίησης (false positive) σε μία πληθώρα κατωφλίων. Όσο υψηλότερη είναι η AUC, τόσο καλύτερα μπορεί το μοντέλο να ξεχωρίσει αν υπάρχει πτώση ή όχι. Σε ένα μοντέλο που έχει εξαιρετικό μέτρο διαχωρισμού, η AUC είναι ίση με 1. Αντίθετα, ένα μοντέλο που δεν μπορεί να διαχωρίσει τα δύο γεγονότα μεταξύ τους έχει AUC ίση με 0.5.

Η καμπύλη ROC παράγεται υπολογίζοντας τον ρυθμό των πραγματικά θετικών περιπτώσεων (True Positive rate-TPR) ή αλλιώς της ευαισθησίας (Sensitivity) του δικτύου και των ρυθμό των ψευδώς θετικών περιπτώσεων (False Positive rate-FPR), σύμφωνα με το κατώφλι που έχει οριστεί κατά τις προβλέψεις των δεδομένων χωρίς πτώση των δικτύων.

$$TPR = \text{Sensitivity} = \frac{TP}{TP + FN}$$

Όπου TP είναι οι πραγματικά θετικές περιπτώσεις και FN (False Negatives), οι πραγματικά αρνητικές περιπτώσεις.

$$FPR = 1 - \text{Specificity} = \frac{FP}{FP + TN}$$

Όπου FP είναι οι ψευδώς αρνητικές περιπτώσεις και TN οι πραγματικά αρνητικές περιπτώσεις. Ο δείκτης FPR είναι ουσιαστικά το μέτρο του κάθε πότε μία πραγματικά αρνητική μεταβλητή θα ταξινομηθεί λαθεμένα, ως θετική.

Παρακάτω παρατίθενται οι γραφικές παραστάσεις των ROC AUC για τους τέσσερις διαχωρισμούς των δεδομένων. Στα δύο πρώτα γραφήματα, παρουσιάζονται οι παραστάσεις για τα θερμικά δεδομένα. Συγκεκριμένα, στην εικόνα 12 (α), παρουσιάζεται η γραφική παράσταση για τα δεδομένα οριζόντιας γωνίας λήψης της κάμερας, ενώ στην 12 (β) η γραφική παράσταση για τα δεδομένα από τη γωνία λήψης 45°.

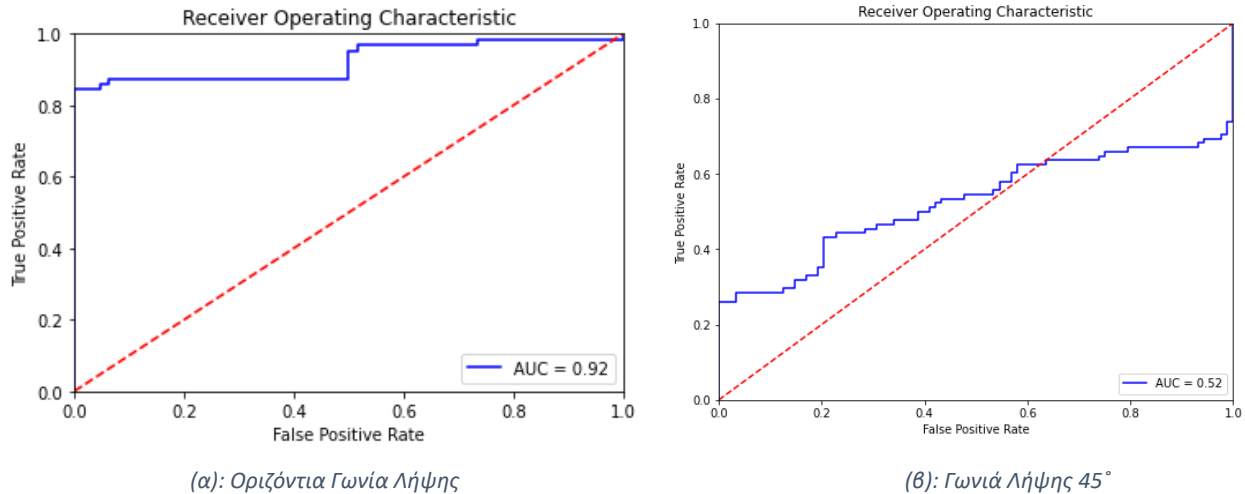


Figure 12: Γραφήματα AUC για τα θερμικά δεδομένα

Όπως φαίνεται στις παραπάνω εικόνες, ο αλγόριθμος συνελκτικού αυτοκωδικοποιητή που δημιουργείται με δεδομένα από την οριζόντια γωνία λήψης, έχει πολύ καλή απόδοση ίση με 0.92 που είναι πολύ κοντά στο 1. Το γεγονός αυτό σημαίνει πως ο αλγόριθμος λειτουργεί σωστά και μπορεί να ξεχωρίσει τα δεδομένα που έχουν πτώση έναντι αυτών που δεν έχουν με ποσοστό ίσο του 92%.

Στο γράφημα 12 (β), το μοντέλο δεν έχει καλή απόδοση και μάλιστα είναι ίση με 0.52. Επομένως, ο αλγόριθμος δεν μπορεί να ξεχωρίσει σε ποιες εικόνες υπάρχει πτώση και σε ποιες όχι ή μπορεί να τις ξεχωρίσει σε ένα ποσοστό 52%, το οποίο δεν καλύπτει τις ανάγκες του προβλήματος.

Στο σημείο αυτό γίνεται μετάβαση στις γραφικές παραστάσεις ROC AUC των οπτικών δεδομένων RGB. Συγκεκριμένα, στην εικόνα 13 (α) παρουσιάζεται η γραφική παράσταση των δεδομένων που αφορούν την οριζόντια γωνία λήψης της κάμερας, ενώ στην 13 (β), τα δεδομένα που αφορούν την γωνία λήψης 45°.

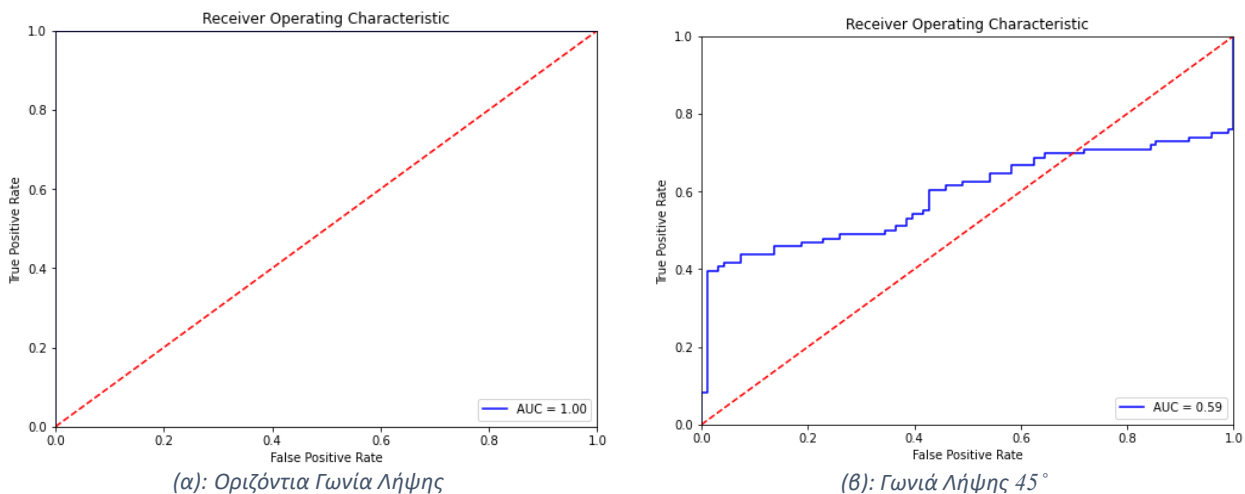


Figure 13: Γραφική Παράσταση AUC για τα οπτικά δεδομένα

Στην περίπτωση των οπτικών δεδομένων, φαίνεται πως το μοντέλο για τα δεδομένα από την οριζόντια λήψη της κάμερας, 13 (α), είναι ικανό να κάνει την εξαιρετική ταξινόμηση με ίση με 1 και να διαχωρίσει πλήρως τις εικόνες όπου υπάρχει πτώση σε σχέση με τις υπόλοιπες.

Αντίθετα, στην εικόνα 13 (β), φαίνεται πως η ταξινόμηση δεν είναι τόσο ακριβής. Η απόδοσή της είναι ίση με 0.59. Δηλαδή το μοντέλο του συνελκτικού αυτοκωδικοποιητή έχει 59% πιθανότητες να ξεχωρίσει τις δύο κλάσεις μεταξύ τους. Η απόδοση αυτή δεν είναι καλή, καθώς όσο πιο κοντά στο ποσοστό 50% βρίσκεται ένα μοντέλο, τόσο δυνατό είναι να ταξινομούνται λάθος τα δεδομένα.

Γενικά είναι αντιληπτό πως και στα οπτικά αλλά και στα θερμικά δεδομένα, οι αλγόριθμοι που τροφοδοτούνται με εικόνες από την οριζόντια γωνία λήψης της κάμερας, έχουν πολύ καλύτερη απόδοση σε σχέση με τις εικόνες από την γωνία λήψης 45°. Ακόμα, τα οπτικά δεδομένα, φαίνεται να έχουν καλύτερη απόδοση σε σχέση με τα θερμικά, ωστόσο έχουν πολύ μικρές διαφορές.

5.2.2. Δείκτες αξιολόγησης

Για την αξιολόγηση των δικτύων χρησιμοποιούνται και κάποιοι μετρητικοί δείκτες. Αποτελούνται από τον δείκτη accuracy, τον δείκτη recall, precision score και F1 score. Για να βρεθούν οι δείκτες αυτοί, είναι απαραίτητο να υπολογιστούν οι τιμές True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) αντίστοιχα. Οι τιμές αυτές μπορούν να ληφθούν από τον Πίνακα Σύγχυσης (Confusion Matrix), όπου σε όρους μη επιβλεπόμενης ταξινόμησης ονομάζεται συνήθως Πίνακας Αντιστοίχισης (Matching Matrix). Κάθε σειρά του πίνακα αντιπροσωπεύει την πραγματική κλάση, ενώ κάθε στήλη αντιπροσωπεύει την προβλεπόμενη κλάση. Ο πίνακας αποτελείται από τέσσερις διαφορετικούς συνδυασμούς των πραγματικών και προβλεπόμενων κλάσεων, ο οποίος παρουσιάζεται παρακάτω.

5.2.2.1. Κατανόηση των τιμών του Πίνακα Σύγχυσης

True Positive (TP): Το δίκτυο προβλέπει πως υπάρχει πτώση και πραγματικά υπάρχει

True Negative (TN): Το δίκτυο προβλέπει πως δεν υπάρχει πτώση και πραγματικά δεν υπάρχει

False Positive (FP): Το δίκτυο προβλέπει πως υπάρχει πτώση, αλλά δεν υπάρχει

False Negative (FN): Το δίκτυο προβλέπει πως δεν υπάρχει πτώση, αλλά υπάρχει

Table 5: Πίνακας Σύγχυσης

Πίνακας Σύγχυσης		Πραγματικές Αξίες	
Προβλεπόμενες Αξίες	Positive (1)	TP	FP
	Negative (0)	FN	TN

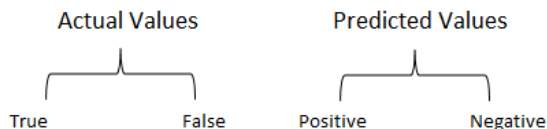


Figure 14: Κατανόηση των τιμών του Πίνακα Σύγχυσης

5.2.2.2. Δείκτης Ακρίβειας (*Accuracy Score*)

Ο δείκτης ακρίβειας αποτελεί έναν από τους πιο συνηθισμένους μετρητικούς δείκτες. Είναι το μέτρο όλων των σωστά προσδιορισμένων παρατηρήσεων προς όλες τις παρατηρήσεις. Συνήθως χρησιμοποιείται όταν οι κλάσεις έχουν την ίδια βαρύτητα σημασίας.

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

5.2.2.3. Δείκτης Precision

Ο δείκτης αυτός αποτελεί τον λόγο των πραγματικά θετικών περιπτώσεων προς το σύνολο όλων των θετικών προβλέψεων. Ο δείκτης Precision είναι ένα καλό μέτρο για να καθοριστεί αν το ποσοστό των False Positive κλάσεων είναι υψηλό. Δηλαδή πόσα δεδομένα έχουν εσφαλμένα θεωρηθεί από το δίκτυο πως περιέχουν πτώση.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Total\ Predicted\ Positive}$$

5.2.2.4. Δείκτης Recall

Ο μετρητικός αυτός δείκτης εκφράζει τον λόγο των πραγματικά θετικών παρατηρήσεων προς τον λόγο όλων των παρατηρήσεων που έχουν προβλεφθεί να ανήκουν σε πραγματικά θετικές περιπτώσεις. Ουσιαστικά αποτελεί τις κλάσεις των πραγματικών πτώσεων προς το σύνολο των πραγματικών πτώσεων και τις περιπτώσεις που δεν έχει προβλεφθεί πτώση, αλλά στην πραγματικότητα υπάρχει.

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

5.2.2.5. Δείκτης F1

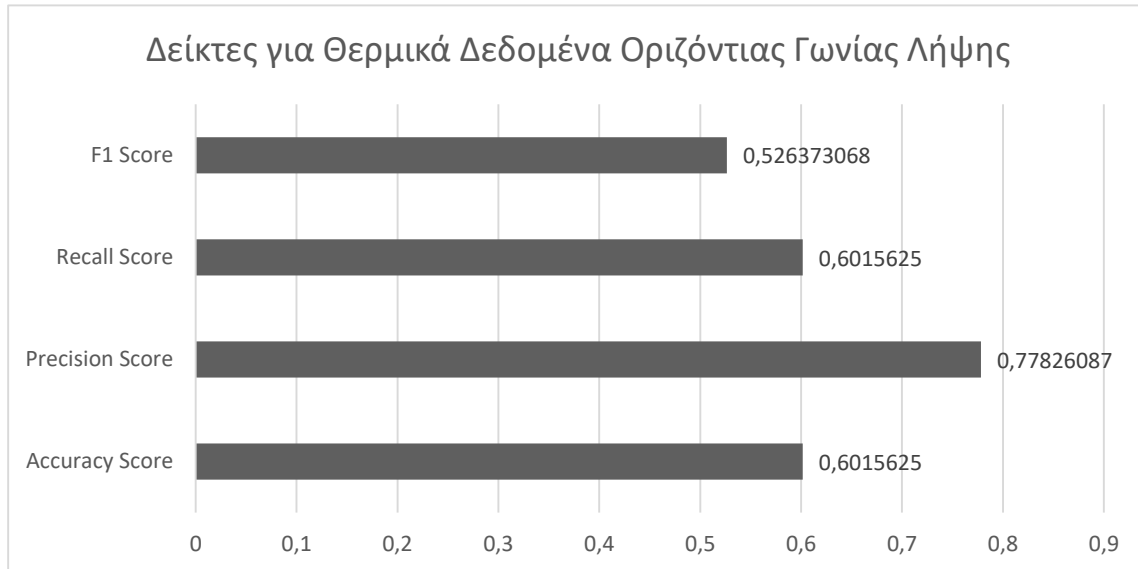
Ο δείκτης αυτός αποτελεί μία αρμονική μέση τιμή των δεικτών Precision και Recall και υπολογίζει πολύ καλύτερα τις λαθεμένες ταξινομήσεις από τον δείκτη της ακρίβειας (accuracy score).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

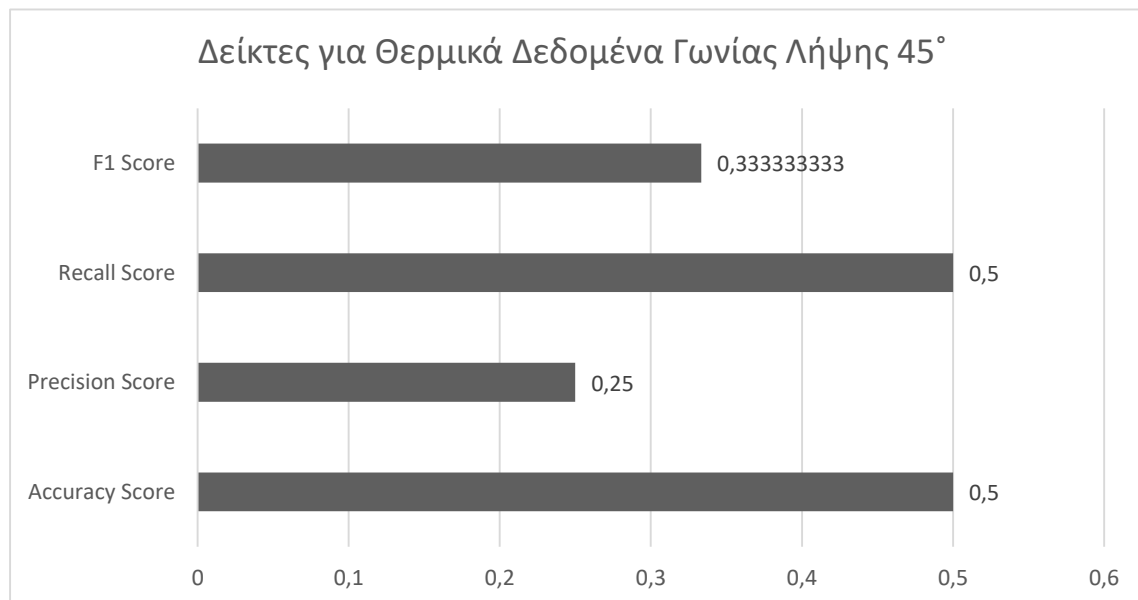
5.2.3. Παρουσίαση Δεικτών Αξιολόγησης για το Δίκτυο του Αυτοκωδικοποιητή

Οι μετρητικοί δείκτες υπολογίζονται για το σύνολο των προβλέψεων των δεδομένων σύμφωνα με τον διαχωρισμό που αναφέρεται προηγουμένως δηλαδή, σύμφωνα με τη γωνία λήψης του αισθητήρα και με τη φύση των δεδομένων, θερμικά και οπτικά. Παρακάτω παρουσιάζονται τα γραφήματα των δεικτών αυτών.

Για τις θερμικές εικόνες

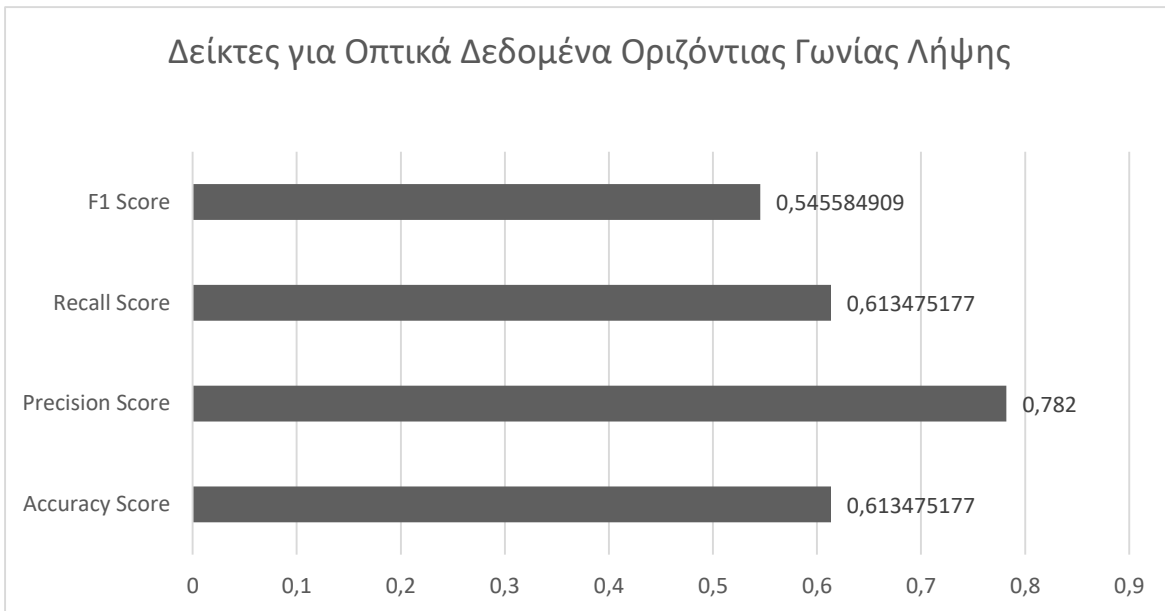


Γράφημα 1: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Οριζόντιας Γωνίας Λήψης

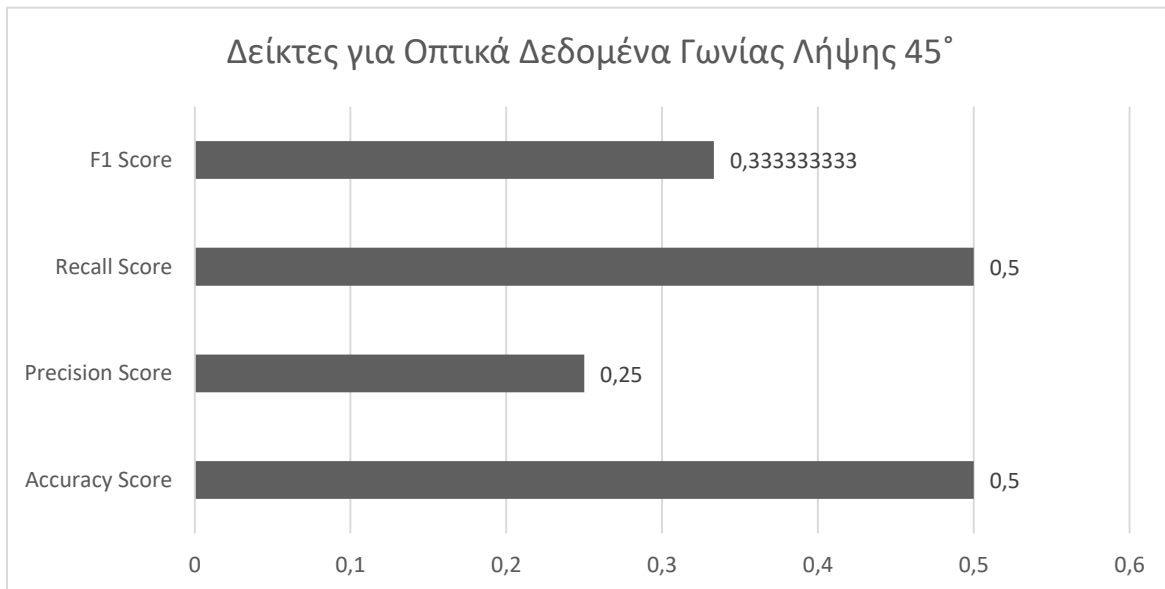


Γράφημα 2: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Γωνίας Λήψης 45°

Για τις Οπτικές εικόνες



Γράφημα 3: Γράφημα Μελέτης για τα Οπτικά Δεδομένα Οριζόντιας Γωνίας Λήψης



Γράφημα 4: Γράφημα Μελέτης για τα Οπτικά Δεδομένα Γωνίας Λήψης 45°

Γενικά, όσο αφορά τη φύση των δεδομένων, δεν φαίνεται να υπάρχει κάποια σημαντική αλλαγή των μέτρων των δεικτών. Οι διαφορές τους παρουσιάζονται να είναι αμελητέες. Ωστόσο, στον διαχωρισμό μεταξύ των γωνιών λήψης, φαίνεται πως η οριζόντια οπτική υπερισχύει κατά μεγάλο ποσοστό έναντι της οπτικής των 45° .

Table 6: Αξιολόγηση Εκπαίδευσης δικτύου για τα Δεδομένα του Συνελκτικού Αυτοκωδικοποιητή

Υπολογιστικό Κόστος για τον Συνελκτικό Αυτοκωδικοποιητή				
	Θερμικά Δεδομένα Οριζόντιας Γωνίας Λήψης	Θερμικά Δεδομένα Γωνίας Λήψης 45°	Δεδομένα RGB Οριζόντιας Γωνίας Λήψης	Δεδομένα RGB Γωνίας Λήψης 45°
Υπολογιστική Ισχύς (hh:mm:ss)	0:11:06	0:19:29	0:07:14	0:09:34
Αριθμός Εποχών	16	17	21	24
Loss Fuction (MSE)	0.0165	0.0091	0.0613	0.0449
Metric (MSE)	0.0165	0.0091	0.0613	0,0449
Validation Loss	0.01275	0.00961	0.06441	0.04501

Στον παραπάνω πίνακα, παρουσιάζεται η απόδοση του Συνελκτικού Αυτοκωδικοποιητή σύμφωνα με τον διαχωρισμό των δεδομένων. Φαίνεται πως ο αλγόριθμος, έχει πολύ καλή απόδοση. Η απόδοση του δικτύου κρίνεται κυρίως από την τιμή που έχει η συνάρτηση απωλειών (MSE), η οποία είναι πολύ χαμηλή και στις τέσσερις περιπτώσεις και οι κατώτατες τιμές της έγκεινται κατά την εκπαίδευση των θερμικών δεδομένων. Ο στόχος της εκπαίδευσης του αυτοκωδικοποιητή είναι να ελαχιστοποιηθεί η απώλεια. Όσο ελαχιστοποιείται η απώλεια, τόσο είναι δυνατόν να ανακατασκευαστεί η είσοδος του αλγορίθμου ακριβέστερα.

Όσο αφορά τις τιμές των εποχών, διαφοροποιούνται ανάλογα με το πλήθος των δεδομένων που χρήζουν εκπαίδευσης. Οι εποχές φαίνεται να διαφοροποιούνται σύμφωνα με την διαφορετική γωνία λήψης των καμερών. Ακόμα, φαίνεται πως η υπολογιστική ισχύς, δεν έχει μεγάλες αποκλίσεις μεταξύ των διαφορετικών δεδομένων.

5.3. Κατασκευή Συνελκτικού Ταξινομητή Μάκρο-Βραχυπρόθεσμης Μνήμης (LSTM)

Για σκοπούς συγκρίσεως με την κύριο μέθοδο που ακολουθεί η εργασία, αυτή της μη επιβλεπόμενης μάθησης, κρίθηκε αναγκαίο να κατασκευαστή αλγόριθμος επιβλεπόμενης ταξινόμησης με τα ίδια δεδομένα. Ο αλγόριθμος που κατασκευάζεται αποτελεί έναν Συνελκτικό Ταξινομητή Μάκρο-Βραχυπρόθεσμης Μνήμης (Convolutional LSTM-CONV LSTM) και ακολουθεί τη μέθοδο των συνελκτικών νευρωνικών δικτύων (CNN). Στο υποκεφάλαιο αυτό, παρουσιάζεται αναλυτικά η αρχιτεκτονική του αλγορίθμου, η εκπαίδευση του δικτύου καθώς και τα πειραματικά του αποτελέσματα.

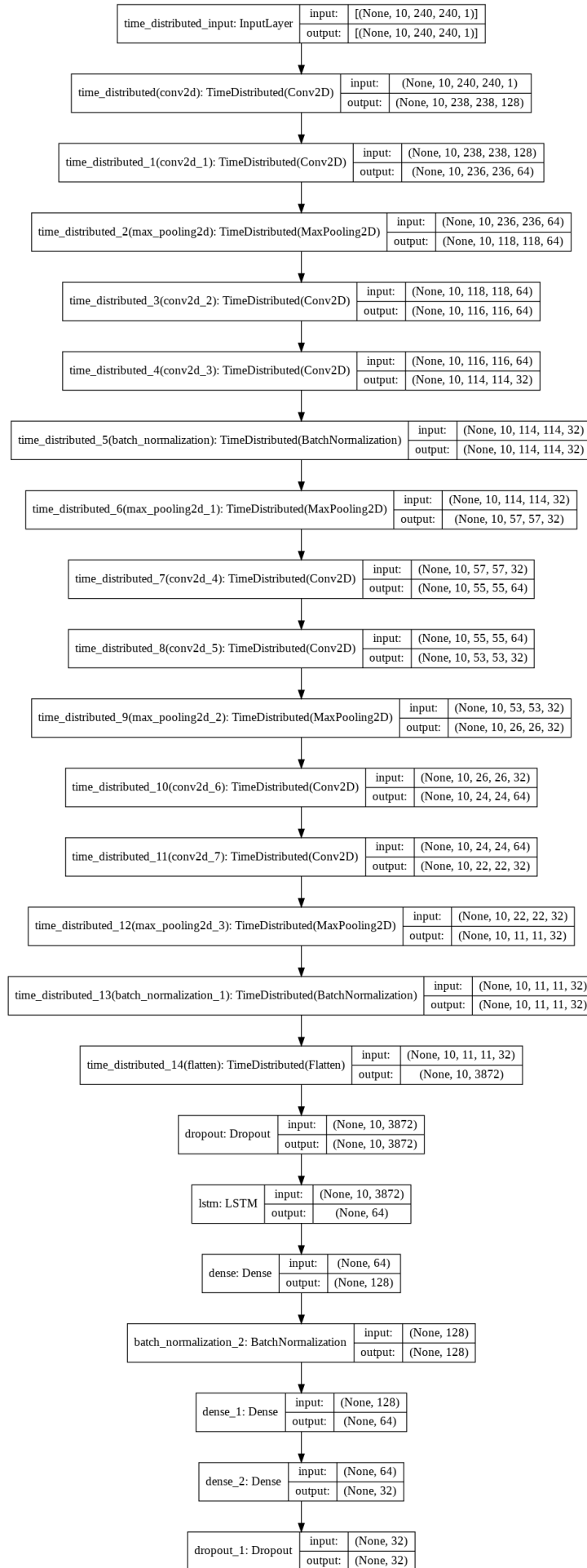
5.3.1. Αρχιτεκτονική Συνελκτικού Ταξινομητή

Ξεκινώντας, εφαρμόζεται το επίπεδο TimeDistributed, που καλείται από την βιβλιοθήκη Keras. Το επίπεδο αυτό είναι πολύ χρήσιμο για εργασία με δεδομένα χρονικών σειρών, όπως τα βίντεο. Επιτρέπει τη χρήση ενός επιπέδου για κάθε είσοδο, δηλαδή μπορεί να χρησιμοποιηθεί ένα μοντέλο που εφαρμόζεται σε κάθε είσοδο [67].

Έπειτα, εφαρμόζεται δύο φορές το επίπεδο δισδιάστατης συνέλιξης Conv2D, το οποίο επίσης καλείται από την Keras. Το επίπεδο αυτό χρησιμοποιεί τη συνάρτηση ενεργοποίησης ReLU. Και στη συνέχεια εφαρμόζεται μέγιστη υποδειγματοληψία (Max Pooling). Η διαδικασία αυτή πραγματοποιείται δύο φορές. Μετά τη διαδικασία της τελευταίας υποδειγματοληψίας εφαρμόζεται το επίπεδο Batch Normalization το οποίο ομαλοποιεί τα δεδομένα εξόδου του χρησιμοποιώντας μέση και τυπική απόκλιση για τα δεδομένα εισόδου του. Στη συνέχεια εφαρμόζεται το επίπεδο Flatten, ώστε να μετατρέψει την πληροφορία που δέχεται σε μονοδιάστατη για να τροφοδοτήσει το επίπεδο LSTM.

Αφού διεκπεραιωθεί η επαναλαμβανόμενη διαδικασία των δισδιάστατων συνέλιξεων και μέγιστων υποδειγματοληψιών, εισέρχεται στο δίκτυο το επίπεδο LSTM. Το οποίο στη συνέχεια περνάει από επαναλαμβανόμενα επίπεδα Dense. Το επίπεδο Dense, είναι ένα συνελκτικό νευρωνικό επίπεδο που είναι βαθιά συνδεδεμένο. Αυτό σημαίνει πως κάθε νευρώνας του επιπέδου αυτού λαμβάνει εισόδους από όλους τους νευρώνες του προηγούμενου επιπέδου.

Η Αρχιτεκτονική του Συνελκτικού Ταξινομητή παρουσιάζεται στην επόμενη σελίδα.



5.3.2. Προεπεξεργασία Δεδομένων του Συνελκτικού Ταξινομητή

Αρχικά, τα δεδομένα βίντεο «διαβάζονται» από τον αλγόριθμο με την βοήθεια της βιβλιοθήκης OS που προσφέρει η γλώσσα προγραμματισμού Python. Έπειτα, ακολουθεί μία διαδικασία επισήμανσης, δίνεται σε κάθε βίντεο μία ετικέτα, η οποία αναφέρει αν ανήκει σε δεδομένα που δεν περιέχουν πτώση (normal data), είτε σε αυτά που περιέχουν (abnormal data). Ύστερα, τα δεδομένα διαχωρίζονται σε δεδομένα εκπαίδευσης (train data) και έλεγχου (test data). Στο σημείο αυτό σημειώνεται πως στον συγκεκριμένο αλγόριθμο επιβλεπόμενης μάθησης τα δεδομένα για την εκπαίδευση και τον έλεγχο του δικτύου είναι μεικτά, δηλαδή αποτελούνται από δεδομένα βίντεο που δεν περιέχουν πτώση και από δεδομένα που περιέχουν. Για την εκπαίδευση χρησιμοποιείται το 70% των συνολικών δεδομένων, ενώ για τον έλεγχο χρησιμοποιείται το 30%. Αφού γίνει η διαδικασία του διαχωρισμού.

Αφού πραγματοποιηθεί η διαδικασία της επισήμανσης, γίνεται εξαγωγή των εικόνων από τις ακολουθίες βίντεο και μετατροπή αυτών σε κλίμακα του γκρι για να μειωθούν οι διαστάσεις των καναλιών. Ακόμα, από τα δεδομένα εκπαίδευσης κρατείται ένα ποσοστό 10% για αξιολόγηση (validation data). Ακολουθεί ένας συγκεντρωτικός πίνακας κατάτμησης των δεδομένων.

Table 7: Συγκεντρωτικός Πίνακας Διαμόρφωσης Δεδομένων

Πίνακας Δεδομένων	
Δεδομένα Εκπαίδευσης	60%
Δεδομένα Ελέγχου	30%
Δεδομένα Αξιολόγησης	10%

Αφού γίνει ο διαχωρισμός των δεδομένων, γίνεται αλλαγή του μεγέθους των δεδομένων αυτών με διαστάσεις 240×240, με σκοπό να μειωθούν οι διαστάσεις και να διατηρηθεί η απαραίτητη πληροφορία. Χρησιμοποιείται η τεχνική ANTIALIAS, με σκοπό να αφαιρεθούν οι παράγοντες του σήματος που έχουν μεγαλύτερη συχνότητα από αυτήν που μπορεί να επιλυθεί σωστά από την κάμερα.

```
image=image.resize((240, 240), Image.ANTIALIAS)
```

Έπειτα, τα δεδομένα εισέρχονται σε έναν πίνακα npy και στη συνέχεια πραγματοποιείται ομαλοποίηση αυτών (data normalization), με σκοπό να μειωθεί η ευαισθησία των αρχικών βαρών εκκίνησης.

```
datu=np.asarray(image)
```

```
normu_dat=datu/255.
```

```
vid_data.append(normu_dat)
```

Εν συνεχεία, οι πίνακες πηρ ανασηματίζονται σε πενταδιάστατους ταυυστές (5D tensors) για να τροφοδοτήσουν το δίκτυο εκπαίδευσης.

```
test_dataset_new=(batch_size,IMG_HEIGHT,IMG_WIDTH,10, IMG_CHANNELS)  
train_dataset_new=(batch_size,IMG_HEIGHT,IMG_WIDTH,10, IMG_CHANNELS)  
validation_dataset_new=(batch_size,IMG_HEIGHT,IMG_WIDTH,10, IMG_CHANNELS)
```

5.3.3. Κατασκευή Μοντέλου Ταξινόμητῆ

Μετά την αρχικοποίηση των δεδομένων εφαρμόζεται το επίπεδο TimeDistributed και πάνω σε αυτό το επίπεδο δισδιάστατης συνελίξης (2D Convolutional Layer–Conv2D), όπου διαδραματίζονται δύο διαδοχικές συνελίξεις. Για την εκτέλεση του επιπέδου αυτού χρειάζονται κάποια ορίσματα. Τα ορίσματα αυτά περιλαμβάνουν τα φίλτρα (filters), το μέγεθος του πυρήνα (kernel size) και τέλος τη συνάρτηση ενεργοποίησης (activation function). Αρχικά, εφαρμόζονται 128 φίλτρα με μέγεθος πυρήνα 3×3 και βήματα (1,1).

```
model_cnlst.add(TimeDistributed(Conv2D(128, (3, 3),  
strides=(1,1),activation='relu'),input_shape=(10, 240, 240, 1)))  
model_cnlst.add(TimeDistributed(Conv2D(64, (3, 3), strides=(1,1),activation='relu')))
```

Έπειτα εφαρμόζεται το επίπεδο δισδιάστατης δειγματοληψίας (Max Pooling 2D Layer), το οποίο υποδειγματοληπτῆ την είσοδό του μαζί με τις χωρικές διαστάσεις (ύψος και πλάτος), παίρνοντας την μέγιστη τιμή του παραθύρου εισόδου. Το παράθυρο αυτό μετατοπίζεται σύμφωνα με τα βήματά του (2,2) κατά μήκος κάθε διάστασης. Στη συνέχεια, εφαρμόζονται δύο διαδοχικές συνελίξεις με αρχικό μέγεθος των φίλτρων αυτή τη φορά ίσο με 64. Τέλος, εισέρχεται το Batch Normalization επίπεδο.

```
model_cnlst.add(TimeDistributed(MaxPooling2D(2,2)))  
model_cnlst.add(TimeDistributed(Conv2D(64, (3, 3), strides=(1,1),activation='relu')))  
model_cnlst.add(TimeDistributed(Conv2D(32, (3, 3), strides=(1,1),activation='relu')))  
model_cnlst.add(TimeDistributed(BatchNormalization()))
```

Η διαδικασία που περιγράφεται παραπάνω εφαρμόζεται διαδοχικά δύο φορές και στη συνέχεια τα δεδομένα εισέρχονται στο επίπεδο Flatten, με σκοπό να μειωθούν χωρικά και να αποκτήσουν μία διάσταση.

```
model_cnlst.add(TimeDistributed(Flatten()))
```

Στη συνέχεια, τα δεδομένα εισέρχονται στο επίπεδο Dropout, το οποίο ορίζει τυχαία τις μονάδες εισόδου στο 0 με συχνότητα ίση με 0.2. Σκοπός του επιπέδου αυτού είναι να αποφευχθεί η υπερπροσαρμογή του δικτύου (overfitting).

```
model_cnlst.add(Dropout(0.2))
```

Ύστερα από την παραπάνω διαδικασία το δίκτυο συστήνεται στο επίπεδο βραχυπρόθεσμης μνήμης LSTM. Τα ορίσματα του επιπέδου αυτού αποτελούνται από τα φίλτρα που εφαρμόζονται με αριθμό ίσο με 64, τις ακολουθίες επιστροφής, που ορίζουν αν θα επιστρέψει η τελευταία έξοδος στην ακολουθία εξόδου ή στην πλήρη ακολουθία και τέλος, τη συνάρτηση dropout που είναι ίση με 0.2.

```
model_cnlst.add(LSTM(64,return_sequences=False,dropout=0.2))
```

Ακόμα, εφαρμόζεται το επίπεδο Dense, το οποίο έχει ορίσματα τις μονάδες (units) που αποτελούνται από θετικό ακέραιο αριθμό και αντιπροσωπεύουν τη διάσταση στον χώρο εξόδου, όπου αρχικά έχουν οριστεί να είναι ίσες με 128. Δεύτερο όρισμά του είναι η συνάρτηση ενεργοποίησης, η οποία είναι η ReLU. Η διαδικασία αυτή εφαρμόζεται τρεις φορές με ένα ενδιάμεσο επίπεδο ομαλοποίησης.

```
model_cnlst.add(Dense(128,activation='relu'))
```

```
model_cnlst.add(BatchNormalization())
```

```
model_cnlst.add(Dense(64,activation='relu'))
```

```
model_cnlst.add(Dense(32,activation='relu'))
```

Τέλος, αφού εφαρμοστεί ξανά ένα επίπεδο Dropout για την αποφυγή της υπερπροσαρμογής του δικτύου, εφαρμόζεται τελικά ένα επίπεδο Dense με διάσταση εξόδου ίση με 1.

```
model_cnlst.add(Dense(1, activation='sigmoid'))
```

5.3.4. Εκπαίδευση Δικτύου Ταξινομητή

Το δίκτυο εκπαιδεύεται χρησιμοποιώντας παρτίδες (batches) μεγέθους δέκα (10) εικόνων για είκοσι (20) εποχές, χρησιμοποιώντας κριτήρια Callbacks, ώστε να αποφευχθεί η υπερπροσαρμογή.

Για να αποφευχθεί η υπερπροσαρμογή χρησιμοποιούνται κάποια χρήσιμα εργαλεία από το Keras. Αρχικά, έχουμε το EarlyStopping, εδώ ορίζεται η μετρική κατά την οποία δηλώνεται πόσο υπομονή θέλουμε να κάνει το δίκτυο κατά τη διάρκεια της εκπαίδευσης, δηλαδή δηλώνεται ο αριθμός των εποχών όπου δεν θα υπάρχει κάποια βελτίωση των αποτελεσμάτων, συγκεκριμένα τρεις εποχές και η διαδικασία εκπαίδευσης θα σταματήσει. Η διαδικασία αυτή παρακολουθείται από την συνάρτηση ακρίβειας (accuracy).

```
keras.callbacks.EarlyStopping(monitor='acc',patience=5)
```

Ύστερα, έχουμε μείωση του ρυθμού εκπαίδευσης όταν κάποια μέτρηση δεν βελτιώνεται (ReduceLROnPlateau). Το εργαλείο αυτό, μειώνει τον ρυθμό εκπαίδευσης για να αποφευχθεί και πάλι η υπερπροσαρμογή, καθώς μπορεί να παρατηρηθεί πως το σκορ μπορεί να ανεβαίνει απότομα, ύστερα, από κάποια βήματα. Ως ορίσματά του έχει τον παράγοντα (factor), όπου ορίζει πότε ο ρυθμός μάθησης θα μειωθεί, την υπομονή (patience) που ισοδυναμεί με τρεις εποχές και παρακολουθείται από την συνάρτηση απωλειών των δεδομένων αξιολόγησης.

```
keras.callbacks.ReduceLROnPlateau(monitor = "val_loss", factor = 0.1, patience = 3)
```

Τέλος, υπάρχει το ModelCheckpoint, κατά το οποίο αποθηκεύεται το καλύτερο μοντέλο, ανάμεσα στις εποχές που έχουν οριστεί και αποθηκεύει αποκλειστικά τα βάρη. Η διαδικασία αυτή παρακολουθείται από την συνάρτηση απωλειών των δεδομένων αξιολόγησης (validation loss).

```
keras.callbacks.ModelCheckpoint(monitor='val_loss',save_best_only=True)
```

Για την εκπαίδευση του δικτύου απαιτείται η μεταγλώττισή του (compile). Η μεταγλώττιση του μοντέλου χρησιμοποιεί τα ορίσματα που αφορούν τον βελτιστοποιητή, τη συνάρτηση απωλειών και τη μετρητική συνάρτηση. Τα ορίσματά του δεν διαφέρουν καθόλου από εκείνα του συναλκτικού αυτοκωδικοποιητή. Δηλαδή χρησιμοποιείται ο βελτιστοποιητής Adam (Adaptive movement), η συνάρτηση απωλειών Mean Squared Error και οι μετρητικές συναρτήσεις Accuracy και Mean Squared Error.

```
model_cnlst.compile(optimizer='adam',loss='mean_squared_error',metrics=['acc','mean_squared_error'])
```

Τελική διαδικασία της εκπαίδευσης του δικτύου αποτελεί η προσαρμογή του μοντέλου(fitting). Στην οποία ορίζονται τα δεδομένα εκπαίδευσης, ελέγχου και, το μέγεθος συστάδων των εικόνων, ο αριθμός των εποχών και τα Callbacks. Για τα δεδομένα που χρησιμοποιούνται, ορίζονται και οι ετικέτες τους.

```
history=model_cnlst.fit(train_dataset_new,train_labels,batch_size=10,epochs=100,  
validation_data=(validation_dataset_new,validation_labels),  
callbacks=callbacks_list_cnlst)
```

Table 8: Συγκεντρωτικός Πίνακας Παραμέτρων Εκπαίδευσης του Συνελκτικού Ταξινομητή LSTM

Συγκεντρωτικός πίνακας παραμέτρων εκπαίδευσης	
Συστάδες εικόνων	10
Ύψος εικόνων	240
Πλάτος εικόνων	240
Βελτιστοποιητής	Adam
Συνάρτηση Ενεργοποίησης	ReLU
Συνάρτηση Απωλειών	MSE
Μετρητική Συνάρτηση	Accuracy, MSE
Αριθμός εποχών	100
Earllystopping (patience)	5
ModelCheckpoint	Save best only
Total Parameters	1,231,713
Trainable Parameters	1,231,329

5.3.5. Έλεγχος του Δικτύου Ταξινομητή LSTM

Στο σημείο αυτό γίνονται οι προβλέψεις του δικτύου. Οι προβλέψεις του δικτύου, αφορούν τις απώλειες λόγω του σφάλματος ανακατασκευής (reconstruction error). Οι απώλειες προβλέπονται για τα δεδομένα ελέγχου του δικτύου που αποτελούνται από normal και abnormal δεδομένα. Αρχικά, βρίσκονται οι προβλέψεις για κάθε εικόνα και έπειτα με την εφαρμογή του μέσου όρου βρίσκεται για όλα τα δεδομένα.

```
reconstructions = model.predict(x)
```

```
loss = tf.keras.losses.mse(reconstructions, x)
```

```
predictions.append(np.mean(loss))
```

5.3.6. Αξιολόγηση του του Δικτύου Ταξινομητή LSTM

Η αξιολόγηση του δικτύου αποτελείται από δύο μέρη, τα αποτελέσματα της καμπύλης ROC AUC και τους μετρητικούς δείκτες Accuracy Score, Precision Score, Recall Score και F1 Score. Τα δεδομένα με σκοπό να αξιολογηθούν χωρίζονται σύμφωνα με την γωνία λήψης της κάμερας και με το αν ανήκουν σε οπτικά δεδομένα, είτε σε θερμικά.

5.3.6.1. Area under the curve of the Receiver operating characteristic (ROC AUC)

Παρακάτω παρουσιάζονται τα γραφήματα της περιοχής κάτω από την καμπύλη των χαρακτηριστικών λειτουργιών του δέκτη (Area under the curve of the Receiver operating characteristic –ROC AUC), για τις κατηγορίες των εικόνων.

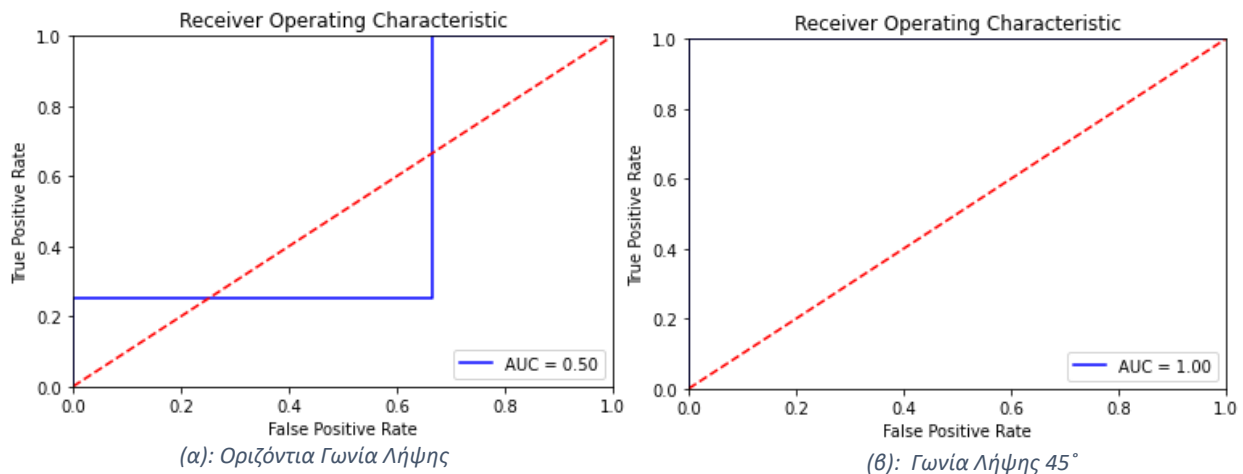
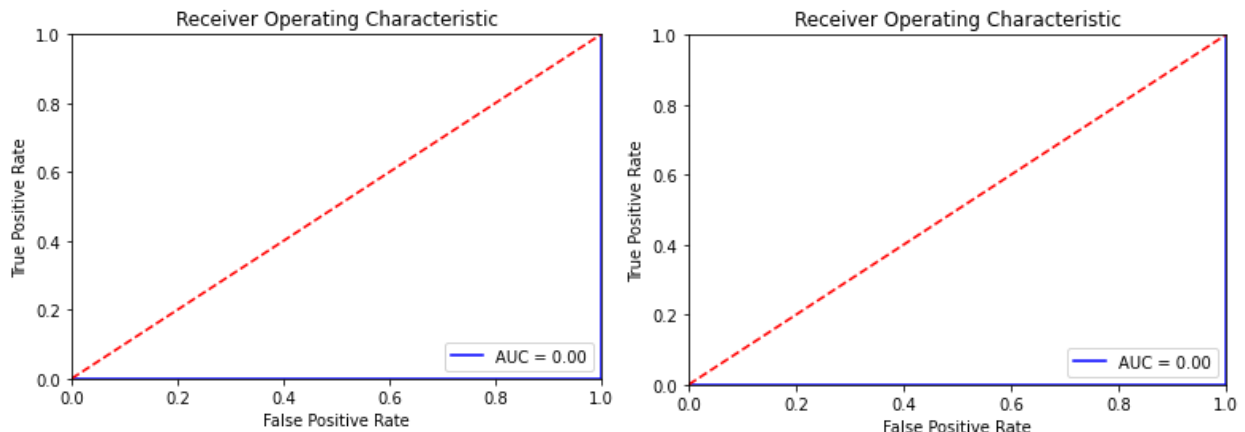


Figure 15: Γραφήματα AUC για τα Θερμικά Δεδομένα

Στην εικόνα 15 (α) η AUC έχει τιμή ίση με 0.5, που σημαίνει πως ο αλγόριθμος δεν μπορεί να ξεχωρίσει τις δύο κλάσεις που αφορούν πτώση και μη. Αντίθετα, στην 15 (β), φαίνεται πως η τιμή της AUC είναι ίση με 1. Το γεγονός αυτό σημαίνει πως ο αλγόριθμος έχει άριστη απόδοση.

Για τα δεδομένα RGB όπως φαίνεται και στις εικόνες 16 (α) και (β) φαίνεται πως η απόδοση είναι ακόμα καλύτερη. Όχι μόνο στη μία αλλά και στις δύο περιπτώσεις η AUC είναι ίση με 0. Ο αλγόριθμος φαίνεται πως όχι μόνο δεν μπορεί να ξεχωρίσει τις κλάσεις, αλλά έχει μπερδέψει τις κλάσεις μεταξύ τους. Δηλαδή έχει ταξινομήσει εικόνες που δεν είχαν πτώση σε κλάσεις που είχαν και το αντίστροφο.



(α): Οριζόντια Γωνία Λήψης

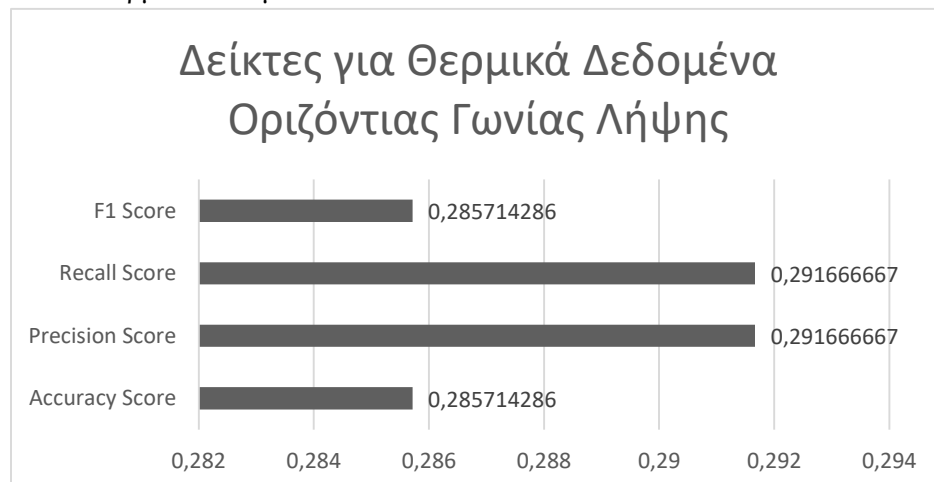
(β): Γωνία Λήψης 45°

Figure 16: Γραφήματα AUC για τα Οπτικά Δεδομένα

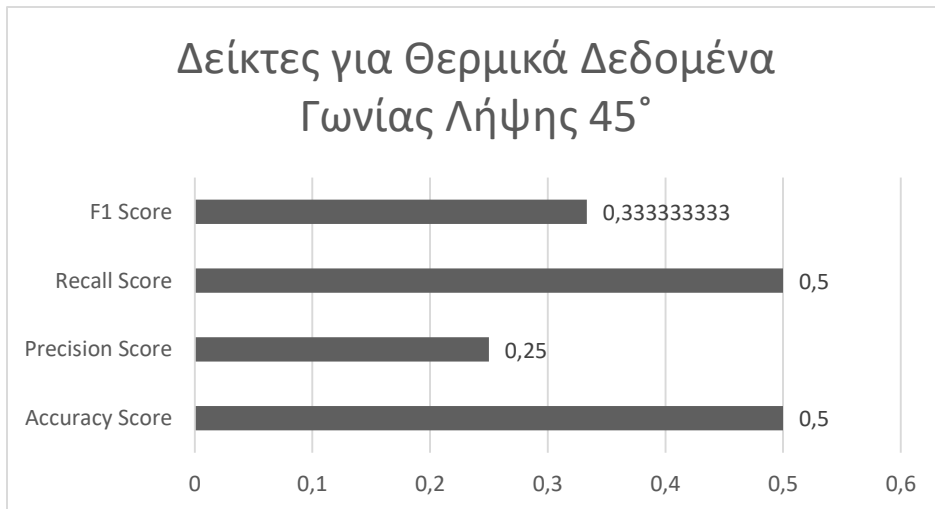
5.3.6.2. Παρουσίαση Δεικτών Αξιολόγησης για το Δίκτυο του Ταξινομητή

Οι μετρητικοί δείκτες υπολογίζονται για το σύνολο των προβλέψεων των δεδομένων σύμφωνα με τη γωνία λήψης του αισθητήρα και με τη φύση των δεδομένων, θερμικά και οπτικά. Παρακάτω παρουσιάζονται συνολικά γραφήματα των δεικτών Accuracy Score, Precision Score, Recall Score και F1 Score.

Για τα Θερμικά Δεδομένα



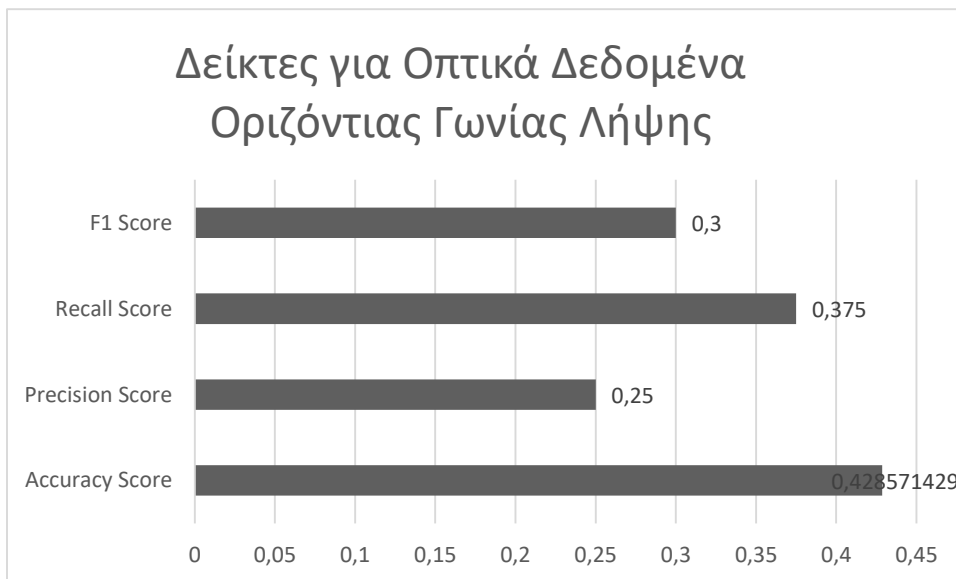
Γράφημα 5: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Οριζόντιας Γωνίας Λήψης



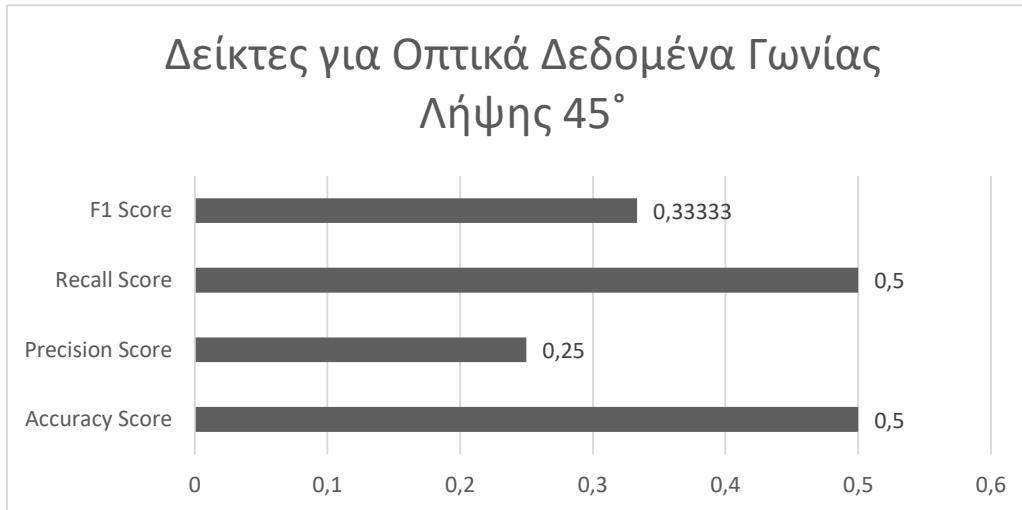
Γράφημα 6: Γράφημα Μελέτης για τα Θερμικά Δεδομένα Γωνίας Λήψης 45°

Όσο αφορά τα θερμικά δεδομένα, είναι αντιληπτό πως δεν υπάρχουν πολύ καλές αποδόσεις. Τα δεδομένα που αφορούν την γωνία λήψης 45° του αισθητήρα φαίνεται να έχουν αρκετά καλύτερα αποτελέσματα σε σχέση με αυτά της οριζόντιας γωνίας λήψης. Ωστόσο, φαίνεται πως ο αλγόριθμος δεν έχει ανταποκριθεί σωστά σε καμία από τις δύο περιπτώσεις.

Για τα Δεδομένα RGB



Γράφημα 7: Γράφημα Μελέτης για τα RGB Δεδομένα Οριζόντιας Γωνίας Λήψης



Γράφημα 8: Γράφημα Μελέτης για τα RGB Δεδομένα Γωνίας Λήψης 45°

Όσο αφορά τα οπτικά δεδομένα, από τα γραφήματα 7 και 8 φαίνεται πως δεν υπάρχουν μεγάλες διαφορές όσο αφορά την επιλογή της γωνίας λήψης. Ωστόσο, ο αλγόριθμος δεν έχει πολύ καλή απόδοση σε καμία από τις δύο περιπτώσεις.

Γενικά, από τους μετρητικούς δείκτες που παρουσιάζονται για τον αλγόριθμο του συνελκτικού ταξινομητή LSTM, φαίνεται πως δεν υπάρχει καλή απόδοση σε καμία εκ των περιπτώσεων. Όσο αφορά τη φύση των δεδομένων, φαίνεται πως για τα οπτικά δεδομένα το μοντέλο ανταποκρίνεται λίγο καλύτερα. Ακόμα, φαίνεται πως τα δεδομένα γωνίας λήψης 45° της κάμερας κατέχουν πιο υψηλούς μετρητικούς δείκτες σε σχέση με τα δεδομένα της οριζόντιας κάμερας, αλλά σε πολύ μικρό ποσοστό. Τέλος, αποδεικνύεται πως ο αλγόριθμος του συνελκτικού ταξινομητή δεν είναι ικανός σε καμία εκ των περιπτώσεων να αποδώσει σωστά και να ξεχωρίσει τις σωστές κλάσεις.

Ακόμα, παρουσιάζεται η απόδοση του δικτύου του συνελκτικού ταξινομητή για κάθε ένα από τα σετ δεδομένων.

Table 9: Αξιολόγηση Δικτύου Συνελικτικού Ταξινομητή για τα Σετ Δεδομένων

Υπολογιστικό Κόστος για τον Συνελικτικό Ταξινομητή				
	Θερμικά Δεδομένα Οριζόντιας Γωνίας Λήψης	Θερμικά Δεδομένα Γωνίας Λήψης 45°	Δεδομένα RGB Οριζόντιας Γωνίας Λήψης	Δεδομένα RGB Γωνίας Λήψης 45°
Υπολογιστική Ισχύς (hh:mm:ss)	0:12:44	0:11:02	0:12:05	0:11:29
Αριθμός Εποχών	19	17	17	17
Loss Fuction (MSE)	0.2520	0.2483	0.2500	0.2487
Metrics (accuracy, MSE)	0.500, 0.2520	0.7500, 0.2483	0.500, 0.2500	0.7500, 0.0740
Validation Loss	0.2510	0.2500	0.2497	0.2500
Validation Accuracy	0.3333	0.5000	0.6667	0.500

Στην περίπτωση του Συνελικτικού Ταξινομητή, η υπολογιστική ισχύς φαίνεται να είναι πολύ μικρή. Ακόμα, φαίνεται πως οι εποχές εκπαίδευσης δεν έχουν μεγάλες αποκλίσεις όσο αφορά τα διαφορετικά σετ των δεδομένων και όλα τα σετ χρειάζονται παρόμοιο αριθμό εποχών για να εκπαιδευτούν.

Όσο αφορά την μετρητική συνάρτηση ακριβείας (accuracy), φαίνεται να έχει μέτρια έως κακή απόδοση, σε όλες τις περιπτώσεις, με την υψηλότερη τιμή της να φθάνει σε ποσοστό 75%. Γενικά, σε αυτού του είδους την ταξινόμηση, η συνάρτηση ακρίβειας, έχει σημαντικό σκοπό καθώς εκφράζει την θεωρητική απόδοση του δικτύου. Επίσης, η συνάρτηση ακριβείας για τα δεδομένα ελέγχου, φαίνεται αρκετά χαμηλή.

Κεφάλαιο 6^ο: Συμπεράσματα

Στην παρούσα διπλωματική εργασία, παρουσιάστηκε και αξιολογήθηκε το πρόβλημα της αυτόματης ανίχνευσης πτώσεων μέσα από τεχνολογίες συνελκτικών αυτοκωδικοποιητών σε θερμικά και οπτικά δεδομένα βίντεο. Εν συνεχεία, πραγματοποιείται εντοπισμός των πτώσεων από την επιτυχία είτε την αποτυχία των αυτοκωδικοποιητών να ανακατασκευάσουν μία σκηνή λόγω της παρουσίας ανώμαλων γεγονότων. Το πρόβλημα αντιμετωπίστηκε με επιτυχία με τεχνικές μη επιβλεπόμενης μάθησης και ο αλγόριθμος είναι ικανός να διαχωρίσει ποια δεδομένα έχουν πτώσεις και ποια όχι.

Ο αλγόριθμος που χρησιμοποιείται είναι ο Spatiotemporal Convolutional LSTM Autoencoder που ανήκει στην οικογένεια των συνελκτικών νευρωνικών δικτύων μη επιβλεπόμενης μάθησης. Στόχος του είναι μέσα από την παρακολούθηση του σφάλματος ανακατασκευής και θέτοντας τα σχετικά κατώφλια να μπορέσει να αντιμετωπίσει το πρόβλημα της δυαδικής ταξινόμησης που απαιτείται και αν το σφάλμα ανακατασκευής είναι μεγαλύτερο από το καθορισμένο κατώφλι, να σημάνει συναγερμό. ο Spatiotemporal Convolutional LSTM Autoencoder, έχει την σημαντική ιδιότητα πως εκπαιδεύεται με μη επιβλεπόμενο τρόπο, επομένως δεν έχει σημασία η ποσότητα των δεδομένων του και έχει σχετικά μικρή υπολογιστική ισχύ. Ο αλγόριθμος αυτός, αποδεικνύει πως μία διαδικασία παρακολούθηση πολλών δεδομένων βίντεο, μπορεί να πραγματοποιηθεί πιο εύκολα και πιο γρήγορα με ελάχιστη ή και καθόλου ανθρώπινη επίβλεψη.

Σημαντική εξέλιξη για την απόδοση του αυτοκωδικοποιητή αποτελεί η καμπύλη αξιολόγησής του ROC AUC, η οποία δείχνει πόσο ικανός είναι ο αλγόριθμος να ταξινομήσει σωστά τα δεδομένα που περιέχουν ανθρώπινες πτώσεις και αυτά που δεν περιέχουν. Ακόμα, οι μετρητικοί δείκτες Accuracy Score, Precision Score, Recall Score και F1 Score, μπορούν να μας κατατοπίσουν στην ασφαλή εξαγωγή συμπερασμάτων για την τελική απόφαση της καλύτερης μεθόδου.

Σύμφωνα με την καμπύλη αξιολόγησης ROC AUC (στην οποία δίνεται μεγαλύτερη βαρύτητα για την εξαγωγή ασφαλών συμπερασμάτων), αλλά και των μετρητικών δεικτών, φαίνεται πως η καλύτερη επιλογή για τον εντοπισμό πτώσεων είναι η οριζόντια γωνία λήψης του αισθητήρα. Ακόμα, όσο αφορά τη φύση των δεδομένων, οι οπτικές εικόνες, έχουν ελάχιστα καλύτερη απόδοση σε σχέση με τις θερμικές. Ωστόσο, είναι πιο ασφαλείς οι αποδόσεις των θερμικών δεδομένων, καθώς αναζητάμε τον εντοπισμό ανθρώπινης πτώσης και οι θερμικοί αισθητήρες λειτουργούν με τη θερμοκρασία. Επομένως, δεν θα προκύψει κάποια λανθασμένη ταξινόμηση, αν για παράδειγμα υπάρχει πτώση κάποιου μεγάλου αντικειμένου γιατί αυτό θα έχει τη θερμοκρασία του παρασκηνίου. Ακόμα, τα θερμικά δεδομένα δεν επηρεάζονται από τις αλλαγές του φωτισμού και τις καιρικές συνθήκες, καθώς πάντοτε οι θερμοκρασία του ανθρώπινου σώματος θα κυμαίνεται σε ένα όριο συγκεκριμένων θερμοκρασιών που θα γίνονται αντιληπτές στην εκάστοτε εικόνα.

Η κύρια μέθοδος της εργασίας, αυτή του Spatiotemporal Convolutional LSTM Autoencoder, συγκρίνεται με μία διαφορετική μέθοδο επιβλεπόμενης ταξινόμησης. Η μέθοδος της επιβλεπόμενης ταξινόμησης χρησιμοποιεί τον αλγόριθμο του Συνελκτικού Ταξινομητή LSTM (Convolutional LSTM Classifier), ο οποίος επίσης ακολουθεί την ομάδα των Συνελκτικών Νευρωνικών Δικτύων (CNN). Η μέθοδος αυτή υλοποιήθηκε με σκοπό να γίνει σύγκριση με τον αλγόριθμο του συνελκτικού αυτοκωδικοποιητή και να βρεθεί ποιος αλγόριθμος έχει την καλύτερη απόδοση.

Ο Συνελκτικός Ταξινομητής LSTM, είχε πολύ κακή απόδοση όσο αφορά τα δεδομένα, όχι μόνο δεν μπόρεσε να κάνει τον διαχωρισμό μεταξύ των κλάσεων, αλλά μπέρδευε τις κλάσεις μεταξύ τους και εξήγαγε λάθος συμπεράσματα. Φαίνεται πως οι τεχνικές επιβλεπόμενης μάθησης δεν είναι κατάλληλες για την επίλυση του προβλήματος, για τον αυτόματο εντοπισμό πτώσεων.

Σε μελλοντικά σχέδια περιλαμβάνονται τεχνικές σύντηξης με πρόσθετες μονάδες απεικόνισης. Ακόμα, είναι δυνατόν να προστεθούν εικόνες με πτώση αντικειμένων, ώστε ο αλγόριθμος να μπορεί να ξεχωρίσει την ανθρώπινη πτώση έναντι αυτής του αντικειμένου και να σημάνει τον απαραίτητο συναγερμό. Επίσης θα προστεθεί η μελέτη πρόσθετων τρόπων κωδικοποίησης όλων των διαθέσιμων τρόπων για τη μεγιστοποίηση των δυνατοτήτων ανίχνευσης.

Βιβλιογραφία

- [1] I. Katsamenis, E. Protopapadakis, A. Voulodimos, D. Dres, and D. Drakoulis, “Man overboard event detection from RGB and thermal imagery: possibilities and limitations,” in *Proceedings of the 13th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, Corfu Greece, Jun. 2020, pp. 1–6. doi: 10.1145/3389189.3397998.
- [2] “Man overboard,” *Wikipedia*. Apr. 04, 2021. Accessed: Oct. 03, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Man_overboard&oldid=1015980345
- [3] N. Bakalos, I. Katsamenis, and A. Voulodimos, “Man Overboard: Fall detection using spatiotemporal convolutional autoencoders in maritime environments,” in *The 14th Pervasive Technologies Related to Assistive Environments Conference*, New York, NY, USA, Jun. 2021, pp. 420–425. doi: 10.1145/3453892.3461326.
- [4] Π. Μπιοταζόπουλος, “Διακρίσιμα Αρσενικά Ένεργοποιημένα Συμπεριφορικά Δεδομένα,” 2020.
- [5] A. S. Voulodimos, D. I. Kosmopoulos, N. D. Doulamis, and T. A. Varvarigou, “A top-down event-driven approach for concurrent activity recognition,” *Multimed. Tools Appl.*, vol. 69, no. 2, pp. 293–311, Mar. 2014, doi: 10.1007/s11042-012-0993-4.
- [6] N. D. Doulamis, A. S. Voulodimos, D. I. Kosmopoulos, and T. A. Varvarigou, “Enhanced human behavior recognition using HMM and evaluative rectification,” in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams - ARTEMIS '10*, Firenze, Italy, 2010, p. 39. doi: 10.1145/1877868.1877880.
- [7] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and N. Matsatsinis, “3D measures exploitation for a monocular semi-supervised fall detection system,” *Multimed. Tools Appl.*, vol. 75, Apr. 2015, doi: 10.1007/s11042-015-2513-9.
- [8] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Robust Video Surveillance for Fall Detection Based on Human Shape Deformation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 611–622, May 2011, doi: 10.1109/TCSVT.2011.2129370.
- [9] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, “A Posture Recognition-Based Fall Detection System for Monitoring an Elderly Person in a Smart Home Environment,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1274–1286, Nov. 2012, doi: 10.1109/TITB.2012.2214786.
- [10] Y. Liu, C.-L. Li, and B. Póczos, “Classifier Two-Sample Test for Video Anomaly Detections,” p. 12.
- [11] X. Mo, V. Monga, R. Bala, and Z. Fan, “Adaptive sparse representations for video anomaly detection,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 631–645, Apr. 2014, doi: 10.1109/TCSVT.2013.2280061.
- [12] S. Wan, X. Xu, T. Wang, and Z. Gu, “An Intelligent Video Analysis Method for Abnormal Event Detection in Intelligent Transportation Systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4487–4495, Jul. 2021, doi: 10.1109/TITS.2020.3017505.
- [13] R. Leyva, V. Sanchez, and C.-T. Li, “Fast Detection of Abnormal Events in Videos with Binary Features,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 1318–1322. doi: 10.1109/ICASSP.2018.8461759.
- [14] S. Lee, H. G. Kim, and Y. M. Ro, “BMAN: Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection,” *IEEE Trans. Image Process.*, vol. 29, pp. 2395–2408, 2020, doi: 10.1109/TIP.2019.2948286.
- [15] “[1609.08938] A Discriminative Framework for Anomaly Detection in Large Videos.” <https://arxiv.org/abs/1609.08938> (accessed Oct. 03, 2021).
- [16] J. K. Dutta and B. Banerjee, “Online Detection of Abnormal Events Using Incremental Coding Length,” p. 7.

- [17]R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, “Unmasking the Abnormal Events in Video,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2914–2922. doi: 10.1109/ICCV.2017.315.
- [18]F. Jiang, Y. Wu, and A. K. Katsaggelos, “A Dynamic Hierarchical Clustering Method for Trajectory-Based Unusual Video Event Detection,” *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 907–913, Apr. 2009, doi: 10.1109/TIP.2008.2012070.
- [19]K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Gaussian Process Regression-Based Video Anomaly Detection and Localization With Hierarchical Feature Representation,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5288–5301, Dec. 2015, doi: 10.1109/TIP.2015.2479561.
- [20]C. Lu, J. Shi, and J. Jia, “Abnormal Event Detection at 150 FPS in MATLAB,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 2720–2727. doi: 10.1109/ICCV.2013.338.
- [21]M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning Temporal Regularity in Video Sequences,” *ArXiv160404574 Cs*, Apr. 2016, Accessed: Oct. 03, 2021. [Online]. Available: <http://arxiv.org/abs/1604.04574>
- [22]D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, “Learning Deep Representations of Appearance and Motion for Anomalous Event Detection,” *ArXiv151001553 Cs*, Oct. 2015, Accessed: Oct. 03, 2021. [Online]. Available: <http://arxiv.org/abs/1510.01553>
- [23]L. Wang, F. Zhou, Z. Li, W. Zuo, and H. Tan, “Abnormal Event Detection in Videos Using Hybrid Spatio-Temporal Autoencoder,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 2276–2280. doi: 10.1109/ICIP.2018.8451070.
- [24]K. Makantasis, A. Nikitakis, A. D. Doulamis, N. D. Doulamis, and I. Papaefstathiou, “Data-Driven Background Subtraction Algorithm for In-Camera Acceleration in Thermal Imagery,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2090–2104, Sep. 2018, doi: 10.1109/TCSVT.2017.2711259.
- [25]A. S. Voulodimos, N. D. Doulamis, D. I. Kosmopoulos, and T. A. Varvarigou, “Improving Multi-Camera Activity Recognition by Employing Neural Network Based Readjustment,” *Appl. Artif. Intell.*, vol. 26, no. 1–2, pp. 97–118, Jan. 2012, doi: 10.1080/08839514.2012.629540.
- [26]G. Goos *et al.*, “Lecture Notes in Computer Science,” p. 760.
- [27]D. Yeo, J. Son, B. Han, and J. H. Han, “Superpixel-Based Tracking-by-Segmentation Using Markov Chains,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 511–520. doi: 10.1109/CVPR.2017.62.
- [28]D. I. Kosmopoulos, A. S. Voulodimos, and A. D. Doulamis, “A System for Multicamera Task Recognition and Summarization for Structured Environments,” *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 161–171, Feb. 2013, doi: 10.1109/TII.2012.2212712.
- [29]H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, “Analyzing Tracklets for the Detection of Abnormal Crowd Behavior,” in *2015 IEEE Winter Conference on Applications of Computer Vision*, Jan. 2015, pp. 148–155. doi: 10.1109/WACV.2015.27.
- [30]Sk. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, “Trajectory-Based Surveillance Analysis: A Survey,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1985–1997, Jul. 2019, doi: 10.1109/TCSVT.2018.2857489.
- [31]A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A Brief Review,” *Comput. Intell. Neurosci.*, vol. 2018, p. e7068349, Feb. 2018, doi: 10.1155/2018/7068349.
- [32]“Τεχνητή Νοημοσύνη,” *Τεχνητή Νοημοσύνη*. <https://aibook.gr/> (accessed Oct. 03, 2021).
- [33]“Deep Learning.” <https://www.deeplearningbook.org/> (accessed Oct. 03, 2021).
- [34]“Glossary of Terms Journal of Machine Learning.” <http://ai.stanford.edu/~ronnyk/glossary.html> (accessed Oct. 03, 2021).
- [35]S. S. Haykin and S. S. Haykin, *Neural networks and learning machines*, 3rd ed. New York: Prentice Hall, 2009.

- [36]“Everything you need to know about ‘Activation Functions’ in Deep learning models | by Vandit Jain | Towards Data Science.” <https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253> (accessed Oct. 03, 2021).
- [37]A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU),” *ArXiv180308375 Cs Stat*, Feb. 2019, Accessed: Oct. 03, 2021. [Online]. Available: <http://arxiv.org/abs/1803.08375>
- [38]C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” *ArXiv181103378 Cs*, Nov. 2018, Accessed: Oct. 03, 2021. [Online]. Available: <http://arxiv.org/abs/1811.03378>
- [39]“Understanding different Loss Functions for Neural Networks | by Shiva Verma | Towards Data Science.” <https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718> (accessed Oct. 03, 2021).
- [40]“Deep learning,” *Wikipedia*. Sep. 20, 2021. Accessed: Oct. 03, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Deep_learning&oldid=1045496580
- [41]W. J. Zhang, G. Yang, Y. Lin, C. Ji, and M. M. Gupta, “On Definition of Deep Learning,” in *2018 World Automation Congress (WAC)*, Jun. 2018, pp. 1–5. doi: 10.23919/WAC.2018.8430387.
- [42]“What is Deep Learning and How does it work? | Towards Data Science.” <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac> (accessed Oct. 03, 2021).
- [43]Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in *The handbook of brain theory and neural networks*, Cambridge, MA, USA: MIT Press, 1998, pp. 255–258.
- [44]“A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science.” <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (accessed Oct. 04, 2021).
- [45]“Auto-Encoder: What Is It? And What Is It Used For? (Part 1) | by Will Badr | Towards Data Science.” <https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726> (accessed Oct. 04, 2021).
- [46]“Autoencoder,” *Wikipedia*. Aug. 30, 2021. Accessed: Oct. 04, 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Autoencoder&oldid=1041423882>
- [47]“Understanding LSTM Networks -- colah’s blog.” <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed Oct. 04, 2021).
- [48]“Medium.” <https://towardsdatascience.com/prototyping-an-anomaly-detection-system-for-videos-step-by-step-using-lstm-convolutional-4e06b7dcdd29> (accessed Oct. 04, 2021).
- [49]“About,” Apr. 20, 2012. <https://web.archive.org/web/20120420010049/http://www.python.org/about/> (accessed Oct. 04, 2021).
- [50]“The Making of Python,” Sep. 01, 2016. <https://web.archive.org/web/20160901183332/http://www.artima.com/intv/pythonP.html> (accessed Oct. 04, 2021).
- [51]“The History of Python: A Brief Timeline of Python.” <https://web.archive.org/web/20200605032200/https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html> (accessed Oct. 04, 2021).
- [52]“What’s New in Python 2.0 — Python v2.7.3 documentation,” Oct. 23, 2012. <https://web.archive.org/web/20121023112045/http://docs.python.org/whatsnew/2.0.html> (accessed Oct. 04, 2021).
- [53]“Python 3.0 Release | Python.org.” <https://web.archive.org/web/20200614153714/https://www.python.org/download/releases/3.0/> (accessed Oct. 04, 2021).
- [54]“Εισαγωγή στον προγραμματισμό με την Python.” <http://aggelid.mysch.gr/pythonbook/> (accessed Oct. 04, 2021).

- [55] “4 Reasons Why You Should Use Google Colab for Your Next Project | by Orhan G. Yalçın | Towards Data Science.” <https://towardsdatascience.com/4-reasons-why-you-should-use-google-colab-for-your-next-project-b0c4aaad39ed> (accessed Oct. 04, 2021).
- [56] M. Abadi *et al.*, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *ArXiv160304467 Cs*, Mar. 2016, Accessed: Oct. 04, 2021. [Online]. Available: <http://arxiv.org/abs/1603.04467>
- [57] M. Abadi *et al.*, *TensorFlow, Large-scale machine learning on heterogeneous systems*. 2015. doi: 10.5281/zenodo.4724125.
- [58] K. Team, “Keras documentation: Keras API reference.” <https://keras.io/api/> (accessed Oct. 04, 2021).
- [59] K. Team, “Keras documentation: Core layers.” https://keras.io/api/layers/core_layers/ (accessed Oct. 04, 2021).
- [60] T. Oliphant, *Guide to NumPy*. 2006.
- [61] G. R. Bradski and A. Kaehler, *Learning OpenCV: computer vision with the OpenCV library*, 1. ed., [Nachdr.]. Beijing: O’Reilly, 2011.
- [62] K. Pulli, A. Baksheev, K. Korniyakov, and V. Eruhimov, “Real-Time Computer Vision with OpenCV,” *Commun. ACM*, vol. 55, pp. 61–69, Jun. 2012, doi: 10.1145/2184319.2184337.
- [63] “GitHub Stats — Matplotlib 3.4.3 documentation.” https://matplotlib.org/stable/users/github_stats.html (accessed Oct. 04, 2021).
- [64] G. Hackeling, *Mastering machine learning with scikit-learn: apply effective learning algorithms to real-world problems using scikit-learn*. Birmingham: Packt Publ, 2014.
- [65] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv14126980 Cs*, Jan. 2017, Accessed: Oct. 04, 2021. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [66] “Understanding AUC - ROC Curve | by Sarang Narkhede | Towards Data Science.” <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> (accessed Oct. 04, 2021).
- [67] K. Team, “Keras documentation: TimeDistributed layer.” https://keras.io/api/layers/recurrent_layers/time_distributed/ (accessed Oct. 04, 2021).
- [68] “A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databases | Elsevier Enhanced Reader.” <https://reader.elsevier.com/reader/sd/pii/S1077314299907610?token=95506FC1DC928554C41D65F29FA18ED8A541305AA1A89ED32FE6C668F46F87591DC5D9A5AEE05E9FC7735443C15A8F9F&originRegion=eu-west-1&originCreation=20211006164236> (accessed Oct. 06, 2021).
- [69] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, “An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources,” *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 150–166, Jan. 2003, doi: 10.1109/TNN.2002.806645.
- [70] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks,” in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2015, pp. 4959–4962. doi: 10.1109/IGARSS.2015.7326945.
- [71] “ScienceDirect Full Text PDF.” Accessed: Oct. 06, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168400000190/pdf?md5=2e0fd07ee48a4c285f2161a962b62212&pid=1-s2.0-S0165168400000190-main.pdf&isDTMRedir=Y>
- [72] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, “An efficient fully unsupervised video object segmentation scheme using an adaptive neural-network classifier architecture,” *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 616–630, May 2003, doi: 10.1109/TNN.2003.810605.
- [73] A. Temenos, E. Protopapadakis, A. Doulamis, and N. Temenos, “Building Extraction from RGB Satellite Images using Deep Learning: A U-Net Approach,” in *The 14th Pervasive Technologies Related to Assistive Environments Conference*, New York, NY, USA, Jun. 2021, pp. 391–395. doi: 10.1145/3453892.3461320.
- [74] N. Bakalos, I. Rallis, N. Doulamis, A. Doulamis, E. Protopapadakis, and A. Voulodimos, “Choreographic Pose Identification using Convolutional Neural Networks,” in *2019 11th*

International Conference on Virtual Worlds and Games for Serious Applications (VS-Games), Sep. 2019, pp. 1–7. doi: 10.1109/VS-Games.2019.8864522.

- [75] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks,” in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2015, pp. 4959–4962. doi: 10.1109/IGARSS.2015.7326945.
- [76] N. D. Doulamis, A. S. Voulodimos, D. I. Kosmopoulos, and T. A. Varvarigou, “Enhanced human behavior recognition using HMM and evaluative rectification,” in *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams - ARTEMIS '10*, Firenze, Italy, 2010, p. 39. doi: 10.1145/1877868.1877880.