Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Σθεναροί και γενικεύσιμοι αλγόριθμοι τεχνητής νοημοσύνης για κατηγοριοποίηση ανισοκατανεμημένων κλάσεων

Διπλωματικη Εργασια

του

ΒΑΣΙΛΗ ΚΑΡΑΜΠΙΝΗ

Επιβλέπων: Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
Μέλος ΕΔΙΠ Ε.Μ.Π

Αθήνα, Νοέμβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Σθεναροί και γενικεύσιμοι αλγόριθμοι τεχνητής νοημοσύνης για κατηγοριοποίηση ανισοκατανεμημένων κλάσεων

Διπλωματικη Εργασια

του

ΒΑΣΙΛΗ ΚΑΡΑΜΠΙΝΗ

Επιβλέπων: Αθανάσιος Βουλόδημος
Επίκουρος Καθηγητής Ε.Μ.Π

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
Μέλος ΕΔΙΠ Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Νοέμβρίου 2023.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Αθανάσιος Βουλόδημος	Γεώργιος Στάμου	Ανδρέας-Γεώργιος Σταφυλοπάτης
Επίκουρος Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π

Αθήνα, Νοέμβριος 2023

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Διπλωματικής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Διπλωματική μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Διπλωματική Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Βασίλης Καραμπίνης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

1η Νοέμβρίου 2023

Περίληψη

Η μηχανική μάθηση και η ανάπτυξη των νευρωνικών δικτύων έχει οδηγήσει στην εκτεταμένη χρήση τους στην επίλυση προβλημάτων κατηγοριοποίηση εικόνων. Για την εκπαίδευση όμως τέτοιων μοντέλων και την παραγωγή συστημάτων, που είναι αποδοτικά και γενικεύσιμα είναι αναγκαία η εκπαίδευση σε ένα μεγάλο πλήθος απο επισημασμενα δεδομένα. Σε τομείς όπως η ανάλυση ιατρικών εικόνων, η ανάπτυξη τέτοιων μεγάλων συνόλων δεδομένων αποτελεί μια σημαντική πρόκληση. Για αυτόν τον λόγο σημαντικό ερευνητικό ενδιαφέρον σε αυτόν τον τομέα επικεντρώνεται στην ανάπτυξη αποδοτικών τεχνικών εκπαίδευσης, που να βασίζονται σε μικρό αριθμό δειγμάτων.

Σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι η μελέτη τέτοιων τεχνικών εκπαίδευσης. Πιο συγκεκριμένα εφαρμόστηκε η τεχνική του few shot learning, η οποία χωρίζει τα δεδομένα σε υποσύνολα τα οποία αποτελούνται από συγκεκριμένο αριθμό δειγμάτων και ανήκουν σε τυχαια επιλεγμένες κλάσεις. Αφου τα δεδομένα χωριστουν σε αυτα τα υποσύνολα, στην συνέχεια εκτελείται η διαδικασία εκπαίδευσης σε κάθε ένα από τα υποσύνολα επεισοδιακα. Επιπλέον, χρησιμοποιήθηκαν τεχνικές που εμπνέονται από το μοντέλο εκπαίδευσης δασκάλου μαθητή και έχουν ως σκοπό την εκπαίδευση παραμέτρων ενος δικτύου, που θα είναι εύκολα προσαρμόσιμες σε καινουρια σύνολα δεδομένων μετα απο περιορισμένο αριθμό επαναλήψεων. Στην συνέχεια δοκιμάστηκαν μοντελα μετασχηματιστών όρασης (Vision Transformers) για την κατηγοριοποίηση αυτών των δεδομένων πράγμα, το οποίο ήταν σημαντική πρόκληση λόγω του περιορισμένου αριθμού δειγμάτων τον οποιο διέθεταν τα σύνολα δεδομένων τα οποία χρησιμοποιήθηκαν. Για την εκπαίδευση αυτών των δικτύων χρησιμοποιήθηκε η μέθοδος του self-supervised learning, αλλά και σύνθετες τεχνικές επαύξησης δεδομένων για την δημιουργια πιο περίπλοκων αναπαραστάσεων, με σκοπό την αντιμετώπιση του προβλήματος του χαμηλου αριθμού δειγμάτων.

Τέλος τα αποτελέσματα του μοντέλου μετασχηματιστή (Transformer) που προέκυψαν συγκρίθηκαν με το δίκτυο, το οποίο επιτύγχανε τα καλύτερα αποτελέσματα για το συγκεκριμένο πρόβλημα και σύνολα δεδομένων που επιλέχθηκαν. Συγκρίνοντας αυτά τα δύο δίκτυα παρατηρήθηκε οτι το μοντέλο του μετασχηματιστή Transformer πετυχαίνει αποτελέσματα συγκρίσιμα με την προαναφερθείσα μέθοδο.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Κατηγοριοποίηση, Νευρωνικά Δίκτυα, Συνελικτικά νευρωνικά δίκτυα, Μετασχηματιστες, Μετασχηματιστες όρασης, Επαύξηση δεδομένων,Επιβλεπόμενη μάθηση, Self supervised learning, Few shot learning, PixMix, Cutmix, Mixup, Cutout

Abstract

The ever-growing use of machine learning in different application and the extensive development of neural networks has shown great results in solving image classification problems. However to create efficient models with the ability to generalize, a large amount of labeled data samples is required. The creation of such large datasets could prove to be a challenging task especially in fields like medical imaging where there is a scarcity of data. For this reason, there is a substantial research interest in developing efficient training techniques based on a small number of samples in this field.

This diploma thesis aims to study such techniques. More specifically the few shot learning method is applied, which separates the data in two subsets one used for training and the other used as a novel dataset for the trained model to adapt to and make predictions for the samples. The datasets are then further divided into tasks, which consists of a specific number of samples belonging to randomly selected classes. The training process then proceeds episodically for each task. In addition, techniques inspired by a teacher-student training model were used to train network models which will be capable of adapting to a new dataset domain with a limited amount of training. Furthermore, Vision Transformer models were used for the classification of these data, which posed a significant challenge due to the limited number of samples available in the given datasets. For the training of the vision transformer network the method of self-supervised learning was utilized. To address the problem of the small number of samples advanced augmentation techniques were used.

Finally, the results of the Transformer model that emerged were compared with the model that achieved the best results for the specific problem and the datasets selected. From this comparison it was observed that the transformer model achieved comparable results to the previously mentioned method.

Keywords

Machine Learning, Classification, Neural Networks, Convolution Neural Networks, Transformers, Vision Transformers. Data augmentation, Supervised learning, Self supervised learning, Few shot learning, PixMix, Cutmix, Mixup, Cutout

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ.Βουλόδημο για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο AILS. Επίσης ευχαριστώ ιδιαίτερα την κ.Τσούβελη και τον Αναστάση Αρσένο για την καθοδήγησή τους και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Αθήνα, Νοέμβριος 2023

Βασίλης Καραμπίνης

στην μνήμη της γιαγιάς μου,
Γεωργίας

Περιεχόμενα

Περίληψη 1

Abstract 3

Ευχαριστίες 5

I Πρόλογος 19

1 Εισαγωγή 21
1.1 Αντικείμενο της διπλωματικής . . . 21
1.2 Οργάνωση του τόμου . . . 22

II Θεωρητικό Μέρος 23

2 Θεωρητικό υπόβαθρο 25
2.1 Τεχνητή Νοημοσύνη και Μηχανική Μάθηση . . . 25
2.1.1 Τεχνητή Νοημοσύνη . . . 25
2.1.2 Μηχανική Μάθηση . . . 25
2.2 Νευρωνικά Δίκτυα . . . 27
2.2.1 Βιολογικά Νευρωνικά Δίκτυα . . . 27
2.2.2 Τεχνητά Νευρωνικά Δίκτυα . . . 27
2.2.3 Συνελικτικά Νευρωνικά Δίκτυα . . . 28
2.2.4 Batch Normalization . . . 29
2.2.5 Transformers . . . 29
2.3 Πρόβλημα Ταξινόμησης . . . 35
2.4 Επαύξηση Δεδομένων Data Augmentation . . . 35
2.5 Τεχνικές Εκπαίδευσης . . . 36
2.5.1 Few shot learning . . . 36
2.5.2 Transfer learning . . . 36
2.5.3 Meta Learning[1] . . . 37

III Πρακτικό Μέρος 39

3 Περιγραφή θέματος 41
3.1 Σχετικές εργασίες . . . 41

4 Ανάλυση και σχεδίαση **43**
4.1 Ανάλυση και περιγραφή δεδομένων . . . 43
4.1.1 Pap Smear . . . 43
4.1.2 BreakHis dataset . . . 44
4.1.3 Isic dataset . . . 46
4.2 Προεπεξεργασία δεδομένων . . . 47
4.2.1 Αρχική προεπεξεργασία και απλές τεχνικές επαύξησης δεδομένων . . . 47
4.2.2 Σύνθετες Τεχνικές Επαύξησης Δεδομένων . . . 48
4.3 Ανάλυση διαδικασίας εκπαίδευσης . . . 54
4.3.1 Few Shot Learning . . . 55
4.3.2 Μοντέλο εκπαίδευσης στο MetaMed . . . 56
4.3.3 Αλγόριθμοι εκπαίδευσης Meta training . . . 56
4.3.4 Διαδικασία εκπαίδευσης MetaMed[2] . . . 63
4.4 PFEMED[3] . . . 67
4.4.1 PFENET[4] . . . 67
4.4.2 PT-MAP[5] . . . 68
4.4.3 Feature Extraction and Enhancement(FEE)[3] . . . 71
4.5 Few Ture[6] . . . 75
4.5.1 Vision Transformers . . . 77
4.5.2 Masked Image Modeling . . . 78
4.5.3 Self-Distillation . . . 78
4.5.4 iBOT[7] . . . 78
4.5.5 Pre-Train . . . 80
4.5.6 Ταξινόμηση με επαναυπολογισμό των παραμέτρων της ομοιότητας των tokens . . . 80
4.5.7 Υπολογισμός τιμών πίνακα σημαντικότητα . . . 82

5 Υλοποίηση **85**
5.1 Βασικές Μέθοδοι Αναπτυξής των Αλγορίθμών και επεξεργασίας των Δεδομένων 85
5.1.1 Ανάλυση και Προεπεξεργασια Δεδομένων . . . 85
5.1.2 Αλγόριθμοι υλοποιήσης MetaMed . . . 86
5.1.3 Τεχνικές υλοποιήσης Transfer Learning . . . 87
5.1.4 Τεχνικές υλοποιήσης Meta Learning . . . 91
5.1.5 Τεχνικές υλοποιήσης FewTURE αλγορίθμου . . . 92

IV Επίλογος **111**

6 Επίλογος **113**
6.1 Συμπεράσματα . . . 113
6.2 Μελλοντικές Επεκτάσεις . . . 114

Βιβλιογραφία **122**

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια **123**

Απόδοση ξενόγλωσσων όρων **125**

Κατάλογος Σχημάτων

2.1 Transformer architecture [8] . . . 30
2.2 ViT Transformers [9] . . . 31
2.3 Attention Value pipeline [10] . . . 34
2.4 Attention Block [10] . . . 34
2.5 Multi Head Attention Block [10] . . . 35

4.1 Pap-Smear samples . . . 44
4.2 Pap-Smear with no augmentation . . . 44
4.3 BreakHis_40 samples . . . 45
4.4 BreakHis_40 with no augmentation . . . 46
4.5 Isic_2018 samples . . . 47
4.6 Isic_2018 with no augmentation . . . 47
4.7 Pap-Smear with cutout . . . 48
4.8 BreakHis_40 with cutout . . . 48
4.9 ISIC_2018 with cutout . . . 49
4.10 Pap-Smear with cutmix . . . 49
4.11 BreakHis_40 with cutmix . . . 50
4.12 ISIC_2018 with cutmix . . . 50
4.13 Pap-Smear with mixup . . . 51
4.14 BreakHis_40 with mixup . . . 51
4.15 ISIC_2018 with mixup . . . 51
4.16 Pap-Smear with Pixmix . . . 53
4.17 BreakHis_40 with Pixmix . . . 54
4.18 ISIC_2018 with Pixmix . . . 54
4.19 MetaMed pipeline . . . 67
4.20 PFEMED feature enrichment . . . 71
4.21 PFEMED pipeline . . . 72
4.22 FewTURE important generation . . . 81
4.23 FewTURE important generation . . . 83

5.1 Pap Smear width dimensions . . . 86
5.2 Pap Smear height dimensions . . . 87
5.3 Ibot framework . . . 94
5.4 BreakHis x40 samples . . . 101
5.5 BreakHis x100 samples . . . 101

5.6 BreakHis x200 samples . . . 101
5.7 BreakHis x400 samples . . . 101
5.8 Pap Smear 2way 3shot . . . 102
5.9 Pap Smear 2way 5shot . . . 102
5.10 Pap Smear 2way 10shot . . . 102
5.11 Confidence score of Pap Smear in 2way tasks . . . 102
5.12 Pap Smear 2way 3shot loss . . . 103
5.13 Pap Smear 2way 5shot loss . . . 103
5.14 Pap Smear 2way 10shot loss . . . 103
5.15 Losses of Pap Smear in 2way tasks . . . 103
5.24 Pap Smear 3way 3shot loss . . . 103
5.25 Pap Smear 3way 5shot loss . . . 103
5.26 Pap Smear 3way 10shot loss . . . 103
5.27 Losses of Pap Smear in 3way tasks . . . 103
5.16 Pap Smear 2way 3shot accuracy . . . 104
5.17 Pap Smear 2way 5shot accuracy . . . 104
5.18 Pap Smear 2way 10shot accuracy . . . 104
5.19 Accuracy of Pap Smear in 2way tasks . . . 104
5.28 Pap Smear 3way 3shot accuracy . . . 104
5.29 Pap Smear 3way 5shot accuracy . . . 104
5.30 Pap Smear 3way 10shot accuracy . . . 104
5.31 Accuracy of Pap Smear in 3way tasks . . . 104
5.20 Pap Smear 3way 3shot . . . 105
5.21 Pap Smear 3way 5shot . . . 105
5.22 Pap Smear 3way 10shot . . . 105
5.23 Confidence score of Pap Smear in 3way tasks . . . 105
5.32 BreakHis 40 2way 3shot loss . . . 106
5.33 BreakHis 40 2way 5shot loss . . . 106
5.34 BreakHis 40 2way 10shot loss . . . 106
5.35 Losses of BreakHis 40 in 2way tasks . . . 106
5.36 BreakHis 40 2way 3shot accuracy . . . 107
5.37 BreakHis 40 2way 5shot accuracy . . . 107
5.38 BreakHis 40 2way 10shot accuracy . . . 107
5.39 Accuracy of BreakHis 40 in 2way tasks . . . 107
5.40 BreakHis 40 3way 3shot loss . . . 107
5.41 BreakHis 40 3way 5shot loss . . . 107
5.42 BreakHis 40 3way 10shot loss . . . 107
5.43 Losses of BreakHis 40 in 3way tasks . . . 107
5.44 BreakHis 40 3way 3shot accuracy . . . 108
5.45 BreakHis 40 3way 5shot accuracy . . . 108
5.46 BreakHis 40 3way 10shot accuracy . . . 108
5.47 Accuracy of BreakHis 40 in 3way tasks . . . 108
5.48 ISIC 2018 2way 3shot loss . . . 108

5.49 ISIC 2018 2way 5shot loss 108
5.50 ISIC 2018 2way 10shot loss 108
5.51 Losses of ISIC 2018 in 2way tasks 108
5.52 ISIC 2018 3way 3shot accuracy 109
5.53 ISIC 2018 3way 5shot accuracy 109
5.54 ISIC 2018 3way 10shot accuracy 109
5.55 Accuracy of ISIC 2018 in 2way tasks 109
5.56 ISIC 2018 3way 3shot loss 109
5.57 ISIC 2018 3way 5shot loss 109
5.58 ISIC 2018 3way 3shot loss 109
5.59 Losses of ISIC 2018 in 3way tasks 109
5.60 ISIC 2018 3way 3shot accuracy 110
5.61 ISIC 2018 3way 5shot accuracy 110
5.62 ISIC 2018 3way 10shot accuracy 110
5.63 Accuracy of ISIC 2018 in 3way tasks 110

Κατάλογος Πινάκων

5.1 Information about datasets 86
5.2 PapSmear Transfer Learning model accuracy 88
5.3 ISIC2018 Transfer Learning model accuracy 89
5.4 BreakHis Transfer Learning model accuracy 90
5.5 Pretrain parameters 96
5.6 Meta train hyperparameters 96
5.7 PixMix hyperparameters 97
5.8 *Pap-Smear Accuracy for all models* 98
5.9 ISIC train results 98
5.10 *ISIC Accuracy for all models* 99
5.11 BreakHis train results 99
5.12 *BreakHis Accuracy for all models* 100

Μέρος Ι

Πρόλογος

Κεφάλαιο 1

Εισαγωγή

Η τεχνητή νοημοσύνη αποτελεί ένα συνεχώς εξελισσόμενο επιστημονικό τομέα με ενα ευρύ φάσμα εφαρμογών, ο οποίος έχει ως σκοπό την ανάπτυξη ευφυή συστημάτων για την επίλυση προβλημάτων. Πιο συγκεκριμένα σκοπός της τεχνητής νοημοσύνης είναι η ανάπτυξη συστημάτων, τα οποία μπορούν να λάβουν αποφάσεις και να πραγματοποιήσουν προβλέψεις χωρις την ανθρώπινη παρέμβαση. Η μηχανική μάθηση αποτελεί μια κατηγορία της τεχνητής νοημοσύνης, που έχει ως σκοπό την εκπαίδευση δικτύων σε μεγάλα σύνολα δεδομένων, με στόχο την εκμάθηση προτύπων και την χρήση αυτών στην λήψη αποφάσεων.

Τα νευρωνικά δίκτυα έχουν δείξει να επιτυγχάνουν πολύ καλή απόδοση στα προβλήματα μηχανικής μάθησης σε διάφορους τομείς, όπως και αυτός της ανάλυσης ιατρικών εικόνων. Το κύριο μειονεκτήματα της εφαρμογής μεθόδων μηχανικής μάθησης είναι η ανάγκη για μεγάλο όγκο δεδομένων κατα την εκπαίδευση του δικτύου, ώστε αυτό να επιτύχει αποδοτικά αποτελέσματα. Κάτι τέτοιο αποτελεί ιδιαίτερη πρόκληση στο τομέα των ιατρικών εικόνων, που είναι πιο δύσκολη η συλλογή μεγάλου αριθμού δεδομένων και ακόμα και αν αυτό είναι δυνατό τα δείγματα για την ίδια ασθένεια μπορεί να έχουν αρκετές διαφορές στον τρόπο με τον οποίο απεικονίζονται (τέτοιες διαφορές μπορεί να εξαρτώνται από το ιατρικό μηχάνημα που εξάγει τα αποτελέσματα, την τεχνική που χρησιμοποιείται για την απεικόνιση κ.λ). Ακομά για την διάγνωση ασθενειών είναι σημαντικό το δίκτυο να εξάγει αξιόπιστα αποτελέσματα, με σκοπό να αποφευχθεί όσο είναι δυνατόν η πιθανότητα false negative ταξινόμησης. Η ανάπτυξη τέτοιων αξιόπιστων συστημάτων μηχανικής μάθησης για την κατηγοριοποίηση νόσων μπορεί να βοηθήσει σε μεγάλο βαθμό στην ακριβή και γρήγορη διάγνωση επικίνδυνων ασθενειών με σκοπό την όσο δυνατόν ταχύτερη προσπάθεια περίθαλψης τους.

1.1 Αντικείμενο της διπλωματικής

Η συγκεκριμένη διπλωματική εργασία έχει ως σκοπό την μελέτη του προβλήματος της ανάπτυξης αποδοτικών μοντέλων μηχανικής μάθησης για αξιόπιστη κατηγοριοποίηση ιατρικών εικόνων. Πιο συγκεκριμένα μελετήθηκε το πρόβλημα της ανάπτυξης εύρωστων δικτύων για την κατηγοριοποίηση εικόνων, που να βασίζονται σε μικρό αριθμό δεδομένων για την εκπαίδευση τους. Για την αντιμετώπιση αυτής της πρόκλησης μελετήθηκαν διάφορες τεχνικές εκπαιδευσης, που προσπαθούν να αξιοποιήσουν αποδοτικά τον περιορισμένο αριθμό δεδομένων, με σκοπό την παραγωγή μοντέλων που είναι γενικεύσιμα και μπορούν να προσαρμοστούν γρήγορα σε μια διαφορετική κατανομή δεδομένων από αυτήν που είχαν

εκπαιδευτεί με περιορισμένη εκπαίδευση. Επιπλέον εφαρμόστηκαν διάφοροι αλγόριθμοι για την εκπαίδευση μοντέλων, που θα είναι εύκολα προσαρμόσιμα σε καινούρια δεδομένα. Τέλος συγκρίθηκαν διαφορετικές αρχιτεκτονικές μοντέλων για την ταξινόμηση των ιατρικών εικόνων, όπου κάποιες αρχιτεκτονικές χρησιμοποιούν συνελικτικά νευρωνικά δίκτυα, ενώ επιπλέον δοκιμάστηκε και η χρήση αρχιτεκτονικών, που χρησιμοποιούν δίκτυα Vision Transformers.

1.2 Οργάνωση του τόμου

Η συγκεκριμένη διπλωματική εργασία οργανώθηκε σε 6 κεφάλαια, όπου στο πρώτο κεφάλαιο πραγματοποιείται μια εισαγωγή στο θέμα, που μελετήθηκε στην διπλωματική, στην συνέχεια στο κεφάλαιο 2 πραγματοποιείται μια θεωρητική ανάλυση των βασικών τεχνικών, που χρησιμοποιήθηκαν για την υλοποίηση της συγκεκριμένης διπλωματικής εργασίας. Στο κεφαλαιο 3 αναφέρθηκαν παρόμοιες μελέτες που έχουν πραγματοποιηθεί στο αντικείμενο το οποίο μελετάμε, αλλά και συγκεκριμένες μέθοδοι πάνω στις οποίες βασιστήκαμε στην παρούσα διπλωματική εργασία. Στο επόμενο κεφάλαιο παρουσιάστηκαν αναλυτικότερα τα συστήματα τα οποία χρησιμοποιήσαμε καθώς πραγματοποιήθηκε πιο λεπτομερής παρουσίαση των τρόπων λειτουργίας αυτών. Επιπλέον αναλύθηκαν και τεχνικότερα ζητήματα όπως η διαδικασια προ επεξεργασίας των δεδομένων και ο τρόπος με τον οποίο οργανώθηκαν τα δεδομένα για την πραγματοποιήση της εκπαίδευσης. Εκτος βεβαια απο τα δεδομένα που χρησιμοποιήθηκαν, έγινε παρουσίαση και των βασικών αλγορίθμων που μελετήθηκαν και ανάλυση της βιβλιογραφίας πάνω στην οποία βασίστηκαν κάποιες τεχνικές. Το κεφάλαιο 5 αποτελεί μια παρουσίαση των πειραματικών αποτελεσμάτων που προέκυψαν απο την μελέτη μας για τα διαφορετικά δίκτυα και των παρατηρήσεων που αναδύθηκαν απο την εξαγωγη αυτων των αποτελεσμάτων, επιπλέον αναφερθηκαν οι βασικές υπερ παράμετροι που χρησιμοποιήθηκαν για την εκπαίδευση των δικτύων. Τέλος στο κεφαλαιο 6 μελετήθηκαν τα βασικά συμπεράσματα αυτής της διπλωματικής εργασίας και παρουσιάστηκαν κάποιες μελλοντικές επεκτάσεις, που μπορούν να χρησιμοποιηθούν για την ανάπτυξη των μεθόδων που χρησιμοποιήσαμε.

Μέρος II

Θεωρητικό Μέρος

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό θα πραγματοποιηθεί ανάλυση των θεωρητικών εννοιών πάνω στις οποίες βασίζεται η πειραματική μελέτη που πραγματοποιήθηκε. Αρχικά θα αναλυθούν γενικότερες έννοιες για την τεχνητή νοημοσύνη και στην συνέχεια πιο εξειδικευμένες τεχνικές που χρησιμοποιούνται στην εκπαίδευση νευρωνικών δικτύων.

2.1 Τεχνητή Νοημοσύνη και Μηχανική Μάθηση

2.1.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη αναφέρεται στον κλάδο της επιστήμης που ασχολείται με την ανάπτυξη ευφυών υπολογιστικών συστημάτων. Ευφυή είναι τα συστήματα τα οποία μιμούμενα την ανθρώπινη αντίληψη, γνώση και συμπεριφορά μπορούν να λάβουν αποφάσεις και να εξάγουν συμπεράσματα. Η τεχνητή νοημοσύνη εκτείνεται σε πολλούς κλάδους και βρίσκει εφαρμογή σε όλο και περισσότερους τομείς τα τελευταία χρόνια. Στην συγκεκριμένη μελέτη πραγματοποιήθηκε εφαρμογή της τεχνητής νοημοσύνης στην ανάλυση και επεξεργασία εικόνας στον επιστημονικό τομέα της ιατρικής.

2.1.2 Μηχανική Μάθηση

Η Μηχανική μάθηση αποτελεί ένα τομέα της τεχνητής νοημοσύνης, ο οποίος περιγράφει συστήματα τα οποία εκτελούν εργασίες με ευφυή τρόπο και βελτιώνουν την επίδοση τους με επαναλαμβανόμενο αριθμό επαναλήψεων. Ενώ οι μέθοδοι τεχνητής νοημοσύνης βασίζονται σε αυστηρούς και πολύπλοκους κανόνες τους οποίους το πρόγραμμα ακολουθεί για να δράσει, οι αλγόριθμοι μηχανικής μάθησης βασίζονται στον όγκο των δεδομένων στα οποία εκπαιδεύονται και μαθαίνουν από αυτά. Ο τρόπος που πραγματοποιείται η μάθηση από αυτά τα δεδομένα είναι συνήθως η αναγνώριση προτύπων και μοτίβων που παρατηρείται σε ένα σύνολο δεδομένων, που ανήκει στην ίδια κατηγορία. Στην συνέχεια θα αναλύσουμε τρεις τεχνικές μηχανικής μάθησης την επιβλεπόμενη μάθηση (supervised learning)[11],μη επιβλεπόμενη μάθηση (unsupervised learning)[12], αυτό-επιβλεπόμενη μάθηση (self supervised learning)[13].

Επιβλεπόμενη μάθηση(supervised learning)[11]

Οι αλγόριθμοι επιβλεπόμενης μάθησης αφορούν τους αλγόριθμους, οι οποίοι εκπαιδεύονται σε δεδομένα τα οποία συνοδεύονται επίσης από ετικέτες, οι οποίες παρέχουν πληροφορίες σχετικά με την σωστή ταξινόμηση των δεδομένων. Τα συστήματα επιβλεπόμενης μάθησης μαθαίνουν να εκτελούν μια αντιστοίχιση μεταξύ ενός συνόλου μεταβλητών εισόδου J σε μια μεταβλητή εξόδου C και εφαρμόζουν αυτή την αντιστοίχιση για την πρόβλεψη των εξόδων σε νέα δεδομένα. Αυτού του είδους η μάθηση χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης, τα οποία έχουν σκοπό την κατηγοριοποίηση των δεδομένων εισόδου, αλλά και σε προβλήματα παλινδρόμησης στα οποία προσεγγίζεται μια συνεχής τιμή, όπως η πρόβλεψη της μελλοντικής τιμής ενός προϊόντος.

Μη επιβλεπόμενη μάθηση (unsupervised learning)[12]

Σε αντίθεση με τα συστήματα επιβλεπόμενης μάθησης, στην μη επιβλεπόμενη μάθηση τα δεδομένα εισόδου δεν συνοδεύονται από ετικέτες. Οι αλγοριθμοι μη επιβλεπόμενης μαθησης καλούνται να αναγνωρίσουν πρότυπα σε δεδομένα χωρίς να έχουν καμια προηγούμενη πληροφορία και καθοδήγηση σε αυτά συνεπώς το σύστημα καλείται, να αναγνωρίσει συσχετίσεις μεταξύ των δεδομένων. Τέτοιου είδους αλγόριθμοι χρησιμοποιούνται σε προβλήματα ομαδοποίησης, συσχετισμών και μείωσης διαστάσεων. Κάποιοι από αυτούς τους αλγόριθμους είναι ο K-means, K-NN (k nearest neighbors) και PCA (Principal Component Analysis)

Αυτό-επιβλεπόμενη μάθηση (self supervised learning)[13]

Η τεχνική της αυτό-επιβλεπόμενης μάθησης αποτελεί μια κατηγορία της μη-επιβλεπόμενης μάθησης καθώς το σύστημα εκπαιδεύεται σε ένα σύνολο δεδομένων χωρίς να έχει τις ετικέτες για αυτά. Με αυτό τον τρόπο το μοντέλο μαθαίνει πιο λεπτομερείς αναπαραστάσεις των δεδομένων σε σχέση με αυτές που δίνονται από τον απλό διαχωρισμό των δεδομένων. Αφου το μοντέλο εκπαιδευτεί και μάθει πρότυπα για αυτά τα δεδομένα που δεν έχουν ετικέτες. Στην συνέχεια χρησιμοποιείται, για να κατηγοριοποιήσει δεδομένα χρησιμοποιώντας τις δοσμένες ετικέτες. Για καλύτερη κατανόηση της τεχνικής της αυτό-επιβλεπόμενης μάθησης μπορούμε να την χωρίσουμε στα παρακάτω βήματα

- Δημιουργία των δεδομένων εισόδου και των ετικετών τους
- Pre-training: εκπαίδευση του μοντέλου με τα δεδομένα και ετικέτες του προηγούμενου βήματος.
- Fine-tune: αρχικοποιήση του μοντέλου με τις προεκπαιδεύμενες παραμέτρους του pre-trained μοντέλου και εκπαίδευση του στα δεδομένα ενδιαφέροντος

Η μέθοδος της αυτό-επιβλεπόμενης μάθησης βοηθά το μοντέλο στο να αποκτήσει περισσότερες χρήσιμες πληροφορίες για κάθε δείγμα σε σχέση με την απλή επιβλεπόμενη μάθηση. Αυτό συμβαίνει διότι συνήθως οι ετικέτες, που δίνουν οι ανθρωποι στα δεδομένα επικεντρώνονται στα πιο χαρακτηριστικά στοιχεία των δεδομένων με αποτέλεσμα πιο μικρές

λεπτομέρειες που τα απεικονίζουν να χάνονται. Για παράδειγμα αν θεωρήσουμε ότι τα δεδομένα, στα οποία θέλουμε να εκπαιδεύσουμε το μοντέλο είναι εικόνες που αναπαριστούν ζώα τότε οι ετικέτες θα δίνουν πληροφορίες μόνο σχετικά με το ζώο που απεικονίζεται και θα χάνονται λεπτομέρειες σχετικά με το περιβαλλον που απεικονίζεται στην εικόνα. Με την αυτο-επιβλεπόμενη μάθηση όμως τα δεδομένα χωρίζονται σε πιο λεπτομερείς κατηγορίες, έτσι το μοντέλο μπορεί να εξάγει περισσότερες και πιο λεπτομερείς πληροφορίες σχετικά με τα δείγματα. Αυτό βοηθάει το μοντέλο να μάθει πιο περίπλοκες αναπαραστάσεις των δεδομένων. Γενικώς χρησιμοποιείται για να παράξει μοντέλα που μπορούν να γενικεύσουν καλύτερα, όταν προσαρμόζονται σε νέες αναπαραστάσεις δεδομένων.

2.2 Νευρωνικά Δίκτυα

2.2.1 Βιολογικά Νευρωνικά Δίκτυα

Ο ανθρώπινος εγκέφαλος αποτελείται από εκατομμύρια νευρώνες, οι οποίοι αποτελούν κύτταρα που είναι συνδεδεμένα και επικοινωνουν μεταξύ τους. Κάθε νευρώνας αποτελείται από το κυρίως σώμα τον άξονα και τους δενδρίτες. Οι νευρώνες ανταλλάσουν σήματα ηλεκτρικής μορφής ο ένας με τον άλλον μεσω νευροδιαβιβαστών. Οι νευρώνες βρίσκονται είτε σε ενεργή είτε σε ανενεργή κατάσταση έτσι μπορούμε να θεωρήσουμε ότι ακολουθούν έναν δυαδικό τρόπο λειτουργίας. Όταν ο νευρώνας βρίσκεται σε ενεργή κατάσταση τότε παράγει ηλεκτρικό παλμό. Σε έναν νευρώνα μπορούν να φτάνουν πολλά σήματα μεσω των συνάψεων αυτά τα ηλεκτρικά σήματα αθροίζονται και αν ξεπερνούν ένα κατώφλι τότε ο νευρώνας ενεργοποιείται και παράγει έναν παλμό. Σε αντίθετη περίπτωση δεν παράγει κάποιο σήμα και το δυναμικό χάνεται. Κάθε σήμα που εισέρχεται σε έναν νευρώνα μπορεί να είναι είτε διεγερτικό είτε ανασταλτικό, τα όποια προσθέτουν και αφαιρούν αντίστοιχα απο το κατώφλι του δυναμικού.

2.2.2 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα αποτελούν μια μοντελοποίηση των Βιολογικών Νευρωνικών Δικτύων προσαρμοσμένων στις ικανότητες των υπολογιστικών συστημάτων. Τα τεχνητά νευρωνικά δίκτυα μοντελοποιούνται με γράφους, όπου οι νευρώνες συνδέονται μεταξύ τους και ανταλλάσουν μαθηματικά δεδομένα σε αντίθεση με τους ηλεκτρικούς παλμούς που ανταλλάσσουν τα βιολογικά νευρωνικά δίκτυα. Κάθε είσοδος του νευρώνα πολλαπλασιάζεται με ένα βάρος η τιμή του οποίου προσαρμόζεται κατα την διάρκεια της εκπαίδευσης και υποδηλώνει αν ο νευρώνας δρα ενισχυτικά ή αποσβετικά. Στην συνέχεια όλοι οι είσοδοι αθροίζονται και το άθροισμα πολλαπλασιάζεται με το bias του νευρώνα και έπειτα εισάγεται στην συνάρτηση ενεργοποίησης. Με την σειρά της η συνάρτηση ενεργοποίησης καθορίζει αν ο τεχνητός νευρώνας θα ενεργοποιηθεί ή οχι (θα δώσει ως έξοδο 0 ή 1). Οι νευρώνες του τεχνητού νευρωνικού δικτύου ακολουθούν μια παράλληλη οργάνωση που ονομάζουμε layers.

Το πρώτο layer στα νευρωνικά δίκτυα ονομάζεται layer εισόδου, ενω το τελευταίο layer ονομάζεται layer εξόδου. Το τελικό layer εξόδου δίνει μια πρόβλεψη για τα δεδομένα που εισάγονται. Στην περίπτωση του supervised learning η πρόβλεψη εξόδου συγκρίνεται με την

αναμενόμενη έξοδο και υπολογίζεται η τιμή της loss function. Η τιμή του loss function που προκύπτει προωθείται προς τα πίσω στο δίκτυο (backpropagation) και ανανεώνονται οι παράμετροι του δικτύου με βάση τον optimizer που έχει επιλεχθεί (στις περισσότερες περιπτώσεις ο optimizer ακολουθεί μια μέθοδο gradient descent και πολλαπλασιάζεται με μια σταθερά learning rate). Αυτή η διαδικασία εκπαίδευσης έχει ως στόχο να ελαχιστοποιήσει την τιμή της loss function και σταματά μετα από έναν συγκεκριμένο αριθμό επαναλήψεων που έχουμε ορίσει ή όταν το σφάλμα φτάσει κάτω απο ένα συγκεκριμένο κατώφλι.

2.2.3 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα CNNs[14] είναι μια κατηγορία νευρωνικών δικτύων, τα οποία χρησιμοποιούνται στην Βαθια μάθηση κυρίως στον τομέα της επεξεργασίας εικόνων. Πιο συγκεκριμένα τα συνελικτικά νευρωνικά δίκτυα δέχονται ως είσοδο εικόνες και συνήθως έχουν σκοπό την ταξινομηση τους. Σημαντική είναι επίσης και η ικανότητα τους να εξάγουν χαρακτηριστικά από εικόνες πράγμα που τα καθιστά χρήσιμα σε πολλές εφαρμογές, αλλά μπορεί να χρησιμοποιηθούν και ως δομικά στοιχεία για την δημιουργία πιο σύνθετων δικτύων. Τα CNNs αποτελούνται απο τρια διαφορετικά επίπεδα (layers) τα convolutional layers, τα pooling layers[15] και τα fully connected layers ως είσοδο παίρνουν συνήθως εικόνες και στην περίπτωση του προβλήματος ταξινόμησης επιστρέφουν στο τελευταίο επίπεδο μια σειρά απο πιθανότητες που δηλώνουν πόσο πιθανή είναι η είσοδος να ανήκει στην αντίστοιχη κλάση.

Τα convolutional layers εφαρμόζουν φίλτρα στην εικόνα, τα οποία πραγματοποιούν την πράξη της συνέλιξης στα δεδομένα εισόδου και εξάγουν χάρτες χαρακτηριστικών(feature maps). Στην συνέχεια αυτοί οι χάρτες χαρακτηριστικών, οι οποίοι αποτελούν πίνακες, εισάγονται στα επόμενα convolutional layers. Σε αυτούς τους πίνακες χαρακτηριστικών εφαρμόζονται επίσης συναρτήσεις ενεργοποίησης (Relu[16], Sigmoid) διοτι η συνέλιξη από μόνη της αποτελεί γραμμική πράξη. Επίσης υπάρχουν τα επίπεδα υποδειγματοληψίας (pooling layers), τα οποία έχουν ως στόχο την μείωση των διαστάσεων των πινάκων χαρακτηριστικών που παίρνουν ως είσοδο. Στόχος αυτών των επιπέδων υποδειγματοληψίας είναι να μειωθεί η ακρίβεια των χαρακτηριστικών, που έχει ως αποτέλεσμα την μείωση της ακρίβειας των διαφορών μεταξύ των εικόνων. Κάτι τέτοιο είναι χρήσιμο, γιατι εμποδίζει το δίκτυο να υπερεκπαιδευτεί σε ένα συγκεκριμένο σύνολο δεδομένων πράγμα που μας οδηγεί στο overfitting. Για να επιτευχθεί αυτό και να μειωθεί η λεπτομέρεια εξαγωγής χαρακτηριστικών μετα απο κάθε convolution layer στο CNN τοποθετείται ένα pooling layer, για να μειώσει την εξαγωγή λεπτομερών χαρακτηριστικών, που θα οδηγήσουν στο overfitting. Τα πιο συχνά χρησιμοποιούμενα pooling layers είναι το average polling και το max pooling, τα οποία παίρνουν την μέση και την μέγιστη τιμή αντίστοιχα γύρω από ένα παράθυρο που ορίζεται από τον χρήστη.

Το τελευταίο επίπεδο του CNN είναι το fully connected layer το συγκεκριμένο επίπεδο δέχεται ως είσοδο ένα μονοδιάστατο διάνυσμα, οπότε άμα το προηγούμενο επίπεδο δίνει ως έξοδο ένα πολυδιάστατο feature map, χρειάζεται να εφαρμοστεί ένα flattening για να μετατραπεί σε μονοδιάστατο. Σε αυτό το επίπεδο όλοι οι νευρώνες εξόδου συνδέονται με όλα τα στοιχεία του μονοδιάστατου διανύσματος

Τα CNN απο τις αρχικές πιο απλές εφαρμογές τους, όπως το LeNet5[17], AlexNet[18], VGG[19], GoogleNet[20] αποδείχθηκαν πολύ επιτυχή στις εφαρμογές της ταξινόμησης καθώς παρουσίασαν state of the art αποτελέσματα σε πολλά προβλήματα κατηγοριοποίησης, ξεπερνώντας άλλους τότε παραδοσιακούς αλγόριθμους ταξινόμησης. Στην συνέχεια με την εξέλιξη των CNN και την εφαρμογή τους σε δίκτυα όπως το ResNet[21] αξιοποιήθηκαν πιο σύνθετες αρχιτεκτονικές των CNN με περισσότερα layer. Τα CNN δίκτυα ακόμα και τώρα πετυχαίνουν αποδοτικά αποτελέσματα με χρήση μεθόδων όπως το NFNet[22], ConvNext[23], ResNest[24].

2.2.4 Batch Normalization

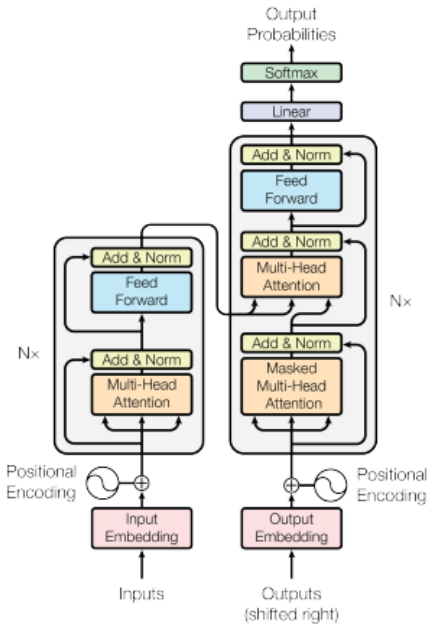
Το Batch Normalization[25] αποτελεί μια μέθοδο, που χρησιμοποιείται κατά την διάρκεια εκπαίδευσης των νευρωνικών δικτύων και έχει ως σκοπό να κάνει την εκπαίδευση πιο ευσταθή και πιο γρήγορη μέσω κανονικοποίησης των επιπέδων. Ο τρόπος με τον όποιο κανονικοποιεί τα δεδομένα είναι να τα επανκλιμακοποιεί και να τους επαναρυθμίζει το κέντρο. Η τεχνική του batch normalization εφαρμόζεται μόνο κατα την διάρκεια της εκπαίδευσης, ενώ απενεργοποιείται κατα την διάρκεια της επικύρωσης (validation) και του ελέγχου (test). Το batch normalization χρησιμοποιείται κυρίως για να αντιμετωπίσει το internal covariate shift, δηλαδή την πιθανή μετατόπιση που μπορεί να έχει η κατανομή των δεδομένων που εισαγονται ως batches στο δίκτυο, κάτι τέτοιο επιτυγχάνεται μέσω της κανονικοποίησης των δεδομένων αυτών.

2.2.5 Transformers

Οι Transformers[8] είναι μοντέλα νευρωνικών δικτύων, τα οποία παρατηρώντας τις σχέσεις μεταξύ σειριακών δεδομένων μπορούν να εξάγουν πληροφορίες σχετικά με νοηματικές συνδέσεις μεταξύ των δεδομένων. Οι αρχιτεκτονικές αυτές βασίζονται σε ένα σύνολο εξελισσόμενων μαθηματικών τεχνικών που ονομάζονται attention ή self-attention. Αυτές οι τεχνικές βοηθούν στην εξαγωγή πληροφοριών σχετικά με το πως μακρινα στοιχεία σε σειριακά δεδομένα μπορούν να επηρεάσουν το ένα το άλλο.

Οι Transformers σε πολλά προβλήματα αντικαθιστούν τα συνελικτικά νευρωνικά δίκτυα CNN και τα επαναλαμβανόμενα νευρωνικά δίκτυα RNN, τα οποία αποτελούσαν τα πιο διάσημα νευρωνικά δίκτυα τα προηγούμενα χρόνια. Πιο συγκεκριμένα τα τελευταία χρόνια το 70% των paper που δημοσιεύονται στο arXiv αφορούν transformers[26].

Ενα απο τα βασικά πλεονεκτήματα των transformers είναι ότι εισήγαγαν την εκπαίδευση χωρίς την ανάγκη ετικετών. Πριν απο τους transformers για την εκπαίδευση μοντέλων χρειαζόντουσαν μεγάλα σύνολα δεδομένων τα δείγματα των οποίων είχαν ετικέτες, κάτι το οποίο χρειάζονταν χρόνο και ήταν ακριβό. Οι transformers με την ικανότητα τους να βρίσκουν μαθηματικά πρότυπα μεταξύ των αντικειμένων δίνουν αυτόματα πρόσβαση σε ένα μεγάλο σύνολο από δεδομένα. Λόγω αυτής της ικανότητας τους μπορούν να χρησιμοποιηθούν τεχνικές εκπαίδευσης, που χειρίζονται μεγαλύτερο όγκο δεδομένων χωρίς την ανάγκη καλώς ορισμένων ετικετών, όπως η μέθοδος της αυτο-επιβλεπόμενης μάθησης που αναλύθηκε στο 2.1.2

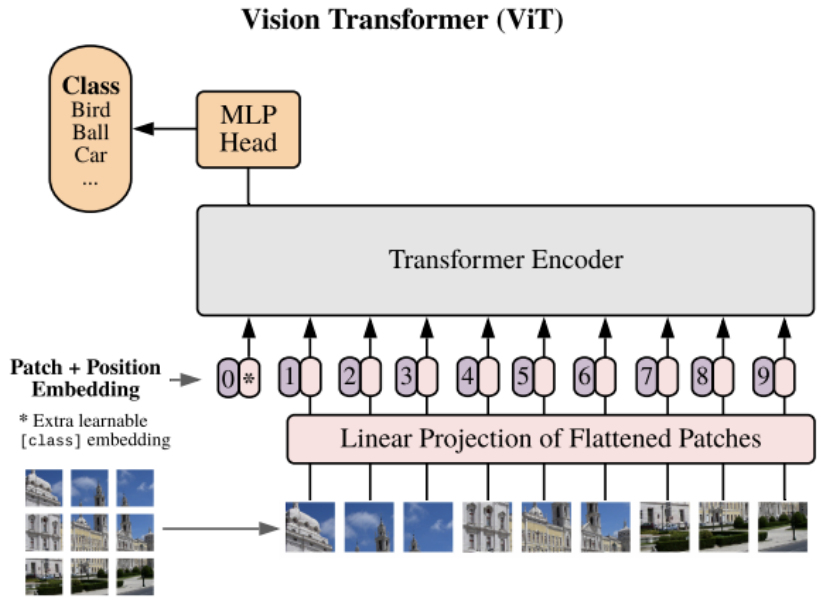


Σχήμα 2.1: *Transformer architecture [8]*

Vision Transformers

Οι Transformers αρχικά χρησιμοποιήθηκαν στον τομέα της επεξεργασίας φυσικής γλώσσας NLP και έδωσαν πολύ καλά αποτελέσματα όπως το γλωσσικό μοντέλο BERT[27] και GPT[28].

Όπως περιγράφηκε προηγουμένως οι transformers υπολογίζουν την σχέση μεταξύ ζευγαριών από στοιχεία που εισάγονται στο σύστημα αυτό πραγματοποιείται με τον μηχανισμό attention. Για τις εικόνες αυτά τα στοιχεία (tokens) είναι τα pixel, ωστόσο γίνεται αντιληπτό ότι ο υπολογισμός σχέσεων μεταξύ όλων των pixel είναι κάτι το οποίο υπολογιστικά είναι πολύ ακριβό και δεν είναι δυνατή η διάθεση της αναγκαίας μνήμης για την πραγματοποίηση αυτού. Για αυτό οι ViT (vision transformers) χωρίζουν τις εικόνες σε μικρότερα τμήματα και υπολογίζουν τις σχέσεις μεταξύ αυτών των τμημάτων της εικόνας, τα μικρότερα αυτά τμήματα εικόνων στην βιβλιογραφια αναφέρονται ως patches. Αυτός ο τρόπος είναι προφανές ότι μικραίνει το υπολογιστικό κόστος σε πολύ μεγάλο βαθμό. Αφου η εικόνα χωριστεί σε τμήματα, αυτά τοποθετούνται στην σειρά και στην συνέχεια κάθε τμήμα της εικόνας οργανώνεται σε μια γραμμική σειρά και πολλαπλασιαζεται απο το αντίστοιχό embedding (τα embedding αποτελούν διανύσματα τα οποία μαθαίνει το μοντέλο κατα την διάρκεια της εκπαίδευσης). Τέλος αυτά τα γραμμικοποιημένα και πολλάπλασιασμένα με τα embeddings τμήματα patches της εικόνας τροφοδοτούνται στην είσοδο του transformer.



Σχήμα 2.2: *ViT Transformers [9]*

Αναλύοντας πιο διεξοδικά τους vision transformers θεωρούμε εικόνες με ύψος H και πλάτος W και τις χωρίζουμε σε patches. Μετα τον χωρισμό της εικόνας σε μικρτότερα τμήματα προκύπτει $N = \frac{H \cdot W}{P^2}$ σε αριθμό patches, με κάθε patch να έχει διαστάσεις (*PxP*). Πριν αυτά τα μικρότερα τμήματος της εικόνας τροφοδοτηθούν εντός του Vision Transformer αναδιοργανόνονται ακολουθιακά, ετσι ώστε να προκύψουν διανύσματα x_p^n τα οποία θα έχουν διαστάσεις (P^2xC), όπου $n = 1, ...N$. Αφου τα δεδομένα εισαχθούν στο σύστημα του transformer δημιουργείται μια ακολουθία απο embedded patches τα οποία έχουν διάσταση D. Επιπλέον δημιουργείται και ενα εκπαιδεύσιμο class embedding το οποίο συνήθως συμβολίζεται με [CLS] και τοποθετείται στην αρχή της ακολουθίας των embeddings.Η τιμή αυτου του embedding x_{class} χρησιμοποιείται για να δηλώνει το label της ταξινόμησης. Στο τελευταίο στάδιο τα embeddings των τμημάτων της εικόνας ενισχύονται με μονοδιάστατα embeddings θέσης τα οποία ανανεόνονται κατα την διάρκεια της εκπαίδευσης και έχουν ως σκοπό να εξάγουν πληροφοριά σχετικά με τις συσχετίσεις που έχουν patches, τα οποία βρίσκονται σε απομακρυσμένες θέσεις μεταξύ τους. Τελικά τα embeddings που προκύπτουν μπορούν να συμβολιστούν με τον παρακάτω τρόπο

$$z_0 = [x_{class} : x_p^1 E; ..; x_p^N E] + E_{pos} \tag{2.1}$$

Για την πραγματοποίηση της ταξινόμησης τα embeddings που προκύπτουν τροφοδοτουνται στο Transformer που αποτελείται από ακολουθιακά συνδεδεμένα layer που είναι ίδια μεταξύ τους. Στην συνέχεια επιστρέφεται η τιμή από το τελευταίο layer και τροφοδοτείται σε ενα component, που ονομάζεται head το οποίο πραγματοποιεί την ταξινόμηση. Τα classification heads συνήθως αποτελούνται απο multi layer perceptrons και εφαρμόζουν Gaussian Error Linear Unit (GELU)[29].

Πιο συνοπτικά ο ViT δέχεται μια ακολουθία από embedded τμήματα εικόνων (patches)

τα οποία χωρίστηκαν πριν την δημιουργία των embeddings. Εκτός από τα embeddings των patches στην αρχή της ακολουθίας αυτών τοποθετείται ένα embedding, το οποίο χρησιμοποιείται για το classification. Επιπλέον προστίθεται ένα embedding, που περιέχει πληροφορίες για τις θέσεις των τμημάτων της εικόνας.

Εκτός απο την κλασσική υλοποίηση των Vision Transformers όπου οι αρχικές εικόνες χωρίζονται σε patches και εισάγονται στο δίκτυο του ViT, έχουν αναπτυχθεί και υβριδικές τεχνικές που αξιοποιούν συνελικτικά νευρωνικά δίκτυα. Πιο συγκεκριμένα οι αρχικές εικόνες εισάγονται σε αυτά τα δίκτυα απο τα οποια εξαγονται χαρακτηριστικά. Στην συνέχεια αυτοι οι χάρτες χαρακτηριστικών, που εξάγονται από τα CNN χωρίζονται στα patches και από εκεί και έπειτα ακολουθείται η ίδια διαδικασία που περιγράφηκε και προηγουμένως. Τα ViT δίκτυα αρχικά προ εκπαιδεύονται σε μεγαλύτερα datasets, όπως το (ImageNet, JFT-300M)[30] και στην συνέχεια πραγματοποιείται fine-tuning σε αυτά με χρήση μικρότερου αριθμού κλάσεων.

Κατά την διάρκεια του fine-tuning το MLP δίκτυο αντικαθίσταται από ένα feed forward layer το οποίο έχει διαστάσεις DxK με D να είναι το embedding dimension και το K δηλώνει τον αριθμό των κλάσεων. Κατά την διάρκεια του fine-tuning σε πολλές περιπτώσεις χρησιμοποιούνται εικόνες μεγαλύτερης διάστασης απο οτι στο στάδιο του pre-train κάτι που προφανώς οδηγεί σε μεγαλύτερες ακολουθίες εικόνων και στην ύπαρξη περισσότερων positional embeddings.

Το βασικό πλεονέκτημα των Vision Transformers σε σχέση με τα συνελικτικά νευρωνικά δίκτυα είναι ότι με την χρήση των self attention layers μπορούν να εξάγουν περιγραφές για όλο το εύρος της εικονας και δεν είναι περιορισμένοι απο τα γειτονικά στοιχεία στις δύο διαστάσεις τους όπως τα CNN.

Βέβαια τα μοντέλα των Vision Transformers χρειάζονται μεγάλο όγκο δεδομένων εκπαίδευσης, για να εξάγουν αποτελέσματα, τα οποία θα είναι καλύτερα από τις μεθόδους CNN που έχουν αναπτυχθεί. Σε αντίθετη περίπτωση οταν το σύνολο των δεδομένων είναι μικρό τα μοντέλα ViT τείνουν, να επιστρέφουν χαμηλότερα αποτελέσματα σε σχέση με αυτά, που αξιοποιούν τα μοντέλα CNN.

Attention/Self-Attention Mechanism

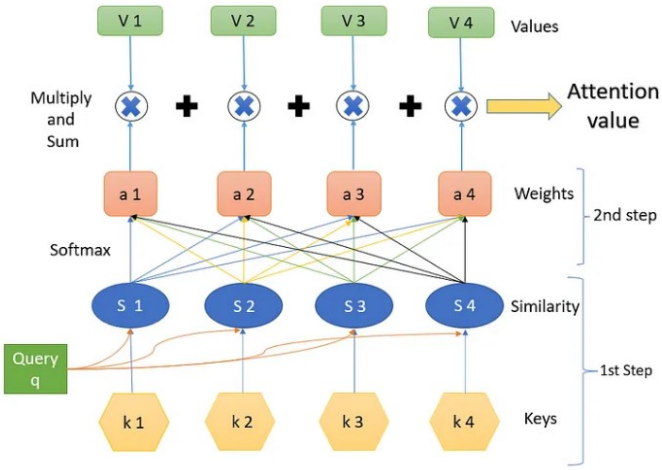
Οι μηχανισμοί Attention και Self Attention επιτρέπουν στα μοντέλα transformer, να δίνουν προσοχή σε διαφορετικά μέρη στα σειριακά δεδομένα(sequences) όταν πραγματοποιούν προβλέψεις. Οι transformers είναι μοντέλα encoder/decoder που επεξεργάζονται δεδομένα και πιο συγκεκριμένα χρησιμοποιούν positional encoders για να κρατήσουν αναφορές για τα δεδομένα που εισάγονται και εξάγονται από το δίκτυο. Οι μηχανισμοί Attention ακολουθούν αυτές τις επισημανσεις και υπολογίζουν αλγεβρικούς χάρτες για το πως τα στοιχεία σχετίζονται με άλλα.

Για τον υπολογισμό του Attention θεωρούμε αρχικά τα embeddings των δεδομένων τα οποία τα συμβολίζουμε ως $V_1, ..V_N$. Στην συνέχεια υπλογίζονται οι παράμετροι που δείχνουν τις ομοιότητες μεταξύ των διαφορετικών δεδομένων αυτός ο υπολογισμός συνήθως πραγματοποιείται μεσω του εσωτερικού γινομένου μεταξύ των embeddings και των δεδομένων, δηλαδή μπορούμε να συμβολίσουμε αυτές τις παραμέτρους ως $W_{11} = V_1 \cdot V_1, W_{12} =$

$V_1 \cdot V_2, ..., W_{1N} = V_1 \cdot V_N$. Αυτοι οι υπολογισμοί εφαρμόζονται για όλους τους συνδυασμους embeddings και στην ουσία αποτελούν μια μετατροπή των υπολοίπων παράμετρων των δεδομένων ως προς το embedding με το οποίο πολλαπλασιάζονται. Συνεπώς, όσο μεγαλύτερη είναι η τιμή αυτών, τόσο μεγαλύτερη σχέση έχουν αυτά τα δεδομένα μεταξύ τους. Επιπλέον για την πιο ευκολή εξαγωγή πληροφοριών εφαρμόζεται μια κανονικοποίηση στις παραμέτρους $W_{iN}, ..., W_{iN}, i = 1, ..N$ ώστε να αθροίζουν στο 1. Στο τελευταίο στάδιο αυτοι οι παράμετροι πολλάπλασιαζονται με τα αρχικά embeddings και στην συνέχεια προστίθενται μεταξύ τους (αυτό πραγματοποιείται για κάθε ενα απο τα δεδομένα) και προκύπτουν οι τιμες $Y_1 = W_{11} \cdot V_1 + W_{12} \cdot V_2 +, ..., W_{1N} \cdot V_N$. Οι τιμές $Y_1, .., Y_N$ παρέχουν περισσότερες πληροφορίες για το πως σχετίζονται τα δεδομένα μεταξύ τους, που δεν είναι γειτονικά. Αξίζει να σημειωθεί οτι σε αυτήν την μέθοδο δεν υπάρχουν εκπαιδεύσιμες παράμετροι, αλλά μπορούν να προσθεθούν μέσω του πολλαπλασιασμού διανυσμάτων, οι οποίες θα είναι ικάνες να μάθουν πρότυπα που προσφέρουν περισσότερες πληροφορίες για τις σχέσεις μεταξύ των δεδομένων. Για την εφαρμογή αυτών των εκπαιδεύσιμων παραμέτρων εισήχθησαν οι όροι Query, Key, Value στους μηχανισμούς attention.

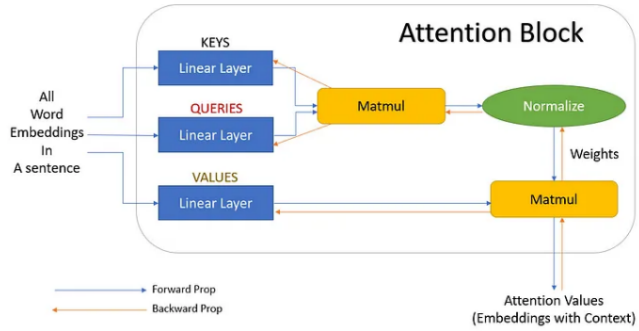
Αυτοι οι τρεις όροι δεν είναι τίποτα παραπάνω απο διαφορετικοί συμβολισμοί των embeddings των δεδομένων που χρησιμοποιήθηκαν παραπάνω για τον υπολογισμό των παραμέτρων. Αρχικά θεωρουμε V_i το embedding πάνω στο οποίο θέλουμε να αναπαραστίσουμε την ομοιότητα μεταξύ των embeddings, των υπολοιπών δεδομένων. Στην συγκεκριμένη περίπτωση αυτό το embedding V_i αναπαρίσταται ως το Query. Στην συνέχεια αυτό το Query embedding πολλαπλασιάζεται, όπως και στα προηγούμενα βήματα με όλα τα embeddings των υπολοιπών δεδομένων, αλλά και με το ίδιο embedding, ώστε να εξάγει τις παραμέτρους weights. Αυτά τα embeddings $(V_1, ..V_N)$ με τα οποία πολλαπλασιάζεται ονομάζονται Key embeddings. Στην συνέχεια με τον ίδιο τρόπο που περιγράφηκε στην διαδικάσια του απλού attention, αυτές οι παράμετροι που προκύπτουν (weights) πολλαπλασιάζονται με τα Values. Έπειτα οι τιμές που υπολογίζονται απο τους αντίστοιχους πολλαπλασισμούς προστίθενται μεταξύ τους και δίνουν το τελικό αποτέλεσμα. Σε αυτήν την διαδικασία μπορούν να προστεθούν εκπαίδεύσιμες παράμετροι, ο τρόπος με τον οποίο πραγματοποιείται η εισαγωγή αυτών είναι με πολλαπλασιασμό των embeddings με πίνακες που θα περιέχουν τις παραμέτρους. Αφου τα embeddings έχουν διαστάσεις $1xD$ ο πολλαπλασιασμός τους απο δεξια με πίνακες MK που έχουν διαστάσεις DxD θα επιστρέψει διανύσματα διαστασεών $(1xD)$. Έτσι καθε ενα απο τα δεδομένα που αποτελούν τα Query, Key, Value πολλαπλασιάζονται με τους αντίστοιχούς πίνακες Mk, Mq, Mv που έχουν διαστάσεις DxD.

Απο το διάγραμμα 2.3 μπορεί να γίνουν κατανοητά τα αλγοριθμικά βήματα που ακολουθούνται για την εξαγωγή των attention. Όπου αρχικά αναπαρίστανται τα keys αλλά και το Query, αυτά στην συνέχεια τροφοδοτούνται σε μια συνάρτηση η οποία επιστρέφει ως έξοδο την ομοιότητα μεταξύ αυτών των embeddings. Στις περισσότερες μεθόδους υλοποίησης attention μηχανισμών, αυτή η πληροφορία εξάγεται με την εφαρμογή εσωτερικού γινομένου μεταξύ του query και του key $(q^T \cdot k_i)$. Σε άλλες περιπτώσεις αυτή η τιμή μπορεί να κανονικοποιείται διαιρώντας με την τετραγωνική ρίζα των διαστάσεων καθε κλειδιού. Στο επόμενο βήμα εφαρμόζεται μια softmax συνάρτηση με σκοπό να αναπαραστήσει ως πιθανότητες τις τιμές που έχουν προκύψει για τις παραμέτρους weights. Αυτές οι παράμετροι που προκύπτουν μετα την εφαρμογή του softmax πολλαπλασιάζονται με τα αντίστοιχα em-



Σχήμα 2.3: *Attention Value pipeline [10]*

beddings των δεδομένων και έπειτα προστίθενται μεταξύ τους, ώστε να υπολογιστεί η τελική τιμή attention value για το συγκεκριμένο query.

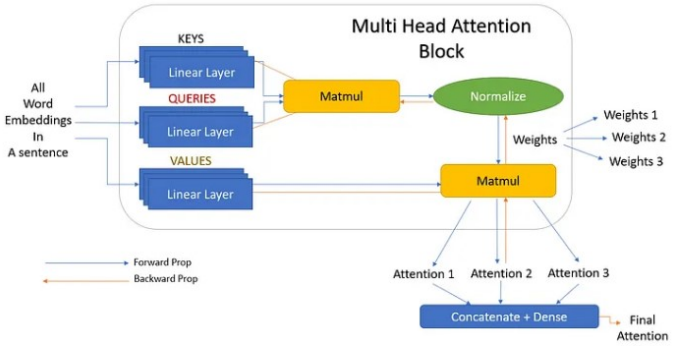


Σχήμα 2.4: *Attention Block [10]*

Η υλοποίηση του Attention Mechanism στα πλαίσια ανάπτυξης νευρωνικών δικτύων φαίνεται στο 2.4, όπου τα embeddings περνάνε απο Linear layers, τα οποία λειτουργούν ως πολλαπλασιαστές πινάκων. Στην συνέχεια το καθε ενα απο αυτά τα layers ορίζεται ως keys, queries, values. Στο επόμενο βήμα εφαρμόζεται ενας πολλαπλασιασμός πινάκων μεταξύ των keys,queries και τα αποτελέσματα τροφοδοτούνται σε μια συνάρτηση normalization(π.χ. σοφτμαξ). Οι παράμετροι που επιστρέφονται μετα απο την εφαρμογή αυτής της συνάρτησης ονομάζονται weights. Τα weights αυτά πολλαπλασιάζονται με τα values και αθροίζονται ώστε να προκύψει το τελικό διάνυσμα για το attention. Η εκπαίδευση των παραμέτρων μπορεί να πραγματοποιηθεί με backpropagate υπολογίζοντας το gradient με στόχο την εκπαίδευση του Attention block.

Το Multi head Attention block αποτελεί μια υποκατηγορία των απλών Attention μηχανισμών η οποία βοηθάει στην επίλυση προβλημάτων που αντιμετωπίζουν τα απλα δίκτυα attention. Πιο συγκεκριμένα οι μηχανισμοι Multi-Head Attention[31] μπορούν να εξερευνήσουν περισσότερες σχέσεις μεταξύ των δεδομένων απο οτι ενα απλο σύστημα Attention. Η μεγάλη διαφορα της υλοποίησης τους σε σχέση με την αρχιτεκτονική που φαίνεται στο 2.4

είναι οτι για την εξαγωγή των στοιχείων Queries, Keys, Values χρησιμοποιούνται πολλαπλά γραμμικά layer αντι για ένα. Αυτά τα layer εκπαιδεύονται παράλληλα και αποτελούνται από ανεξάρτητες παραμέτρους μεταξύ τους. Έτσι κάθε ένα από τα keys, queries, values δίνουν πολλαπλές εξόδους αντι για μια. Αύτες οι πολλαπλές έξοδοι (Attention Values) στο τελικό στάδιο συνενώνονται μεταξύ τους και αφου περάσουν από ένα τελευταίο πλήρως συνδεδεμένο layer επιστρέφουν το τελικό Attention value. Στο 2.5 φαίνεται μια αναπαράσταση του multi-head Attention block.



Σχήμα 2.5: *Multi Head Attention Block [10]*

2.3 Πρόβλημα Ταξινόμησης

Ενα απο κυρια προβλήματα που καλείται να λύσει ένα νευρωνικό δίκτυο είναι αυτό της ταξινόμησης, στο συγκεκριμένο πρόβλημα δίνονται δεδομένα τα οποία ανήκουν σε μια κατηγορια και μετα απο εκπαίδευση του δίκτυο καλείται να κατηγοριοποιήσει άγνωστα δεδομένα σε αυτές τις κατηγορίες. Στόχος λοιπόν αυτού του προβλήματος είναι να αντιστοιχήσουμε κάθε εικόνα σε μια ετικέτα (labels), το δίκτυο εξάγει ένα σύνολο από πιθανότητες οι οποίες αντιστοιχούν στην πιθανότητα η εικόνα να ανήκει στην κάθε κλάση και επιλέγεται η μεγαλύτερη από αυτές ως η τελική ταξινόμηση του δεδομένου εισόδου.

2.4 Επαύξηση Δεδομένων Data Augmentation

Η επαύξηση δεδομένων είναι μια τεχνική που συνήθως χρησιμοποιείται σε σύνολα δεδομένων τα οποία δεν έχουν επαρκή αριθμό δειγμάτων για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης. Σκοπός αυτής της τεχνικής είναι η δημιουργία πρόσθετων δεδομένων εκπαίδευσης απο τα υπάρχοντα παραδείγματα. Ο τρόπος που επιτυγχάνεται αυτό, είναι με την εφαρμογή τυχαίων μετασχηματισμών όπως περιστροφή, περικοπή, προσθήκη θορύβου, αλλαγή χρώματος κ.λ.. Αφού εφαρμοστεί ενας ή ενα σύνολο απο αυτούς τους μετασχηματισμούς σε μια εικονα, προκύπτει ένα καινούριο δείγμα που ταιριάζει στο σύνολο των δεδομένων. Τέτοιες τεχνικές βοηθούν και στην δημιουργία ενός πιο σύνθετου συνόλου δεδομένων με πιο δύσκολα δείγματα, το οποίο με την σειρά του οδηγεί σε ένα πιο καλά εκπαιδευμένο μοντέλο, το οποίο έχει καλύτερη ικανότητα γενίκευσης.

Η επαύξηση δεδομένων αποτελεί βασικό εργαλείο στην αντιμετώπιση της υπερεκπαίδευσης overfitting, η οποία εμφανίζεται όταν το σύνολο των δεδομένων δεν είναι επαρκές για την

εκπαίδευση ή όταν απαρτίζεται από όμοια δεδομένα εκπαίδευσης. Ετσι με την εφαρμογή τροποποιήσεων στα χαρακτηριστικά των εικόνων όπως στην κλίση,στην ανάλυση, στον προσανατολισμό μπορούν, να προκύψουν δείγματα τα οποία έχουν σημαντικές διαφορές μεταξύ τους. Εκτός από την απλή επαύξηση δεδομένων, μπορούν να εφαρμοστούν και πιο σύνθετες τεχνικές που παράγουν δείγματα, που διαφερουν σε σημαντικό βαθμό από την κατανομή του συνόλου δεδομένων και βοηθουν το μοντέλο να εκπαιδευτεί σε πιο γενικές και σύνθετες αναπαραστάσεις κάτι το οποίο οδηγεί σε καλύτερα γενικεύσιμα συστήματα[32, 33]. Τέτοιες τεχνικές είναι το cutout, cutmix, mixup και pixmix, που θα εξηγηθούν με παραπάνω λεπτομέρεια στην συνέχεια.Άλλες τεχνικές που χρησιμοποιήθηκαν ήταν το StyleGuide[34], όπου στην συγκεκριμένη μέθοδο χρησιμοποιείται το content style decomposition, με το οποίο δημιουργούνται ψεύτικές εικόνες που οδηγούνται προς ένα συγκεκριμένο πρότυπο style

2.5 Τεχνικές Εκπαίδευσης

2.5.1 Few shot learning

Τα νευρωνικά δίκτυα έχουν αποδειχθεί πολύ χρήσιμα τα τελευταία χρόνια στην επίλυση προβλημάτων και ιδιαίτερα στην κατηγοριοποίηση δεδομένων, ωστόσο υπάρχουν και κάποια προβλήματα, τα οποία συναντώνται κατα την διάρκεια της εκπαίδευσης. Ενα απο τα σημαντικότερα προβλήματα είναι η ανάγκη για μεγάλο όγκο δεδομένων σε κάθε κατηγορία (class), ώστε να εκπαιδευτεί το δίκτυο σωστά και να επιτύχει ορθή κατηγοριοποίηση των δεδομένων. Παρόλα αυτα σε αρκετά προβλήματα δεν υπάρχει ο αναγκαίος όγκος δεδομένων σε κάθε κλάση, ώστε να πραγματοποιηθεί σωστή κατηγοριοποίηση των δεδομένων. Για την αντιμετώπιση αυτου του προβλήματος έχουν αναπτυχθεί τεχνικές εκπαιδευσης του δικτύου σε μικρότερο σύνολο δειγμάτων. Μια απο αυτές τις τεχνικές είναι το Few shot learning[35] στην συγκεκριμένη μέθοδο το δίκτυο εκπαιδεύεται σε μια σειρά από διαφορετικά tasks. Καθε task ορίζεται από τον αριθμό των κλάσεων και τον αριθμό των δεδομένων που περιέχει. Πιο συγκεκριμένα για τον ορισμό k-way n-shot tasks συλλέγουμε όλα τα διαθέσιμα δεδομένα και βρίσκουμε τις κλάσεις αυτών. Στην συνέχεια επιλέγουμε τυχαία k κλάσεις από το σύνολο δεδομένων και n τυχαία δείγματα από αυτές τις κλάσεις. Για κάθε task επιλέγουμε διαφορετικές κλάσεις και διαφορετικά δείγματα, στην συνέχεια το δίκτυο εκπαιδεύεται για κάθε ένα από αυτά τα tasks ξεχωριστά. Επίσης η τεχνική του Few shot learning είναι χρήσιμη, οταν τα δεδομένα στα οποία εκπαιδεύουμε το δίκτυο αντλούνται από διαφορετικό χώρο κατάστασης και έχουν διαφορετική κατανομή σε σχέση με τα δεδομένα που χρησιμοποιούνται για τον έλεγχο και την επαλήθευση.

2.5.2 Transfer learning

Μια ακόμα μέθοδος προσαρμογής του δικτύου σε νέα δεδομένα είναι η τεχνική του transfer learning[36]. Η τεχνική του transfer learning έχει ως πρότυπο την ικανότητα του ανθρώπινου οργανισμού να προσαρμόζει πρότερη γνώση, που είχε αποκτήσει σε καινούργια προβλήματα. Με τον ίδιο τρόπο αντι η εκπαίδευση ενός δικτύου σε κάποια δεδομένα να ξεκινάει από τυχαίες παραμέτρους, φορτώνονται παράμετροι ενός δικτύου με την ίδια αρχιτεκτονική που είχαν εκπαιδευτεί σε παρομοια δεδομένα ακόμα και αν δεν επίλυαν το ίδιο

πρόβλημα. Στην συνέχεια ξεκινώντας με αυτές τις παραμέτρους το δίκτυο εκπαιδεύεται στα καινούρια δεδομένα και προσαρμόζει τις παραμέτρους, που είχαν φορτωθεί στην εκμάθηση αυτών των νέων δεδομένων. Η συγκεκριμένη μέθοδος ονομάζεται fine-tuning.

Για να περιγράψουμε την παραπάνω σχέση μαθηματικά μπορούμε να ορίσουμε αρχικά ενα πεδιο (domain), το οποίο αποτελείται από έναν χώρο κατάστασης $X = \{x_1, ..., x_n\}$ και μια πιθανοτική κατανομή $P(X)$, κάθε διαφορετικό πεδίο (domain) έχει διαφορετικό χώρο κατάστασης και διαφορετική πιθανοτική κατανομή. Έχοντας ως δεδομένο ένα πεδίο $D = \{X, P(X)\}$ μπορούμε να ορίσουμε μία εργασία (task), που αποτελείται από ένα χώρο ετικετών Y και μία συνάρτηση πρόβλεψης $f(\cdot)$, και συμβολίζεται με $T = \{Y, f(\cdot)\}$. Παίρνοντας ζεύγη απο $\{x_i, y_i\}$ με $x_i \epsilon X$ και $y_i \epsilon Y$ μπορούμε, να ορίσουμε μια συνάρτηση εκτίμησης με σκοπό την εξαγωγή προβλέψεων για τις ετικέτες $f(x)$ ενός δείγματος x. Θεωρούμε ένα αρχικό τομέα για την προ-εκπαίδευση του δικτύου D_S και έναν τόμεα στον οποίο θέλουμε το δίκτυο να κάνει fine-tune D_T. Πιο συγκεκριμένα ορίζουμε το D_S ως :

$$D_S = \{(x_{s_1}, y_{s_1}), ..., (x_{s_{n_S}}, y_{s_{n_S}})\} \tag{2.2}$$

με x_{S_i} να είναι το σύνολο δεδομένων και y_{S_i} οι αντίστοιχες ετικέτες κλάσης. Με τον ίδιο τρόπο για τον δεύτερο τομέα έχουμε :

$$D_T = \{(x_{T_1}, y_{T_1}), ..., (x_{T_{n_T}}, y_{T_{n_T}})\} \tag{2.3}$$

με είσοδο x_{T_i} και έξοδο y_{T_i}.

Έχοντας επεξηγήσει τα παραπάνω μεγέθη, μπορεί να δοθεί ο ορισμός για την μεταφορά μάθησης. Με δεδομένα εναν τομέα - πηγή D_S, έναν τομέα - στόχο D_T και τις αντίστοιχες εκπαιδευόμενες εργασίες τους T_S και T_T , η μεταφορά μάθησης εστιάζει στην βελτίωση της εκμάθησης της συνάρτησης πρόβλεψης του στόχου $f(\cdot)_T$ στο D_T, εκμεταλλευόμενη την γνώση στα D_S και T_S, με $D_S \neq D_T$ και $T_S \neq T_T$.

Βέβαια υπάρχουν αρκετές τροποποιήσεις, που μπορούν να πραγματοποιηθούν στον τρόπο εκτέλεσης της μεθόδου μεταφοράς μάθησης ανάλογα με το επιθυμητό πρόβλημα επίλυσης. Πολλά πειράματα έχουν πραγματοποιηθεί, που συγκρίνουν κλασικές μεθόδους μηχανικής μάθησης με αυτές της μεταφοράς μάθησης και έχουν εξαχθεί συμπεράσματα οτι οι μέθοδοι της μετάφορες μάθησης πέτυχαν καλύτερα αποτελέσματα, έναντι των κλασσικών μεθόδων σε πληθώρα προβλημάτων. Κάποιες εφαρμογές στις οποίες κατάφερε η μεταφορά μάθησης να πετύχει αξιόλογα αποτελέσματα ήταν εκείνες της κατηγοριοποίησης εικόνων, συναισθημάτων αλλά και σε εφαρμογές επεξεργασίας φυσικής γλώσσας. Ωστόσο η μεταφορά μάθησης έχει και κάποιους περιορισμούς, ο κύριος απο αυτους ειναι οτι ο τομέας των δεδομένων πάνω στον οποίο εκπαίδευεται το δίκτυο χρειάζεται να έχει παρόμοιο χώρο χαρακτηριστικών με τον τομέα δεδομένων πάνω στον οποίο πραγματοποιείται το fine tuning.

2.5.3 Meta Learning[1]

Εκτός απο την τεχνική της μεταφοράς μάθησης (transfer learning)[37] έχουν αναπτυχθεί και άλλες μέθοδοι τα τελευταια χρονια, που βασίζονται στον τρόπο εκπαίδευσης του ανθρώπινου οργανισμού και έχουν ως σκοπό να εκπαιδεύσουν νευρωνικά δίκτυα, τα οποία

θα είναι ικανά να προσαρμοστούν γρήγορα σε δεδομένα, τα οποία έχουν κάποιες διαφορές από τον τομέα(domain που εκπαιδεύτηκαν αρχικά. Έτσι θα μπορούν να δημιουργηθούν εύρωστα (robust) δίκτυα, τα οποία με εκπαίδευση σε λίγα δεδομένα του καινούργιο συνόλου θα επιστρέφουν αξιόλογες προβλέψεις[38]. Η τεχνική αυτή που θέλουμε να επιτύχουμε ονομάζεται Domain Generalization και υπάρχουν πολλές μέθοδοι που έχουν αναπτυχθεί για να το επιτύχουν αυτό.

Μια απο αυτές τις τεχνικές στην οποία αξίζει να αναφερθούμε είναι αυτή του meta learning, η οποία βασίζεται στην ιδέα της ανάπτυξης ενός γενικού μοντέλου, το οποίο εκπαιδεύεται σε πολλαπλα tasks είτε μέσω τεχνικών βελτιστοποίησης, τεχνικών που βασίζονται σε μετρικές ή σε μοντέλα (optimization-based,metric-based,model-based techniques. Ο τρόπος με τον οποίο οργανώνεται η εκπαίδευση ενός μοντέλου με την τεχνική του meta learning, είναι ο διαχωρισμός του multi-source domain σε δεδομένα μετα-εκπαίδευσης και μετα-ελέγχου (meta-train,meta-test) με σκοπό να προσομοιώσει την μεταβολή του πεδίουdomain. Αν θεωρήσουμε ως θ τις επιθυμητές παραμέτρους τότε ορίζεται η σχέση:

$$\begin{aligned} \theta^* &= Learn(S_{mte} : \phi^*) \\ &= Learn(S_{mte} : MetaLearn(S_{mtr_n})), \end{aligned} \tag{2.4}$$

όπου $\phi^* = MetaLearn(S_{mtr_n})$ και αποτελούν τις παραμέτρους που εξήχθησαν κατα την διαδικασία εκπαίδευσης στα meta-train δεδομένα. Στην συνέχεια αυτές οι παράμετροι φορτώνονται στο δίκτυο και αυτό εκπαιδεύεται για να μάθει τις θ^* παραμετρους που προκύπτουν απο τα meta-test δεδομένα. Οι συναρτήσεις εκπαίδευσης Learn, MetaLearn μπορούν να υλοποιηθούν με πολλές διαφορετικές μεθόδους για την εκμάθηση των παραμέτρων. Μερικές απο τις πιο γνωστές εφαρμογές είναι το reptile και το MAML, οι οποίες θα αναλυθούν με μεγαλύτερη λεπτομέρεια στο πρακτικό μέρος της παρούσας εργασίας. Η κύρια διαφορά αυτών των τεχνικών αφορα τον τρόπο της ανανέωσης των παραμέτρων με την τεχνική του gradient descent. Το reptile βασίζεται στα gradient πρώτης τάξης, ενώ το MAML χρησιμοποιεί και gradient δεύτερης τάξης.

$$\theta = \theta - a \cdot \frac{\partial(l(S_{mte}; \theta) + \beta \cdot l(S_{mte}; \phi))}{\partial\theta} \tag{2.5}$$

με τα a και β να είναι ρυθμοι μάθησης(learning rates) για τις εξωτερικές και εσωτερικές επαναλήψεις αντίστοιχα.

Μέρος III

Πρακτικό Μέρος

Κεφάλαιο 3

Περιγραφή θέματος

Στο κεφάλαιο αυτό αρχικά γίνεται μια περιγραφή του προβλήματος, που κληθήκαμε να αντιμετωπίσουμε. δηλαδή αυτό της ταξινόμησης βιοϊατρικών εικόνων με υψηλή αξιοπιστία. Πιο συγκεκριμένα θα πραγματοποιηθεί περιγραφή:

- των τριων συνόλων δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου
- των τεχνικών επαύξησης δεδομένων (techniques) που χρησιμοποιήθηκαν
- του συνελικτικού νευρωνικου δικτύου στο οποίο εισήχθησαν τα δεδομένα για την ταξινόμηση των εικόνων
- του τρόπου εκπαίδευσης που χρησιμοποιήθηκε για την επίλυση του προβλήματος κατηγοριοποίησης

3.1 Σχετικές εργασίες

Οι τεχνικές του few shot learning έχουν γνωρίσει μεγάλη επιτυχια σε ενα ευρύ σύνολο προβλημάτων και έχουν αναπτυχθεί αρκετές τεχνικές, που βασίζονται πάνω σε αυτήν την τεχνική εκπαίδευσης. Αυτές τις τεχνικές μπορούμε να τις χωρίσουμε σε δύο κύριες κατηγορίες, η μια απο αυτές αναφέρεται σε τεχνικές που βασίζονται σε μετρικές [39, 40, 41, 42, 43, 44, 45] και η δεύτερη σε τεχνικές που βασίζονται σε optimazation[46, 47, 48, 49, 50]

Τα σύνολα δεδομένων που χρησιμοποιήθηκαν είναι δημοσίως διαθέσιμα και βρισκονται στις εξής πηγές [51, 52, 53]. Η μέθοδος εκπαίδευσης του δικτύου έγινε με τον ίδιο τρόπο που περιγράφεται στο [2].

Επιπλέον χρησιμοποιήθηκε ο τρόπος εκπαίδευσης, που περιγράφεται στο [2] για τα τρια βιοϊατρικά σύνολα δεδομένων, που προαναφέρθηκαν και τα αποτελέσματα που προέκυψαν συγκρίθηκαν με τα αποτελέσματα που δίνονται στο [3].

Κεφάλαιο 4

Ανάλυση και σχεδίαση

Στο κεφάλαιο αυτό παρουσιάζεται η μελέτη, που έγινε για την επιλύση του προβλήματος κατηγοριοποίησης εικόνων με αυξημένη αξιοπιστία. Αρχικά περιγράφουμε τα τρια σύνολα δεδομένων, που χρησιμοποιήσαμε για την εκπαίδευση του δικτύου. Στην συνέχεια περιγράφτηκαν οι τεχνικές επαύξησης δεδομένων, το νευρωνικό δίκτυο που χρησιμοποιήθηκε και η διαδικάσια εκπαίδευσης

4.1 Ανάλυση και περιγραφή δεδομένων

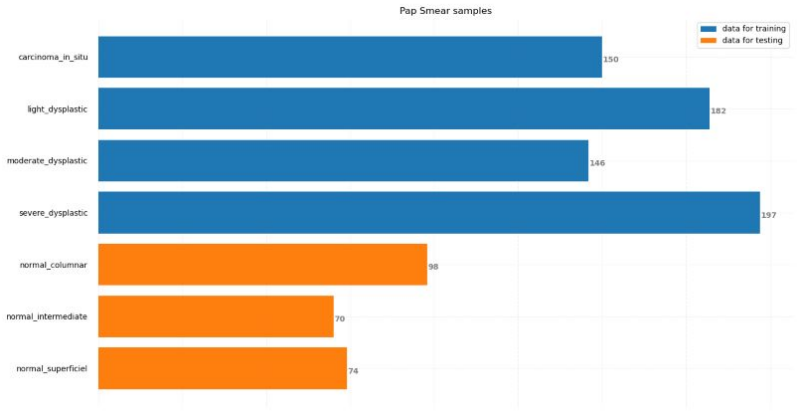
Στην ενότητα αυτή θα παρουσιαστούν αναλυτικά τα τρια σύνολα δεδομένων που χρησιμοποιήθηκαν

4.1.1 Pap Smear

Το Pap Smear dataset[51] αποτελείται απο μικρές εικόνες, που απεικονίζουν δείγματα απο ίστους, που εξήγθησαν απο τον τράχηλο. Αυτά τα δείγματα δημοσιεύθηκαν απο το Hervel University Hospital και χρησιμοποιήθηκε σε αυτά η τεχνική Papanicholau, που αποτελεί μια πολυχρωματική κυτολογική τεχνική χρωματισμού, που χρησιμοποιείται για διάγνωση αλλάγων στα κύτταρα προτού προχώρησουν σε επεμβατικούς καρκίνους του τραχήλου. Το σύνολο αυτών των δεδομένων αποτελείται απο 917 εικόνες οι οποίες ειναι ανισοκατανεμημένες σε 7 διαφορετικές κλάσεις απο ειδικούς πάνω στον συγκεκριμένο τομέα. Οι 7 αυτές κλάσεις είναι οι εξής:

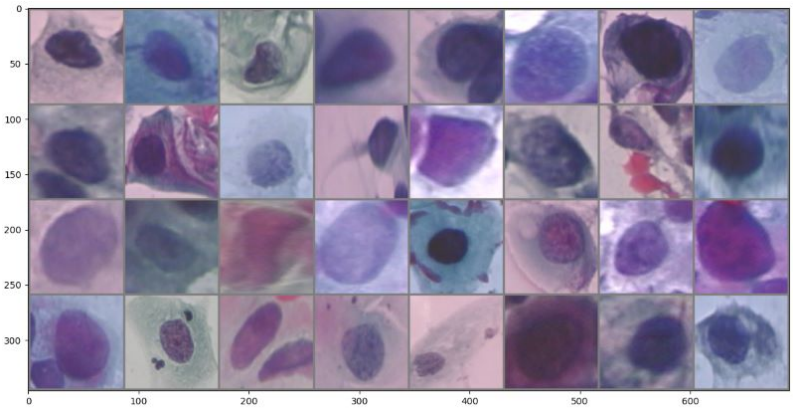
- Carcinoma in situ αφορά καρκινικά κύτταρα τα οποία βρίσκονται εντός του ιστού και δεν έχουν εξαπλωθεί εκτός του ιστού στα γύρω αγγεία
- Light Dysplastic αφορά καρκινικά κύτταρα τα οποία παρουσιάζουν ελαφριά δυσπλάσια
- Moderate Dysplastic αφορά καρκινικά κύτταρα τα οποία παρουσιαζουν μέτρια δυσπλάσια
- Severe Dysplastic αφορά καρκινικά κύτταρα τα οποία παρουσιάζουν σοβαρή δυσπλάσια
- Normal Columnar αφορά κύτταρα τα οποία παρουσιαζουν φυσιολογική στοίβαξη

- Normal Intermediate αφορά φυσιολογικά κύτταρα στους μεσαίους ιστούς.
- Normal Superficial αφορά φυσιολογικά κύτταρα στους επιφανειακούς ιστούς.



Σχήμα 4.1: *Pap-Smear samples*

Απο αυτές τις 7 κλάσεις τέσσερις επιλέχθηκαν ως κλάσεις για το meta-train στάδιο και πιο συγκεκριμένα οι κλάσεις (carcinoma in situ,light dysplastic,moderate dysplastic,severe dysplastic). Ένώ οι υπόλοιπες τρεις κλάσεις (normal columnar,normal intermediate, normal superficial) χρησιμοπποιήθηκαν για το meta-test στάδιο. Στο 4.1 φαίνεται ο αριθμός των δειγμάτων ανα κλάση και στο 4.2 παρουσιάζονται μερικά παραδείγματα ανα κλάση.



Σχήμα 4.2: *Pap-Smear with no augmentation*

4.1.2 BreakHis dataset

Το σύνολο δεδομένων των Ιστοπαθολογικών εικόνων καρκίνου του στήθους[52] περιέχει 9109 εικόνες απο μικροσκόπιο που απεικονίζει οιδήματα στο στήθος απο 82 περιστατικά σε διαφορετικές κλίμακες (x40, x100, x200, x400). Οι εικόνες έχουν διαστάσεις 700x460. Το σύνολο των δεδομένων κατηγοροιοποιείται στις 8 παρακάτω κλάσεις:

- Adenosis αποτελεί έναν καλοήθη όγκο, που οφείλεται στην υπερόγκωση του αδένα που παράγει γάλα

- Tubular Adenoma αποτελεί ενα είδος καρκίνου, που ξεκινάει απο τους γαλακτοφόρους αδένες και εξαπλώνεται στο υπόλοιπο του μαστού

- Phyliodes Tumor αφορά ένα είδος καρκίνου, που αναπτύσσεται στον συνδετικό ιστό

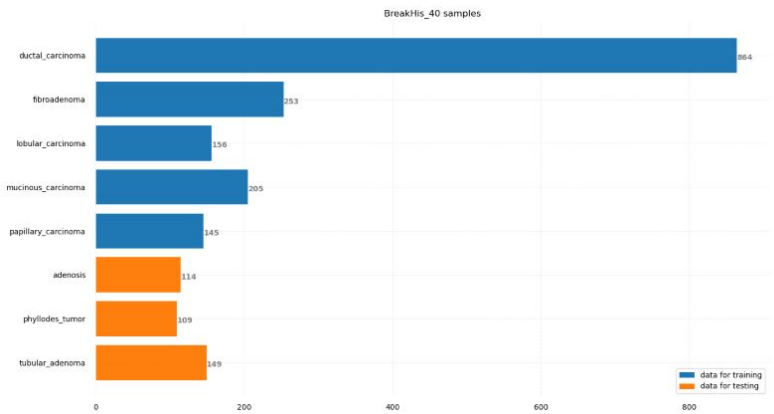
- Papillary Carcinoma είναι το πιο σύνηθες είδος καρκίνου, που αναπτύσσεται στον θυροειδή αδένα.

- Lobular Carcinoma είναι ένα είδος καρκίνου, που ξεκινάει απο τους γαλακτόφορους αδένες και εξαπλώνεται στους γύρω ιστούς του στήθους

- Mucinous Carcinoma Αφορά ένα είδος καρκινού, που εμφανίζεται στα κύτταρα που παράγουν βλενογόνες προτείνεις και στην συνέχεια αυτές οι προτεινες γίνονται μέρος του όγκου.

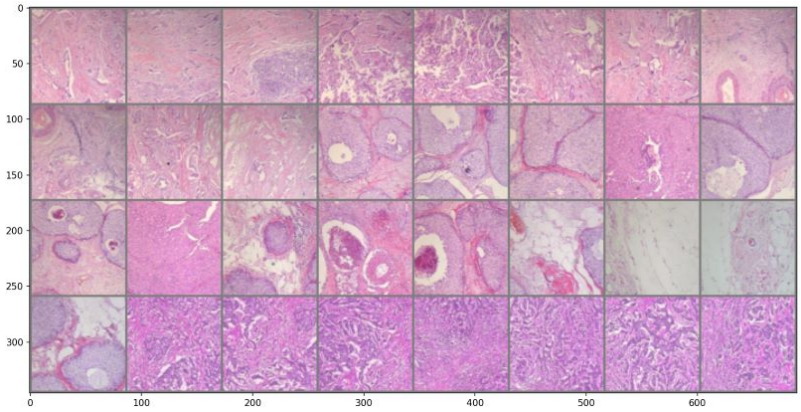
- Fibroadenoma αποτελεί έναν καλοήθη όγκο, που εμφανίζεται ως οίδημα στο στήθος και αποτελείται απο συνδυασμό αδενικού ιστού και συνδετικού ιστού.

- Ductal Carcinoma αποτελεί την ύπαρξη μη φυσιολογικών κυττάρων στους γαλακτοφόρους αγωγούς.



Σχήμα 4.3: *BreakHis_40 samples*

Απο τις προαναφερθήσες 8 κλάσεις 5 (papillary carcinoma,lobular carcinoma,mucinous carcinoma,fibroadenoma,ductal carcinoma) επιλέχθησαν για το τομέα του meta-train ενώ οι υπόλοιπές 3 (Adenosis, Tubular Adenoma, Phyliodes Tumor) χρησιμοποιήθηκαν στον τομέα του meta-testing. Στο 4.3 φαίνεται ο αριθμός των δειγμάτων ανα κλάση και στο 4.4 παρουσιάζονται μερικά παραδείγματα ανα κλάση.



Σχήμα 4.4: *BreakHis_40 with no augmentation*

4.1.3 Isic dataset

Το Isic dataset[53] αποτελείται απο 10,015 είκονες, που απεικονίζουν το διαφορετικές δερματικές παθήσεις κατηγοριοποιημένες σε 7 κλάσεις με περισσότερες αναπαραστάσεις να αφορούν καλοήθη ογκούς και λιγότερες κακοήθεις. Οι εικόνες έχουν διαστάσεις 700x460.

- Dermatofibroma αποτελεί έναν κοινό καλοήθη ινώδη όγκο, που βρίσκεται συνήθως στο δέρμα των κάτω ποδιών

- Vascular Lesion αποτελούν αγγειακές βλάβες και είναι σχετικά συχνές ανωμαλίες του δέρματος και των αντίστοιχων ιστών και εμφανίζονται ως εκ γενετη σημάδια

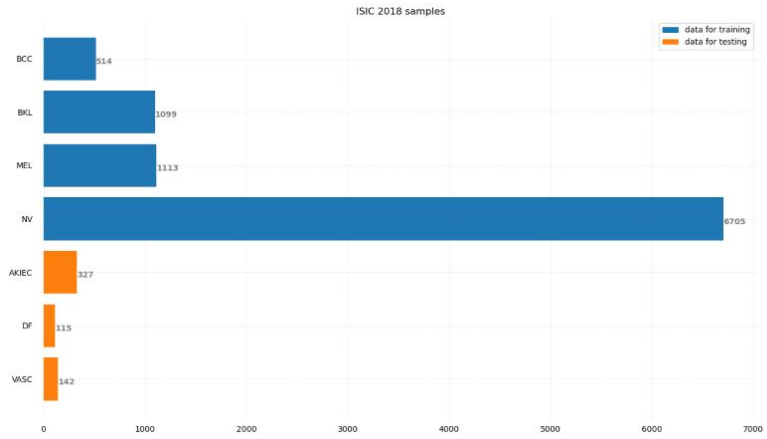
- Actinic Keratosis αποτελούν ξηρές κηλίδες δέρματος, που έχουν καταστραφεί από τον ήλιο. Συνήθως δεν αποτελούν πρόβλημα αλλά μπορούν εξελιχθούν σε καρκίνο του δέρματος.

- Basal Cell Carcinoma αποτελεί τον πιο συχνά εμφανιζόμενο καρκινό του δερμάτος και οφείλεται κύριως στην εκθεση σε υπεριώδη ακτινοβολία.

- Benign Keratosis είναι ένα συχνά εμφανιζόμενο είδος καλοήθης όγκου το οποίο εμφανίζεται στην επιδερμιδά και οφείλεται στην κερτίνη που υπάρχει στα επιδερμικά κύτταρα.

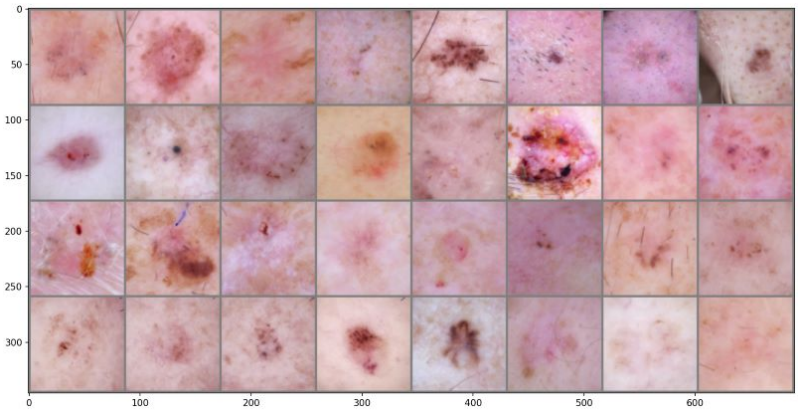
- Melanoma είναι ένα είδος κακοήθη καρκίνου που εμφανίζεται στα κυττάρα τα οποία είναι υπεύθυνα για το χρώμα του δερματος μας τα οποία λέγονται μελανοκύτταρα.

- Melanocytic Nevus είναι συνήθως μια μη καρκινική κατάσταση των κυττάρων του δέρματος που είναι υπεύθυνα για το χρώμα του δέρματος. Αποτελεί ενα είδος μελανοκυτταρικού όγκου που περιέχει κύτταρα σπίλων.



Σχήμα 4.5: *Isic_2018 samples*

Απο τις προαναφερθήσες 7 κλάσεις 4 (Basal Cell Carcinoma, Benign Keratosis, Melanoma, Melanocytic Nevus) επιλέχθησαν για το τομέα του meta-train, ενώ οι υπόλοιπές 3 (Dermatofibroma, Vascular Lesion, Actinic Keratosis) χρησιμοποιήθηκαν στον τομέα του meta-testing. Στο 4.5 φαίνεται ο αριθμός των δειγμάτων ανα κλάση και στο 4.6 παρουσιάζονται μερικά παραδείγματα ανα κλάση.



Σχήμα 4.6: *Isic_2018 with no augmentation*

4.2 Προεπεξεργασία δεδομένων

4.2.1 Αρχική προεπεξεργασία και απλές τεχνικές επαύξησης δεδομένων

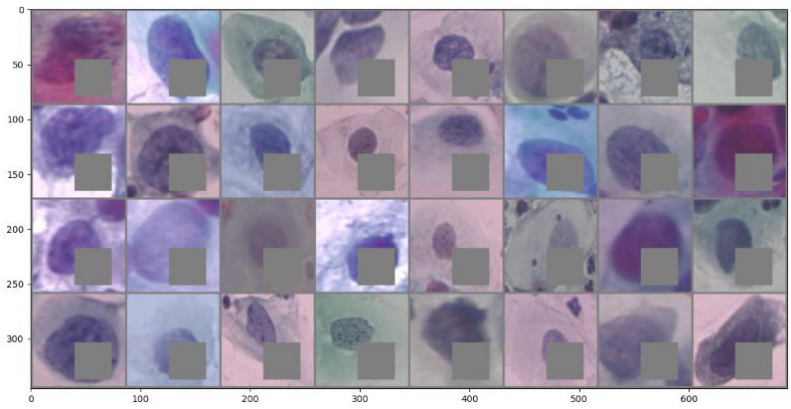
Οι διαστάσεις ολών των εικόνων τροποποιήθηκαν απο τις αρχικές τους διαστάσεις σε μεγέθη 84x84. Στην συνέχεια εφαρμόστηκαν τυχαίες τεχνικές επαύξησης των δεδομένων όπως τυχαία περιστροφή, αναστροφή και τυχαία μετατόπιση. Τέλος εφαρμόστηκε μια κανονικοποίηση των εικόνων ωστε οι τιμές απο 0-256 να τροποποιηθούν σε τιμές εως 1.

4.2.2 Σύνθετες Τεχνικές Επαύξησης Δεδομένων

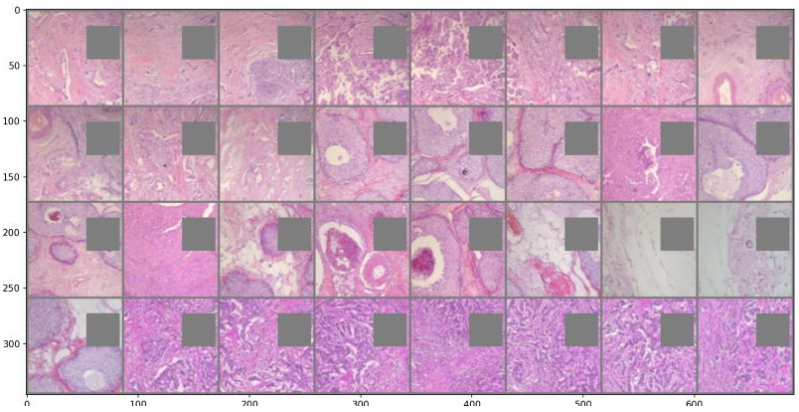
CutOut

Η τεχνική του cutout[54] παράγει τυχαία τετράγωνα τα οποία αλλάζουν τις τιμές των pixel που καλύπτουν σε 0. Παρακάτω φαίνονται μερικά παραδείγματα απο την τεχνική CutOut για κάποια batches εικόνων. Το cutout αποτελεί μια πολύ καλύτερη τενική επάυξησης δεδομένων απο το dropout. Το dropout αφαιρεί τυχαία χαρακτηριστικά απο τους πίνακες χαρακτηριστικών που προκύπτουν, τα οποία υπάρχει πιθανότητα να μην αφαιρεθούν απο άλλους πινάκες χαρακτηριστικών. Αντίθετα επειδη το cutout αφαιρεί χαρακτηριστικά απο το αρχικό σταδιο δεν υπάρχει πιθανότητα να περιέχει χαρακτηριστικά, που ανήκουν στις κρυμμένες περιοχες.

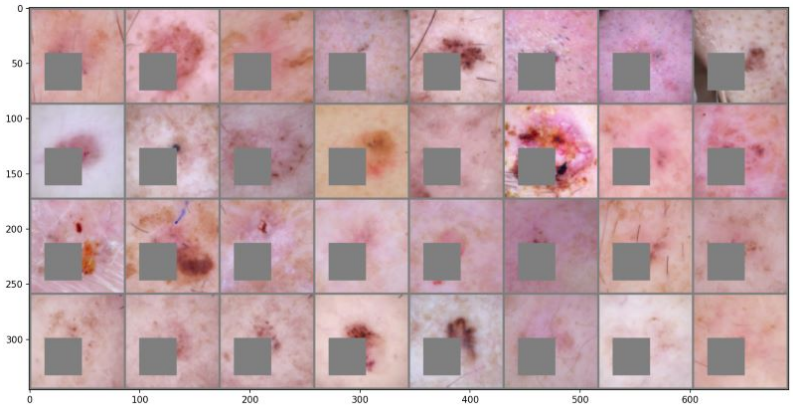
Στα 4.7, 4.8, 4.9 το augmentation cutout που έχει εφαρμοστεί τυχαια σε εικόνες για τα 3 datasets που αναλύσαμε προηγουμένως



Σχήμα 4.7: *Pap-Smear with cutout*



Σχήμα 4.8: *BreakHis_40 with cutout*



Σχήμα 4.9: *ISIC_2018 with cutout*

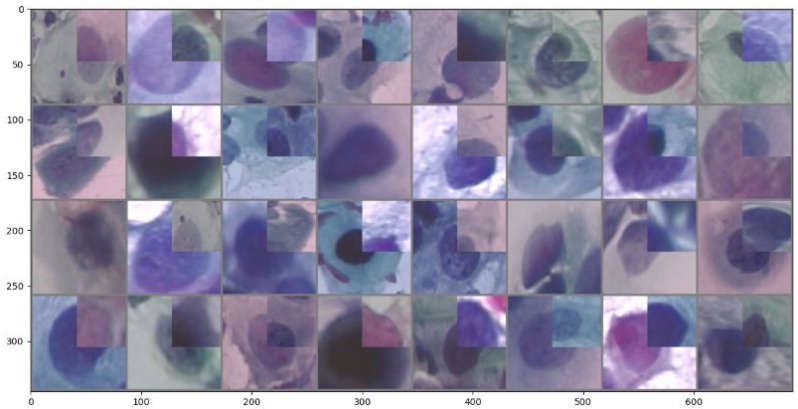
CutMix

Η τεχνική του cutmix[55] παράγει ενα καινούριο δείγμα κόβοντας ένα κομμάτι απο την εικόνα και προσθέτοντας το σε μια διαφορετική εικόνα απο το σύνολο των δεδομένων για εκπαίδευση. Επίσης τα δεδομένα labels προσαρμόζονται κατάλληλα στις περιοχές που έχουν κοπεί. Αν θεωρήσουμε δυο δείγματα (x_i, y_i) και $(x_{m(i)}, y_{m(i)})$. Τότε το νέο εικονικό δείγμα $(x^{'}, y^{'})$ γράφεται ως:

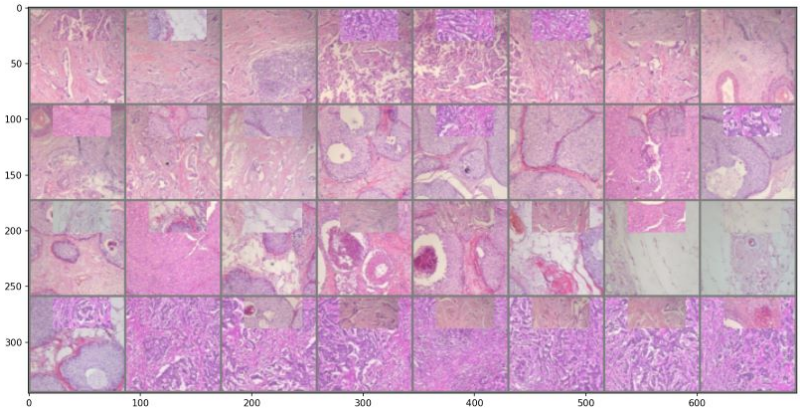
$$x^{'} = F \odot x_i + (1 - F) \odot x_{m(i)} \tag{4.1}$$

$$y^{'} = \lambda \cdot y_i + (1 - \lambda) \cdot y_{m(i)} \tag{4.2}$$

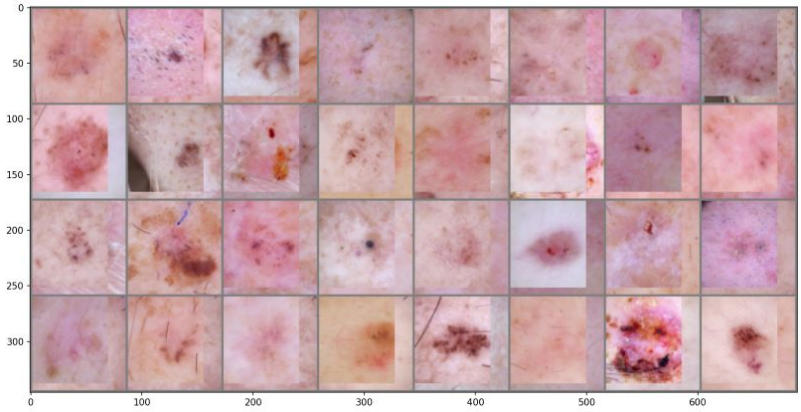
Στα 4.10,4.11,4.12 φαίνεται το augmentation cutmix, που έχει εφαρμοστεί τυχαια σε εικόνες για τα 3 datasets, που αναλύσαμε προηγουμένως



Σχήμα 4.10: *Pap-Smear with cutmix*



Σχήμα 4.11: *BreakHis_40 with cutmix*



Σχήμα 4.12: *ISIC_2018 with cutmix*

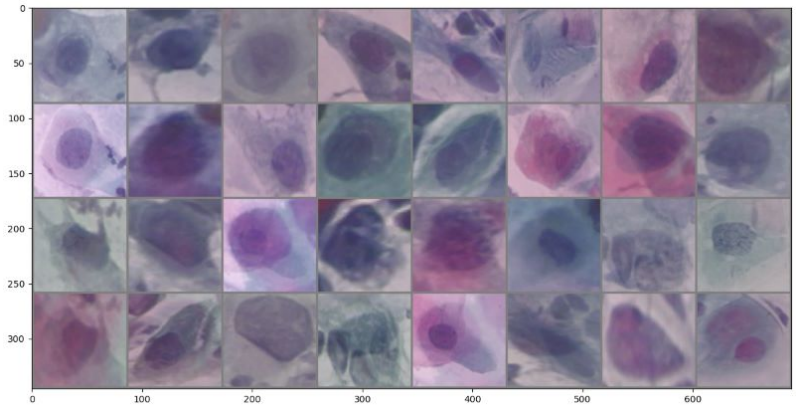
MixUp

Το mixup[56] είναι μια τεχνική επαύξησης δεδομένων, η οποία ενισχύει την ικανότητα του νευρωνικού δικτύου να γενικεύει δημιουργόντας εικονικά δείγματα απο την κατανομή των δεδομένων. Τα εικονικά δείγματα δημιουργούνται μέσω γραμμικής παρεμβολής ενός δείγματος με ενός άλλου τυχαια επιλεγμενου απο το ίδιο batch

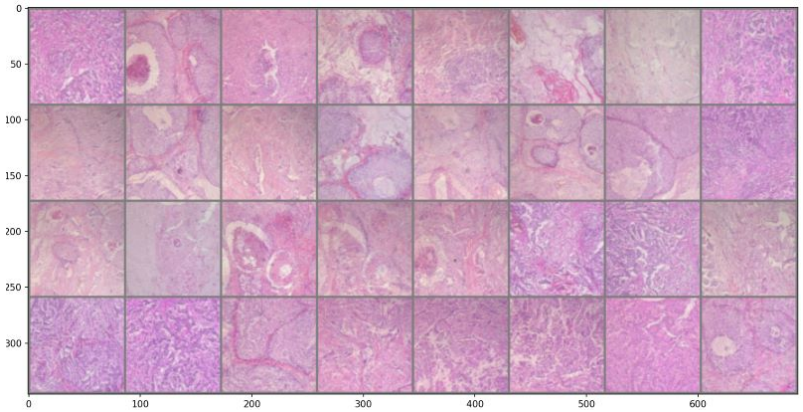
$$x_i^m = \lambda \cdot x_i + (1 - \lambda) \cdot x_{m(i)} \tag{4.3}$$

$$y_i^m = \lambda \cdot x_i + (1 - \lambda) \cdot y_{m(i)} \tag{4.4}$$

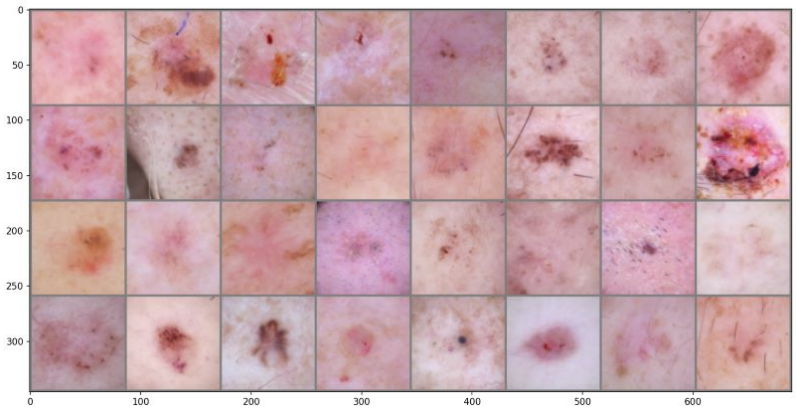
όπου το $\lambda \sim Beta(\zeta, \zeta)$. Στα 4.13,4.14,4.15 φαίνεται ενα batch εικόνων, για κάθε σύνολο δεδομένων που τους έχει εφαρμοστεί η τεχνική του mixup



Σχήμα 4.13: *Pap-Smear with mixup*



Σχήμα 4.14: *BreakHis_40 with mixup*



Σχήμα 4.15: *ISIC_2018 with mixup*

PixMix[57]

Οι τεχνικές της επαύξησης των δεδομένων χρησημοποιούνται συχνά για να παραγούν διαφορετικά και πολυπαραγοντικά δεδομένα. Έτσι οι περισσότερες τεχνικές επαύξησης δεδομένων δημιουργούν εικόνες που έχουν μεγαλύτερη εντροπία μεταξύ τους, δηλαδή εικόνες με μεγαλύτερες διαφορές. Με την τυχαία περιστροφή των εικόνων, την τυχαια περικοπή, την

τυχαια αλλάγη χρωματων αυξάνεται η αβεβαιότητα και η τυχαιότητα των χρωμάτων. Έτσι τα δεδομένα αποκτούν μεγαλύτερη περιπλοκότητα και παραλλαγές στα πρότυπα που εμφανίζουν.Παρολα αυτά πολλές τεχνικές επαύξησης δεδομένων αυξάνουν μερικές μετρικές ασφάλειας[58], αλλά μειώνουν κάποιες άλλες. Λόγω αυτού του προβλήματος έχουν πραγματοποιηθεί έρευνες για την ανάπτυξή τεχνικών επαύξησης, που θα βελτιώνουν όλες τις μετρικές ασφάλειας, μια απο αυτές τις μελέτες που έχουν γίνει αφορά την χρήση σύνθετων εικόνων. Οι σύνθετες εικόνες αποτελούν αναπραστάσεις, που εμφανίζουν μια περιπλοκότητα στον τρόπο με τον όποιο είναι δομημένες, ενα καλό παράδειγμα για αυτές τις εικόνες είναι τα fractals[59] που χρησιμοποιούνται για την εκπαίδευση ταξινομητών[59, 60].

Το PixMix[57] χρησιμοποιεί αυτές τις σύνθετες εικόνες, για να παράξει αναπαραστάσεις που βελτιώνουν όλες τις μετρικές ασφάλειας. Οι σύνθετες εικόνες που χρησιμοποιούνται, είναι fractals και χαρακτηριστικά απεικόνισης (feature visualization). Στόχος του PixMix είναι να βοηθήσει μοντέλα, τα οποία θα έχουν τα εξής χαρακτηρισιτκά:

- Ευρωστότητα (Robustness): αναφέρεται στην ικανότητα δημιουργίας μοντέλων μηχανικής μάθησης, τα οποία είναι ανθεκτικά σε μετατοπίσεις της κατανομής δεδομένων. Ερεύνες έχουν πραγματοποιηθεί[61], που δείχνουν οτι όταν ενα CNN μοντέλο εκπαιδεύεται σε εικόνες με διαφορετικό τρόπο αναπαράστασης(stylized images) παράγει καλύτερα αποτελέσματα ως προς την ευρωστία όταν πραγματοποιεί προβλέψεις. Για τον έλεγχο της ευρώστιας αυτών των μοντέλων χρησιμοποιούνται ειδικά σύνολά δεδομένων ως benchmarks[62]. Ενα παράδειγμα απο αυτά είναι το ImageNet-c[63], το οποίο περιέχει δεδομένα με corruptions και χρησιμοπούνται για τον έλεγχο μοντέλων που έχουν εκπαιδευτεί στο ImageNet.

- Προσαρμοστικότητα (Calibration) αφορά την ικανότητα του δικτύου να βελτιστοποιεί την βεβαιότητα των προβλέψεων[64, 65] που πραγματοποιεί και είναι πολύ χρήσιμο σε tasks που σχετίζονται με την ταξινόμηση δεδομένων. Οι τρόποι με τους οποίους τα μοντέλα πετυχαίνουν καλύτερο calibration, είναι με την χρήση validation set και εφαρμόζοντας pretrain[66].

- Ανίχνευση Ανωμαλιών (Anomaly Detection) αφορά την ικανότητα του μοντέλου να εκτιμάει εαν ένα δεδομένο, που εισάγεται, είναι μη ομάλο, δηλαδή δεν ανήκει στην κατανομή δεδομένων πάνω στην οποία εκπαιδεύεται το μοντέλο. Πολλές έρευνες έχουν πραγματοποιηθεί για τον εντοπισμο δεδομένων OOD (που δεν ανήκουν στην κατανομή του συνόλου δεδομένων εκπαίδευσης)[67] σε αυτές έχουν χρησιμοποιηθεί μοντέλα GAN[68] και άλλα μοντέλα ταξινόμησης για την ανίχνευση διαφορετικότητας και φυσικών ανομαλιών.

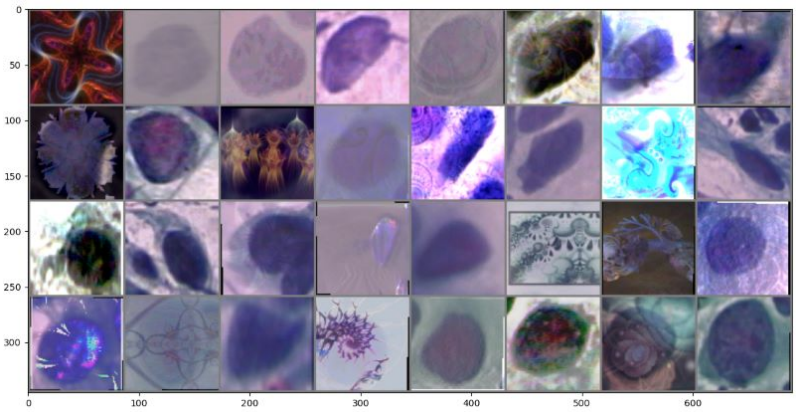
Οι τεχνικές Data Augmentations βοηθάει στην δημιουργία μοντέλων μηχανικής μάθησης, τα οποία είναι πιο εύρωστα και αυξάνουν το accuraccy ενός μοντέλου όσο θα το βελτίωνε μια αύξηση του μεγέθους του μοντέλου κατα 10 φορές. Πολλές τεχνικές επαύξησης δεδομένων, όπως το Cutmix, Cutout, Mixup, AugMix[69] έχουν οδηγηθεί στο συμπέραμσα οτι η προσθήκη περιοχών με τυχαίο θόρυβο στις εικόνες εκπαίδευσης βελτιώνει την ευρωστία (robustness) του δικτύου.

Για την αξιολόγηση των τεχνικών επαύξησης χρησιμοποιούνται μετρικές ασφάλειας. Αυτές οι μετρικές έχουν δείξει οτι οι τεχνικές Cutmix, Mixup, Cutout, Augmix, ShakeDrop[70] επιδρούν διαφορετικά σε αυτές τις μετρικές. Έρευνες έχουν δείξει οτι οι τεχνικές επαύξησεις δεδομένων συνήθως βελτιώνουν μερικές μετρικές, ενώ έχουν σχεδόν μηδενική επίδραση σε άλλες. Η τεχνική επαύξησης PixMix αξιολογήθηκε στις παραπάνω μετρικές και έδειξε οτι πετύχε καλύτερα αποτελέσματα απο τις υπολοίπες τεχνικές επαύξησης σε όλες τις μετρικές.

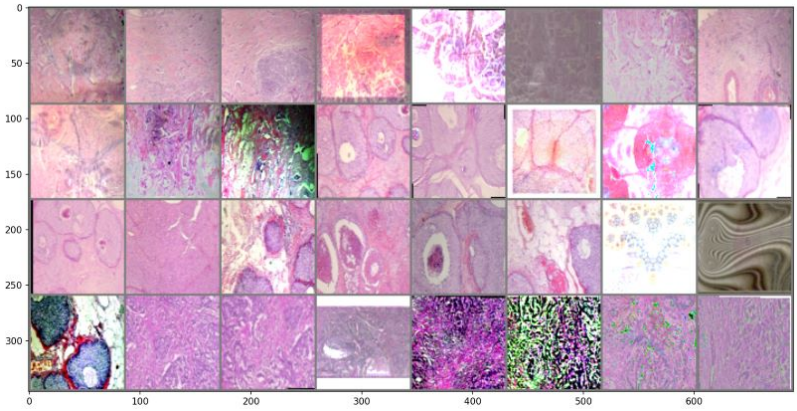
Αναλύοντας την δομή του Pixmix παρατηρούμε οτι αποτελείται απο δύο κύρια μέρη. Το πρώτο μέρος είναι ένα σύνολο απο εικόνες οι οποίες έχουν σύνθετη δομή και το δεύτερο μέρος αποτελείται απο ενα σύστημα, το οποίο εφαρμόζει διάφορες τροποποιήσεις στα δεδομένα εκπαίδευσης με σκοπό την επαύξησης τους. Έτσι το PixMix είναι ένα σύστημα που ενσωματώνει διαφορετικά πρότυπα απο εικόνες σύνθετων δομών (όπως fractals) και χαρακτηριστικών αναπαράστασης στις υπάρχουσες εικόνες εκπαίδευσης. Σε αντίθεση με άλλες τεχνικές επαύξησης το PixMix δεν πραγματοποιεί κάποια τροποποίηση στις κλάσεις των εικόνων εκπαίδευσης, αφού τα fractals που εφαρμόζει σε αυτές δεν έχουν κλάσεις. Επιλέχθηκαν τα fractals επειδή έχουν κάποιες σημαντικές ιδιότητες η κύρια απο αυτές είναι οτι εχούν ιδιότητες μη τυχαιότητας και είναι αδύνατο να προκύψουν απο μέγιστη εντροπία και διαδικασίες τυχαίου θορύβου.

Ο τρόπος με τον οποίο πραγματοποιείται η τεχνική επαύξησης PixMix είναι οτι αρχικά το σύστημα(Pipeline) παίρνει μια εικόνα εκπαίδευσης. Σε αυτή την εικόνα εκπαίδευσης εφαρμοζεται μια τυχαία διαδικασία επαύξησης με πιθανότητα 50%. Στην συνέχεια η εικόνα επαυξάνεται n τυχαίες φορές με μέγιστο αριθμό ενα k το οποίο δίνεται ως υπερπαράμετρος. Οι τεχνικές augmentation των εικόνων πραγματοποιούνται είτε αθροιστικά είτε πολλαπλασιαστικά, όπου η εικόνα εκπαίδευσης αναμειγνύεται με μια καινούρια επαύξημενη εικόνα ή μια εικόνα απο το σύνολο των σύνθετων εικόνων.

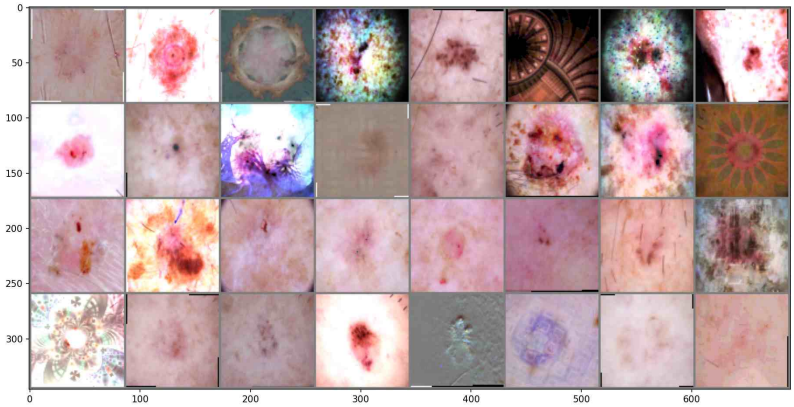
Στα 4.16,4.17,4.18 φαίνονται augmentations pixmix, που έχουν εφαρμοστεί τυχαια σε εικόνες για τα 3 datasets, που αναλύσαμε προηγουμένως



Σχήμα 4.16: *Pap-Smear with Pixmix*



Σχήμα 4.17: *BreakHis_40 with Pixmix*



Σχήμα 4.18: *ISIC_2018 with Pixmix*

4.3 Ανάλυση διαδικασίας εκπαίδευσης

Ακολουθήθηκαν δυο τρόποι εκπαίδευσης του μοντέλου στα datasets που αναλύθηκαν παραπάνω. Ο πρώτος τρόπος ήταν μέσω της διαδικασίας του transfer learning και στην συνέχεια ακολουθήθηκε η διαδικασία του meta learning, όπου χρησιμοποιήθηκε συγκεκριμένα ο αλγόριθμος (Reptile). Τέτοιες τεχνικές εφαρμόστηκαν λόγω του χαμηλού όγκου δεδομένων στα βιοϊατρικά datasets αλλά και των διαφορών στον τρόπο που παρουσιάζονται, τέτοιες διαφορες αφορούν κυρίως τον τρόπο που απεικονίζονται τα δεδομένα(το style των εικόνων). Αυτή η έλλειψη σε μεγάλο όγκο labeled δεδομένων κάνει πολύ δύσκολη την εκπαίδευση μοντέλων που χρησιμοποιούν CNN δικτύα λόγω της ανάγκης αυτών να εκπαιδευτούν σε μεγάλο όγκο δεδομένων για να επιστρέψουν αξιόπιστα αποτελέσματα. Για την αντιμετώπιση αυτου του προβλήματος αναπτύχθηκαν διαφορες τεχνικές[71, 72, 73] μια απο αυτές ήταν η χρηση GAN για να παράγει αναταγωνιστικές εικόνες που θα μοιάζουν με αυτες του αρχικού συνόλου δεδομένων. Επίσης στα περισσότερα μοντέλα που αναπτύχθηκαν για την ταξινόμηση βιοιατρικών εικόνων χρησημοποιήθηκαν τεχνικές transfer learning.

Για την αντιμετώπιση του προβλήματος της διαφορετικότητας των δεδομένων (διαφορετικά styles) χρειάζεται, να δημιουργηθούν μέσω της εκπαίδευσης μοντέλα, που θα γενικεύουν καλά, δηλαδή η εξαγωγή ακριβή αποτελεσμάτων σε δεδομένα, που δεν ανήκουν ακριβώς στην

ίδια κατανομή με τα δεδομένα πάνω στα οποία εκπαιδεύτηκε το μοντέλο. Επίσης επειδή το μοντέλο ταξινομεί βιοιατρικά δεδομένα είναι σημαντικό να παράγει αξιόπιστα αποτελέσματα.

Η τεχνική του Meta learning βασίζεται στην διαδικασία που ακολουθεί μια εκπαίδευση τύπου δασκάλου-μαθητή (Teacher-Student). Όπου αρχικά το δίκτυο εκπαίδευεται σε tasks πάνω σε μια κατανομή δεδομένων, με σκοπό την εκμάθηση των καλύτερων δυνατών παραμέτρων, που να μπορούν να ανανεωθούν και να προσαρμοστούν κατάλληλα σε μια καινούρια κατανομή δεδομένων. Πιο συγκεκριμένα το δίκτυο teacher έχει σκοπό να εκπαιδευτεί καταλλήλα σε μια κατονομή δεδομένων με στόχο να παραξει παραμέτρους πάνω στις οποίες το δίκτυο student θα μπορεί, να μάθει πιο εύκολα και με μεγαλύτερη αξιοπιστία μια καινούρια κατανομή δεδομένων.

4.3.1 Few Shot Learning

Το Few Shot Learning αποτελεί μια διαδικασία εκπαίδευσης, που εφαρμόζεται κυριως σε σύνολα δεδομένων στα οποία δεν υπάρχει μεγάλο πλήθος δειγμάτων. Αυτή η διαδικασία εκπαίδευσης χωρίζει τα δείγματα σε ένα σύνολο απο tasks επιλέγοντας ένα συγκεκριμένο αριθμό δειγμάτων απο έναν αριθμό κλάσεων που επιλέγονται. Αυτά τα προβλήματα εκπαίδευσης ονομάζονται k-way n-shot tasks όπου το k αντιπρωσοπευεί τον αριθμό των κλάσεων πάνω στις οποίες πρέπει να πραγματοποιήθει η ταξινόμηση των δεδομένων εισόδου και το n αντιπρωσοπεύει τον αριθμό των διαθέσιμων εικόνων για εκπαίδευση ανα κλάση.

Πιο συγκεκριμένα για την υλοποίηση του Few shot Learning, αρχικά φορτώθηκαν τα δεδομένα και πραγματοποιήθηκαν σε αυτά κάποια απλά augmentations. Στην συνέχεια με την χρήση ενός Data generator που υλοποιήθηκε, δημιουργήθηκαν τα απαραίτα tasks. Για την δημιουργία του κάθε task, επιλέχθηκαν τυχαίες κλάσεις μέχρι να ικανοποιούν τον επιθυμητό αριθμό k και για την κάθε μια απο αυτές τις k κλάσεις δειγματολήφθηκαν τυχαία n δείγματα απο όλα τα δεδομένα, που είναι διαθέσιμα για αυτήν την κλάση. Εκτός απο τα δείγματα που χρησιμοποιήθηκαν για την εκπαίδευση, επιλέχθηκαν τυχαία δεδομένα για validation που ονομάστηκαν queries αλλά και τα δείγματα, που επιλέχθηκαν για το testing, τα οποια ονομάστηκαν test. Για κάθε task συλλέχθηκαν τυχαία δείγματα για όλες τις κατηγορίες support, query, test και ο αλγόριθμος δημιουργίας των tasks υλοποιήθηκε με τέτοιο τρόπο, ώστε δεδομένα που έχουν επιλεχθει σε μια απο τις τρείς κατηγοριές να μην μπορούν να επιλεχθούν σε άλλη.

Αφου δημιουργήθκε ο επιθυμητός αριθμός tasks σύμφωνα με την διαδικασία που περιγράφηκε προηγουμένως τα δεδομένα χωρίστηκαν σε μικρότερα batches, τα οποία ονομάστηκαν mini batches και αυτά χρησιμοποιήθηκαν ως είσοδοι στο μοντέλο εκπαίδευσης. Πριν απο την είσαγωγή τους, όμως στο μοντέλο εκπαίδευσης εφαρμόζεται στα δεδομένα μια απο τις τεχνικές επαύξησης (augmentation), που αναλύσαμε προηγουμένως (cutout, cutmix, mixup) με πιθανότητα 50%. Αφου περάσουν όλα τα mini_batches απο το μοντέλο ολοκληρώνεται ένας κύκλος εκπαίδευσης και μετά ακολουθεί το validation το οποίο πραγματοποιείται στα queries.

4.3.2 Μοντέλο εκπαίδευσης στο MetaMed

Για την εκπαίδευση μοντέλων που χρησιμοποιούν την τεχνική του Meta learning επιλέγονται συστήματα τα οποία αποτελούνται απο μικρό αριθμό παραμέτρων. Αυτό συμβαίνει επειδή θέλουμε ένα μοντέλο, το οποίο αποτελείται απο βάρη (weights), που μπορούν να προσαρμοστούν εύκολα σε μια κατανομή εκτός πεδίου σε σχέση με αυτήν πάνω στην οποία τα εκπαίδευσαμε. Ένας ακόμα λόγος που επιλέγονται μοντέλα με μικρό αριθμό παραμέτρων είναι το γεγονός οτι εκπαίδευουμε το μοντέλο πάνω σε μικρό αριθμό δεδομένων έτσι σε περίπτωση που επιλέγαμε μοντέλα με μεγάλο αριθμό παραμέτρων θα οδηγούμασταν πολύ γρήγορα σε overfitting και το μοντέλο δεν θα πραγματοποιούσε καλή γενίκευση (generalization).

Στην περίπτωση του MetaMed επιλέχθηκε ένα δίκτυο, το οποίο αποτελείται απο τέσσερα block συνελικτικών νευρωνικών δικτύων. Κάθε block αποτελείτε απο ενα συνελικτικό δίκτυο, το οποίο εξάγει 32 απεικονίσεις χαρακτηριστικών με διαστάσεις 3x3, στην συνέχεια σε αυτές τις απεικονίσεις χαρακτηριστικών εφαρμόζεται μια συνάρτηση max-pooling με βήμα 2x2, δηλαδή η εικόνα χωρίζεται σε τετράγωνα διαστάσεων 2x2 pixel και απο αυτό επιλέγεται, αυτό που έχει την μεγαλύτερη τιμή. Μετα το max-pooling εφαρμόζεται μια Relu function, η οποία λειτουργεί ως κατώφλι για το πια χαρακτηρισιτικά είναι πιο σημαντικά για την ταξινόμηση. Το τελευταίο layer είναι το Batch Normalization, το οποίο εφαρμόζεται αποκλειστικά κατα την διάρκεια της εκπαίδευσης training και πραγματοποιεί μια κανονικοποίηση των εισόδων που εισέρχονται στο μοντέλο, δηλαδή των batches, με στόχο να κάνει την εκπαίδευση πιο γρήγορη αλλά και πιο σταθερή. Μετά απο τα 4 συνελικτικά blocks ακολουθούν layers fully connected δικτύων, τα οποία στην έξοδο τους καταλλήγουν σε διανύσματα με διαστάσεις ίσες με τον αριθμό των κλάσεων. Τέλος εφαρμόζεται μια συνάρτηση softmax, που δίνει ως έξοδο τα confidence score για την κατηγοριοποίηση των δειγμάτων εισόδου σε κάθε κλάση.

4.3.3 Αλγόριθμοι εκπαίδευσης Meta training

Οι αλγόριθμοι Meta training αναπτύχθηκαν βασισμένοι στην ικανότητα του ανθρώπου να προσαρμόζεται σε καινούρια tasks διαθέτοντας λίγες πληροφορίες. Ο τρόπος που προσαρμόζεται σε αυτά τα tasks ο άνθρωπος είναι να χρησιμοποιεί πρότερες γνώσεις, που έχει αποκτήσει απο άλλες παρόμοιες εργασίες. Έτσι σκοπός του meta learning είναι να εκπαιδεύσει ένα μοντέλο σε ενα σύνολο απο tasks εκμάθησης, ώστε να μπορέσει να επιλύσει άλλα tasks διαθέντοντας έναν μικρό αριθμό δειγμάτων εκπαίδευσης. Συνεπώς το μοντέλο εκπαιδεύεται με σκοπό οι παράμετροι του να μπορούν, να προσαρμοστούν, μετά απο περιορισμένο αριθμό gradient steps, με περιορισμένο αριθμό δεδομένων εκπαίδευσης σε ένα καινούριο task και να παράξουν καλή γενίκευση στο task. Πιο συγκεκριμένα η μέθοδος του meta learning προσπαθεί να εκπαιδεύσει μοντέλα, τα οποία θα παράγουν παραμέτρους, οι οποίες θα είναι πιο εύκολο να προσαρμοστούν σε καινούρια δεδομένα κατα την διαδικασια του fine-tuning.

Η μέθοδος αυτή προσπαθεί να δημιουργήσει ένα γρήγορο και εύκολα προσαρμόσιμο μοντέλο κατι που αποτελεί πρόκληση, λόγω του περιορισμένου αριθμού νεων δεδομένων, που έχει το μοντέλο καθώς πρέπει να χρησιμοποιήσει την γνώση, που είχε απο τα προηγούμενα tasks, στα οποία είχε εκπαιδευτεί και να προσαρμοστεί στα καινούρια λίγα δεδομένα

αποφεύγοντας το overfitting

MAML(Model-Agnostic Meta-Learning)[46]

Το MAML αποτελεί μια μέθοδο, η οποία μπορεί να γενικευθεί σε μεγάλο βαθμό και να χρησιμοποιηθεί σε κάθε πρόβλημα, η εκπαίδευση του οποίου βασίζεται σε μέθοδο gradient descent. Η βασική ιδέα αυτης της μεθόδου είναι η εκπαίδευση των αρχικών παραμέτρων του μοντέλου με τέτοιο τρόπο, ώστε να μεγιστοποιηθεί η αποδοσή του σε ένα καινούριο task πάνω στο οποίο θα εκπαιδευτεί για λίγα βήματα με την μέθοδο gradient descent και σε περιορισμένο αριθμό δεδομένων. Αυτή η μέθοδος εκπαίδευσης μπορεί να παράγει καλά αποτελέσματα σε ενα καινούριο task, διοτι μπορεί να αποτυπωθεί ως μια διαδικασία εκπαίδευσης που παράγει απεικονίσεις χαρακτηριστικών, τα οποία είναι ικανά να προσαρμοστούν σε πολλά tasks.

Συνεπώς αυτή η διαδικασία εκπαίδευσης βελτιστοποιείται για μοντέλα, που μπορούν εύκολα και γρήγορα να προσαρμοστούν σε δεδομένα μιας καινούριας κατανομής επαναπροσαρμόζοντας μόνο κάποια απο τα υψηλότερα επίπεδα του μοντέλου (layers), όπως τα fully-connected, feed-forward layers του δικτύου. Η συγκεκριμένη λοιπόν διαδικασία εκπαίδευσης μπορεί, να θεωρηθεί ως μια προσπάθεια μεγιστοποίησης της ευαισθησίας της συνάρτησης σφάλματος (loss function) στα καινούρια δεδομένα ως προς τις παραμέτρους του δικτύου. Όπου οταν αυτή η ευαισθησία είναι μεγάλη, μικρές αλλαγές στις παραμέτρους μπορούν να οδηγήσουν σε μεγάλες βελτιώσεις του σφάλματος σε καινούρια tasks.

Όπως εξηγήσαμε και προηγουμένως σκοπός των τεχνικών του meta-learning είναι η εκπαίδευση ενος γρήγορα προσαρμόσιμου δίκτυου σε καινούρια δεδομένα, για να επιτευχθεί αυτό το μοντέλο πραγματοποιείται μια αρχική εκπαίδευση (meta-learning) σε ένα σύνολο απο tasks, με σκοπό αυτή η διαδικασία να παράγει παραμέτρους για το μοντέλο, οι οποίες με περιορισμένη εκπαίδευση σε νεα δεδομένα θα μπορούν να παραξούν αξιόπιστα και επαρκή αποτελέσματα. Έτσι μπορούμε να ορίσουμε ένα μοντέλο ως f, το οποίο δέχεται ως εισόδους δεδομένα x και τα απεικονίζει ως εξόδους a. Επειδή σκοπός της μεθόδου MAML είναι η δυνατότητα προσαρμογής σε πολλά και διαφορετικά tasks, επιλέχθηκε ένας τρόπος εκπαίδευσης γενικού σκοπού. Πιο συγκεκριμένα κάθε task μπορεί να οριστεί ως

$$\mathcal{T} = \{\mathcal{L}(x_1, a_1, ..., x_H, a_H), q(x_1), q(x_t + 1|x_t, a_t), \mathcal{H}\},$$

που αποτελείται απο μια loss function $\mathcal{L}$, δηλαδή μια κατανομή πάνω στις αρχικές παρατηρήσεις, η οποία κατανομή αναπτύχθηκε στο στάδιο του meta-learning $q(x_1)$, μια κατανομή μετατροπής απο τις αρχικές παρατηρήσεις στις καινούριες $q(x_t + 1|x_t, a_t)$ και μια παραμετρό που δείχνει το μήκος ενός επεισοδίου $\mathcal{H}$ κατα το episodic training. Έτσι το μοντέλο παράγει δείγματα με μήκος $\mathcal{H}$, επιλέγοντας μια απο τις εξόδους a_t σε κάθε χρονική στιγμή t. Σε αυτές τις περιπτώσεις η συνάρτηση σφάλματος $\mathcal{L}(x_1, a_1, ..., x_H, a_H) \rightarrow \mathbb{R}$ παράγει αποτελέσματα σχετικά με το σφάλμα ταξινόμησης, που αφορούν το συγκεκριμένο task. Στην διαδικασία εκπαίδευσης του meta-learning θεωρούμε μια κατανομή $p(\mathcal{T})$, την οποία θέλουμε να προσσεγγίσει το μοντέλο. Κατά την διαδικασία της εκπαίδευσης επιλέγουμε ενα task $\mathcal{T}_i$ απο την κατανομή $p(\mathcal{T})$ χρησιμοποιώντας k δείγματα και παίρνοντας ως πληροφορία το $\mathcal{L}_{\mathcal{T}_i}$. Αφου το μοντέλο πάρει αυτήν την πληροφορία του σφάλματος εκπαιδεύεται σε ενα άλλο

task, οταν το μοντέλο εκπαιδευτεί σε όλα τα tasks, συλλέγονται όλα τα σφάλματα που έχουν προκύψει και ανανεώνονται οι παράμετροι του μοντέλου f με βάση το σφάλμα στα δεδομένα, που χρησιμοποιούνται ως test set (queries). Στην συνέχεια οταν ξεκινήσει η εκπαίδευση της καινούριας εποχής του episodic training, επιλέγονται καινούρια τυχαία tasks $\mathcal{T}_\iota$ απο την κατανομή $p(\mathcal{T})$

Ο αλγόριθμος MAML βασίζεται στην ιδεα οτι κάποιες αναπραστάσεις μπορούν να μεταφερθούν πιο εύκολα σε διαφορετικά tasks και ο τρόπος που χρησιμοποιείται απο την μέθοδο, για να ενθαρρύνει τέτοιες γενικού σκοπού αναπαραστάσεις είναι με την χρήση gradient μεθόδων. Έτσι σκοπός είναι η εκπαίδευση ενός μοντέλου με τέτοιο τρόπο, ώστε ο gradient τρόπος εκπαίδευσης, που χρησιμοποιείται να πραγματοποιεί μεγάλη πρόοδο σε καινούρια tasks, που προέρχονται απο την κατανομή $p(\mathcal{T})$ χωρις να οδηγούνται σε υπερεκπαίδευση (overfitting). Ο τρόπος που πραγματοποιείται αυτό είναι με εύρεση μοντέλων τα οποία ε-ίναι πολύ ευαίσθητα σε αλλαγές στα tasks, έτσι ώστε μικρές αλλαγές στις παραμέτρους να μπορούν να παράγουν μεγάλες βελτιώσεις στην συνάρτηση σφάλματος για κάθε task που παράγεται απο την κατανομή $p(\mathcal{T})$.

Θεωρούμε ένα μοντέλο το οποίο απεικονίζεται απο την συνάρτηση f_θ με παραμέτρους θ. Οταν το μοντέλο προσαρμόζεται σε ενα καινούριο task $\mathcal{T}_\iota$ οι παράμετροι του απο θ γίνονται $\theta_t^{'}$. Στην συγκεκριμένη μέθοδο οι παράμετροι $\theta_t^{'}$ υπολογίζονται μετα απο ένα ή περισσότερα βήματα της μεθόδου gradient descent σε ενα task $\mathcal{T}_\iota$. Εαν πραγματοποιείται ενα βήμα στο gradient descent δηλαδη μια ανανέωση των παραμέτρων τότε η εξίσωση μπορεί να γραφτεί ως

$$\theta_t^{'} = \theta - a\nabla_\theta \mathcal{L}_{\mathcal{T}_\iota}(f_\theta),$$

όπου το a αποτελεί μια υπερ-παράμετρο που αναφέρεται στο μέγεθος του βήματος, που πραγματοποιεί το gradient descent. Η παραπάνω εξίσωση αναφέρεται σε ανανεώσεις παραμέτρων που λαμβάνουν υπόψιν μόνο το gradient πρώτου βαθμού για χάρη απλοποίησης των αναπαραστάσεων, αλλά η συνολική διαδικασια είναι ταυτόσημη και οταν λαμβάνονται υποψιν οι μεγαλύτεροι βαθμοί του gradient..

Η μαθηματική σχέση που απεικονίζει τον τρόπο με τον οποίο εκπαιδευεταί το μοντέλο πανω σε μια εποχή του episodic training περιγράφεται απο την παρακάτω σχέση.

$$\min_\theta \sum_{n=1}^{\infty} \mathcal{L}_{\mathcal{T}_\iota}(f_{\theta_\iota'}) = \sum_{\mathcal{T}_\iota \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_\iota}(f_{\theta - a\nabla_\theta \mathcal{L}_{\mathcal{T}_\iota}(f_\theta)})$$

Σκοπός αυτού του meta-optimization τρόπου εκπαίδευσης είναι ο υπολογισμός παραμέτρων του μοντέλου, ετσι ώστε ένας μικρός αριθμός gradient βημάτων πάνω σε ενα task να παράγει αποτελέσματα, που έχουν καλη απόδοση. Για την διαδικασία του meta-optimazation πάνω σε ολα τα tasks χρησιμοποιείται το stochastic gradient descent (SGD)[74], για να ανανεωθούν οι παράμετροι του. Ο τρόπος με τον οποίο πραγματοποιείται το meta-optimazition φαίνεται στην παρακάτω εξίσωση

$$\theta \leftarrow \theta - \sum_{n=1}^{\infty} \mathcal{L}_{\mathcal{T}_\iota}(f_{\theta_\iota'}) \tag{4.5}$$

Ένα απο τα μεγαλύτερα υπολογιστικά κόστη του αλγορίθμου MAML είναι οτι κατα το

βήμα του meta-update, όπου ανανεώνονται οι γενικές παράμετροι του δικτύου απαιτείται ενα επιπλέον backward pass απο το μοντέλο f, ωστε να υπολογιστεί το Hessian γινόμενο.

Εξηγώντας αναλυτικότερα τον αλγόριθμο MAML θεωρουμε ενα συνολο δεδομένων dataset, απο αυτό δειγματολειπτούμε τυχαία tasks $\mathcal{T}_\iota$, όπου κάθε task παράγει K δείγματα πάνω στα οποία εκπαιδεύεται το μοντέλο. Τα δείγματα που παράγει το task, εαν πρόκειται για few shot learning αποτελούνται απο k τυχαία δείγματα για κάθε n τυχαία κλάση που έχει επιλεχθει για το συγκεκριμένο task, όπου τα k, n αναπαριστούν υπερπραμέτρους, που επιλέγονται στην αρχή της εκπαίδευσης. Το loss του task υπολογίζεται ως το σφάλμα μεταξύ της εξόδου του μοντέλου x και του label y για κάθε δείγμα. Για το πρόβλημα του της κατηγοριοποιήσης, το σύνηθες loss function που χρησιμοποιείται, είναι το cross-entropy loss που περιγράφεται απο την παρακάτω εξίσωση

$$\mathcal{L}_{\mathcal{T}_\iota}(f_{\partial_\phi}) = \sum_{x^{(j)}, y^{(j)} \sim \mathcal{T}_\iota} y^{(j)} \log(f_\phi(x^{(j)})) + (1 - y^{(j)}) \log(1 - f_\phi(x^{(j)})) \tag{4.6}$$

Αφου υπολογιστεί το loss 4.7 για τα δείγματα του task που χρησιμοποιούνται στην εκπαίδευση, στην συνέχεια με την βοήθεια αυτού και του συνόλου δεδομένων $\mathcal{D}$, υπολογίζεται το gradient του loss $\nabla_\partial \mathcal{L}_{\mathcal{T}_\iota}$. Αφου υπολογίστει το gradient, στην συνέχεια με την βοήθεια του optimizer SGD πραγματοποιείται η ανανέωση των βαρών 4.5. Έπειτα επιλέγεται ενα τυχαίο σύνολο δειγμάτων $\mathcal{D}_\iota^{'} = \{x^{(j)}, y^{(j)}\}$ απο κάθε task $\mathcal{T}_\iota$ με σκοπό να χρησιμοποιηθούν στο βήμα του meta-update. Κατα το meta-update χρησιμοποιούναι τα δειγματα $\mathcal{D}_\iota^{'} = \{x^{(j)}, y^{(j)}\}$ που συλλέχθηκαν απο κάθε task και οι loss function $\mathcal{L}_{\mathcal{T}_\iota}(f_{\partial_\iota'})$ που υπολογίστηκαν απο την 4.7 για κάθε task. Η εξίσωση που χρησιμοποιείται για το meta-update περιγράφεται απο την παρακάτω σχέση

$$\partial \leftarrow \partial - \beta \nabla_\partial \sum_{\mathcal{T}_\iota \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_\iota}(f_{\partial_\iota'}) \tag{4.7}$$

Αυτή η διαδικασία επαναλαμβάνεται για συγκεκριμένο αριθμό βημάτων, που συνήθως ορίζεται ως μια υπερπαράμετρος στην αρχή της εκπαίδευσης.

Όπως αναλύθηκε και προηγουμενως ο αλγόριθμος MAML πραγματοποιεί μια εκπαίδευση, η οποία οδηγεί στην εκμάθηση παραμέρων, οι οποίες είναι ικανές να προσαρμοστούν γρηγορα και αποδοτικά σε δεδομένα που ακολουθούν διαφορετικές κατανομές απο τα δεδομένα στα οποία είχε εκπαιδευτεί αρχικά το μοντέλο. Ο τρόπος με τον οποίο πραγματοποιείται αυτο είναι με την δημιουργία ευαίσθητων παραμέτρων, που μπορούν να εντοπίσουν γρήγορα αλλάγες στα χαρακτηριστικά των tasks και να προσαρμοστουν πάνω σε αυτές. Ενα ομως απο τα σημαντικότερα προβλήματα αυτού του αλγοριθμού είναι το υπολογιστικό κόστος, που προέρχεται απο τον υπολογισμό των παραγώγων δεύτερης τάξης στο βήμα του meta-update. Επισης στο [46] συγκρίθηκαν οι επιδόσεις του MAML, με χρήση παραγώγων πρώτου και και δευτέρου βαθμού για το βήμα του meta-update σε σύγκριση με την χρήση μόνο πρώτων παραγώγων. Τα αποτελέσματα έδειξαν οτι αυτές οι δυο μέθοδοι δεν είχαν σημαντικές διαφορές στις επιδόσεις μεταξύ τους. Εκμεταλευόμενοι αυτές τις μικρές διαφορες στις επιδόσεις μεταξύ των δυο μεθόδων αναπτύχθηκε ο αλγόριθμος Reptile [75], που αποτελεί μια πιο απλή έκδοση του MAML, που χρησιμοποιεί παραγώγους μόνο πρώτου βαθμού.

Reptile[75]

Εμπνευσμενοί απο τον αλγόριθμο MAML οι Nichol, Schulman[75] ανέπτυξαν τον αλγόριθμο Reptile, ο οποίος έχει ως σκοπό την παρουσίαση μιας πιο απλής έκδοσης του αλγορίθμού MAML, που να αποδίδει καλά στην εκμάθηση ενός καινούριου task που του δίνεται, το οποίο ανήκει σε παρόμοια κατανομή πάνω στην οποία είχε εκπαιδευτεί. Ο αλγόριθμος Reptile λειτουργεί δειγματολειπτώντας επαναληπτικά τυχαία tasks, στην συνέχεια εκπαιδεύεται πάνω σε αυτά και με αυτόν τον τρόπο μετατοπίζει τις παραμέτρους, που χρησιμοποιούνται για την αρχικοποίηση πιο κοντά στις παραμέτρους που έχουν εκπαιδευτεί πάνω στο task. Η κύρια διαφορά του Reptile με το MAML ειναι οτι στον πρώτο δεν χρειάζεται διαφοροποίηση μέσω της μεθόδου optimization κάνοντας τον πολυ χρήσιμο στη επίλυση προβλημάτων βελτιστοποίησης που χρειάζονται πολλά βήματα ανανέωσης.

Πιο συγκεκριμένα για τον αλγόριθμο Reptile θεωρούμε οτι έχουμε πρόσβαση σε μια κατανομή δεδομένων που αποτελείται απο tasks, όπου κάθε task αποτελεί ενα συγκεκριμένο πρόβλημα (πρόβλημα ταξινόμησης, πρόβλημα κατάτμησης). Απο αυτήν την κατανομή χωρίζουμε δυο σύνολα δεδομένων ενα για εκπαίδευση (training set) και ενα για έλεγχο του μοντέλου (testing set). Ο αλγόριθμος λοιπόν τροφοδοτείται με το training set και πιο συγκεκριμένα με τυχαία επαναλαμβανόμενα tasks, που εξάγονται απο το training set και με αυτό τον τρόπο εκπαιδεύεται, ώστε να παράξει ενα μοντέλο, το οποίο θα παρέχει παραμέτρους, που θα έχουν μια καλή μέση απόδοση στο test set. Αφου κάθε task περιγράφει ενα πρόβλημα εκπαίδευσης η καλή απόδοση σε αυτό αντιστοιχεί σε γρήγορη εκμάθηση του μοντέλου. Η μέθοδος του meta learning έχει προσεγγιστεί απο πολλές διαφορετικές οπτικές γωνιές. Μια μέθοδος χρησιμοποιεί reccurent neural networks (RNN) στον αλγόριθμο εκπαίδευσης για την εκμάθηση των κατάλληλων παραμέτρων, αλλά δεν χρησιμοποιεί μεθόδους gradient descent, κατα την διαδικασία του ελέγχου. Σε τέτοιες περιπτώσεις χρησιμοποιήθηκαν δίκτυα LSTM για την πρόβλεψη στο επόμενο βήμα, όπως περιγραφηκε απο τον Hochreiter et al.[76]. Μια άλλη προσέγγιση είναι η εκμάθηση μιας καλής αρχικοποίησης των παραμέτρων σε ενα μεγαλύτερο σύνολο δεδομένων (όπως το ImageNet[77]) και μετα η προσαρμογή(fine-tune) αυτων των παραμέτρων στο επιθυμητό dataset, τέτοιες προσεγγίσεις φαίνονται στο [78]. Το προβλημα με αυτήν την μέθοδο είναι οτι δεν ειμαστε σιγουροι αν η αρχικοποίηση των παραμέτρων, που θα πάρουμε θα είναι καταλλήλη για να μας επιστέψει μια καλή αναπράσταση οταν θα προσαρμόσουμε (fine-tuning) τις αναφερθείσες παραμέτρους στο μικρότερο σύνολο δεδομένων. Όπως περιγράφηκε και προηγούμενως ο αλγόριθμός MAML αποτελεί μια προσέγγιση για την επίλυση αυτου του προβλήματος, αφου βελτιώνει την απόδοση λαμβάνοντας υποψιν την αρχικοποίηση, ωστόσο το προβλημα που εμφανίζεται με τον αλγόριθμο MAML είναι το μεγάλο υπολογιστικό κόστος λόγω των πολλών gradient step κατα την διάρκεια του ελέγχου του αλγορίθμου.

Ο αλγόριθμος Reptile αποτελεί μια απλοποίηση του MAML, αφου μαθαίνει μια αρχικοποίηση για τις παραμέτρους ενος νευρωνικού δικτύου, με σκοπό την γρήγορη εκμάθηση των παραμέτρων στο βήμα της βελτιστοποίηση (optimization) κατα την διάρκεια του ελέγχου. Παρακάτω φαίνεται ο αλγόριθμός Reptile για n στο πλήθος επαναλήψεις, σε κάθε επανάληψη θεωρουμε οτι δειγματολειπτούνται m τυχαία tasks τ. Επίσης θεωρουμε ϕ διάνυσμα των παραμέτρων του μοντέλου και το $SGD(\mathcal{L}_\tau, \phi, k)$ απεικονίζει την συνάρτηση gradient descent,

που πραγματοποιεί k βήματα στο σφάλμα $\mathcal{L}$ ξεκινώντας απο τις παραμέτρους ϕ

Αλγοριθμος 4.1: *Reptile (batched version)[75]*

Αρχικοποίησε ϕ
for iteration = 1, 2, . . ., n **do** Δημιουργία Tasks $\tau_1, \tau_2, \ldots, \tau_m$
  for $i = 1, 2, \ldots, m$ **do**
    Υπολόγισε $W_i = \text{SGD}(\mathcal{L}_{\tau_i}, \phi, k)$
  end for
  Ανανέωσε $\phi \leftarrow \phi + \frac{\epsilon}{k} \sum_{i=1}^{m} (W_i - \phi)$
end for

Η κύρια διαφορά του παραπάνω αλγορίθμου με την εκπαίδευση πανω στο αναμενόμενο σφάλμα $\mathbb{E}[\mathcal{L}_{\mathcal{T}}(f_{\delta_t'})]$ είναι οτι στον αλγόριθμο Reptile πραγματοποιούνται πολλάπλα βήματα gradient descent κατα την διαδικασια του minimization του expected error, έτσι το μέσω update δεν είναι ισο με την αναννέωση στην μέση συνάρτηση. Άρα η παρακάτω εξίσωση δεν ισχύει για $k > 1$

$$E_\tau[\text{SGD}(\mathcal{L}_\tau, \phi, k)] \neq \text{SGD}(E_\tau[\mathcal{L}_\tau], \phi, k)$$

Αυτό συμβαινει, διοτι το αναμμενόμενο update εξαρταται απο τις παραμέτρους μεγαλύτερης τάξης του $\mathcal{L}_\tau$. Παρακάτω φαίνονται οι εξισωσείς με την χρήση της σειράς Taylor, που προσεγγίζουν τις αναβαθμίσεις, που γίνονται στις παραμέτρους του μοντέλου κατα την εκτέλεση των αλγορίθμων First Order MAML (FOMAML) και Reptile. Θεωρούμε για ευκολία πράξεων δυο βήματα SGD πρώτα στο σφάλμα $\mathcal{L}$, και μετα στο σφάλμα $\mathcal{L}_\infty$ συνεπώς τα 0,1 αναφέρονται σε διαφορετικά minibatches δεδομένων απο το ίδιο task. Θεωρούμε επίσης ϕ τις αρχικές παραμέτρους και a το βήμα ανανέωσης. Επίσης θεωρουμε τις παραγώγους $\mathcal{L}'_{(\phi)} = \frac{\partial}{\partial \phi}\mathcal{L}(\phi)$, $\mathcal{L}''_{(\phi)} = \frac{\partial^2}{\partial \phi^2}\mathcal{L}(\phi)$. Μετα απο δυο βήματα του SGD οι παράμετροι γίνονται

$$\phi_0 = \phi \tag{4.8}$$
$$\phi_1 = \phi_0 - a\mathcal{L}'_0(\phi_0) \tag{4.9}$$
$$\phi_2 = \phi_0 - a\mathcal{L}'_0(\phi_0) - a\mathcal{L}'_1(\phi_1) \tag{4.10}$$

Παρακάτω θα εφαρμοστεί η σειρά Taylor για να προσεγγιστεί η πρώτη παράγωγος του $\mathcal{L}'_1(\phi_1)$, που χρησιμοποιείται για να υπολογίστει το gradient απο τους αλγορίθμους Reptile και MAML

$$L'_1(\phi_1) = L'_1(\phi_0) + L''_1(\phi_0)(\phi_1 - \phi_0) + O(a^2) = L'_1(\phi_0) - aL''_1(\phi_0)L''_0(\phi_0) + O(a^2)$$

Ετσι η παράγωγος του gradient υπολογίζεται ως

$$g_{Reptile} = \frac{\phi_0 - \phi_2}{a} = L'_0(\phi_0) + L'_1(\phi_1) = L'_0(\phi_0) + L'_1(\phi_0) - aL''_1(\phi_0)L'_0(\phi_0) + O(a^2) \tag{4.11}$$

Και η παράγωγος του MAML υπολογίζεται ως

$$\begin{aligned} g_{MAML} &= \frac{\partial}{\partial \phi_0} L_1(\phi_1) = \frac{\partial \phi_1}{\partial \phi_0} L_1^{'}(\phi_1) = (I - aL_0^{"}(\phi_0))L_1^{'}(\phi_1) \\ &= (I - aL_0^{"}(\phi_0))(L_1^{'}(\phi_0) - aL_1^{"}(\phi_0)L_0^{'}(\phi_0)) + O(a^2) \\ &= L_1^{'}(\phi_0) - aL_1^{"}(\phi_0)L_0^{'}(\phi_0) - aL_0^{"}(\phi_0)L_1^{'}(\phi_0) + O(a^2) \end{aligned} \tag{4.12}$$

Επειδή χρησιμοποιούμε το First order MAML το gradient $L_1^{'}(\phi_1)$ μετα απο την πρώτη ανανέωση θεωρείτε σταθερό άρα η δεύτερη παράγωγος του είναι μηδενική συνεπώς, η 4.12 γράφεται

$$g_{FOMAML} = L_1^{'}(\phi_0) - aL_1^{"}(\phi_0)L_0^{'}(\phi_0) + O(a^2) \tag{4.13}$$

Υπολογίζοντας το expectation πάνω στα tasks τ και παίροντας δυο minibatches που ορίζουν τα $L_0(\phi_1)$ και $L_1(\phi_1)$ παρατηρούμε οτι εμφανίζονται δυο όροι. Αρχικά, προκύπτει το AvgGrad = $\mathbb{E}_{\tau,0}[L_0^{'}(\phi_0)]$, το οποίο αποτελει το gradient του αναμμενόμενου σφάλματος. Επίσης, η αρνητική συνιστώσα αυτου του όρου απεικονίζει την κατεύθυνση που οδηγει τις παραμέτρους ϕ προς την ελαχιστοποίηση του σφάλματος.

Ο δεύτερος όρος που εμφανίζεται είναι το

$$\begin{aligned} AvgGradInner &= \mathbb{E}_{\tau,0,1}[L_0^{"}(\phi)L_1^{'}(\phi)] \\ &= \frac{1}{2}\mathbb{E}_{\tau,0,1}[L_0^{"}(\phi)L_1^{'}(\phi) + L_1^{"}(\phi)L_0^{'}(\phi)] \\ &= \frac{1}{2}\mathbb{E}_{\tau,0,1}[\frac{\partial}{\partial \phi}L_0^{"}(\phi)L_1^{'}(\phi)] \end{aligned} \tag{4.14}$$

Οπου η αρνητική συνιστώσα του παραπάνω όρου απεικονίζει την κατεύθυνση, η οποία αυξάνει το εσωτερικό γινόμενο μεταξύ των gradients διαφορετικών minibatches για ενα συγκεκριμένο task το οποίο βελτιώνει την γενίκευση.

$$\begin{aligned} \mathbb{E}[g_{Reptile}] &= 2AvgGrad - a \cdot AvgGradInner + O(a^2) \\ \mathbb{E}[g_{MAML}] &= AvgGrad - 2a \cdot AvgGradInner + O(a^2) \\ \mathbb{E}[g_{FOMAML}] &= AvgGrad - a \cdot AvgGradInner + O(a^2) \end{aligned} \tag{4.15}$$

Απο τις παραπάνω σχέσεις φαίνεται οτι όλες οι εξισώσεις των gradient αρχικά, οδηγούν στην ελαχιστοποίηση του αναμμενόμενου σφάλματος πάνω στα tasks, στην συνέχεια ο όρος του υψηλότερου βαθμού AvgGradInner βοηθάει στο fast learning. Η ανάλυση με την χρήση σειρών Taylor δείχνει οτι η εκπαίδευση με την χρήση μεθόδων stohastic gradient descent πραγματοποιεί μια ανανέωση των παραμέτρων, που μοιάζει με την διαδικασια που ακολουθεί ο αλγόριθμος MAML, που μεγιστοποιεί την γενίκευση μεταξύ διαφορετικών minibatches. Αυτό μπορεί να εξηγήσει το γιατι οι τεχνικές pre-training στο ImageNet και fine-tuning στο επιθυμητό dataset με stohastic gradient descent παραγούν μια καλή αρχικοποίηση στις παραμέτρους που γενικεύουν καλύτερα σε παρόμοια tasks. Αυτή η υπόθεση δείχνει οτι η διαδικασία του joint training plus και στην συνέχεια η εφαρμογή του fine-tuning αποτελούν μια ισχυρή μέθοδο για την επιλύση προβλημάτων μηχανικής μάθηση στον τομέα του meta-learning.

4.3.4 Διαδικασία εκπαίδευσης MetaMed[2]

Ο αλγόριθμος εκπαίδευσης του MetaMed ακολουθεί παρόμοια λογική με την διαδικασία εκπαίδευσης, που περιγράφηκες στο MAML[46] και με τον τρόπο που παρουσιάστηκε στο Reptile[75]. Στην συγκεκριμένη περίπτωση το πρόβλημα που αντιμετωπίζει ο αλγόριθμος είναι αυτό της κατηγοριοποίησης σε long-tailed distributions, δηλάδη θέλουμε να επιλύσουμε το πρόβλημα της ταξινόμησης σε σύνολα δεδομένα των οποίων οι κλάσεις δεν είναι ισοκατανεμημένες. Επίσης το few shot learning θα εφαρμοστεί στις κλάσεις για τις οποίες έχουμε περισσότερα δεδομένα (head), ενω το fine-tuning και ο έλεγχος θα πραγματοποιηθούν στις κλάσεις για τις οποίες δεν διαθέτουμε μεγάλο πλήθος δεδομένων (tail of distribution). Η εκπαίδευση θα πραγματοποιηθεί στα τρια σύνολα δεδομένων, που περιγράψαμε προηγουμένως στα υποκεφάλαια (4.1.1,4.1.2,4.1.3). Έτσι μπορούμε να ορίσουμε ενα σύνολο απο dataset, το οποίο το συμβολίζουμε ως εξης $D = \{D_1, D_2, \ldots, D_n\}_{k=1}$, όπου στην συγκεκριμένη περίπτωση το n είναι ισο με 3. Εξετάζοντας κάθε dataset ξεχωριστά μπορούμε να το ορισούμε ως $D_K = \{(x, y)_j\}_{j=1}^{n}$, όπου το x συμβολίζει μια εικόνα του συνόλου δεδομένο και το y συμβολίζει την αντίστοιχη ετικέτα του. Κάθε ενα απο αυτά τα dataset, στην συνέχεια χωρίζεται σε ενα σύνολο, που περιέχει ολά τα δεδομένα, που θα χρησιμοποιηθουν στην εκπαίδευση, το οποίο ονομάζουμε D_{meta_train} και περιέχει συνήθως τις κλάσεις που έχουν τα περισσότερα δεδομένα. Το άλλο σύνολο το συμβολίζουμε D_{meta_test} και περιέχει τις κλάσεις του dataset που έχουν τα λιγότερα δείγματα. Απο τον τρόπο που ορίσαμε τα παραπάνω σύνολα ισχύει η σχέση $D_{meta\_train} \cap D_{meta\_test}$. Με την σειρά τους τα σύνολα δεδομένων $D_{meta\_train}, D_{meta\_test}$ χωριζονται σε σύνολα train, test που θα χρησιμοποιηθούν για εκπαίδευση και έλεγχο, αντίστοιχα αυτά τα σύνολα τα συμβολίζουμε ως $D^{train}_{meta\_train}, D^{test}_{meta\_train}, D^{train}_{meta\_test}, D^{test}_{meta\_test}$.

Ο τρόπος με τον οποίο θα πραγματοποιηθεί η εκπαίδευση κατα την εκτέλεση της διαδικασίας του meta-learning και fine-tuning είναι με την μέθοδο του few-shot learning, που περιγράφηκε και στο υποκεφάλαιο (2.5.1). Η εκπαίδευση χωρίζεται σε επεισόδια, οπου σε κάθε επεισόδιο πραγματοποιείται εκπαίδευση του μοντέλου πάνω σε ένα σύνολο απο tasks, που εξάγονται απο τα train υποσύνολα των συνόλων $D_{meta\_train}, D_{meta\_test}$ κατα την διαδικασία του meta-learning, fine-tuning αντιστοιχά. Έπειτα πραγματοποιείται το βήμα του meta-update, που ανανεωνεί τις γενικές παραμέτρους του δικτύου που στην συνέχεια χρησιμοποιούνται, για να αρχικοποιήσουν το μοντέλο στο επόμενο επεισόδιο. Για την εκπαίδευση του μοντέλου και την ανανέωση των παραμέτρων χρησιμοποιείται ο αλγόριθμος Reptile, που περιγράφηκε στο υποκεφάλαιο (4.3.3). Το τελευταίο βήμα κάθε επεισοδίου στην εκπαίδευση ονομάζεται meta-update και περιγράφεται απο την παρακάτω σχέση.

$$\phi^{'} \leftarrow \phi + \frac{\epsilon}{m} \sum_{i=1}^{m} (\phi^{'}_{k} - \phi) \tag{4.16}$$

Στην σχέση 4.16 το $\phi^{'}$ συμβολίζει τις καινουριές παραμέτρους που υπολογίζονται στο τέλος του επεισοδίου, το ϕ συμβολίζει τις παραμέτρους που είχαν υπολογιστεί στο προηγούμενο επεισόδιο (αν είμαστε στο πρώτο επεισόδιο αυτές η παράμετροι έχουν αρχικοποιηθεί τυχαια) και είχαν φορτώθει στο μοντέλο για την εκπαίδευση πάνω στα tasks, που επιλέχθηκαν τυχαια για το συγκεκριμένο επεισόδιο απο το υποσύνολο train. Τέλος το $\phi^{'}_{k}$ συμβολίζει τις παραμέτρους που προέκυψαν μετα την εκπαίδευση του μοντέλου πάνω στο k task για

την εκπαίδευση αυτών των παραμέτρων χρησιμοποιήθηκε Adam optimizer.

Οταν τελειώσει η εκπαίδευση πάνω στα δεδομένα D_{meta_train} οι παράμετροι $\phi^{'}$ που προέκυψαν στο τελευταίο επεισόδιο φορτώνονται στο μοντέλο και εκτελείτε η διαδικασια του fine-tuning πάνω σε tasks, τα οποία δειγματολειπτούνται απο το σύνολο δεδομένων $D^{train}_{meta\_test}$ και ακολουθούν την διαδικασια δημιουργιας tasks, που περιγράφει η μέθοδος few-shot learning που αναλύθηκε στο υποκεφάλαιο (2.5.1). Για την ανανέωση των παραμέτρων $\phi^{'}$ κατα την εκτέλεση του fine-tuning χρησιμοποιείται επίσης Adam optimizer. Ο αριθμός των επαναλήψεων που θα πραγματοποιηθούν κατα το fine-tuning ορίζεται ως υπερπαράμετρος h. Στο τέλος των επαναλήψεων προκύπτούν οι παράμετροι $\phi^{"}$ που στην συνέχεια χρησιμοποιούνται, για να πραγματοποιήσουν προβλέψεις σχετικά με το σε πια κλάση ανήκουν τα $D^{test}_{meta\_test}$, απο τον αριθμό των σωστών προβλέψεων πανω σε αυτά τα δεδομένα προκύπτει το accuracy του μοντέλου πάνω στο συγκεκριμένο πρόβλημα ταξινόμησης, που χρησιμοποιείται και ως η κύρια μετρική για τον έλεγχο της απόδοσης του αλγόριθμού MetaMed πάνω στα εξεταζόμενα σύνολα δεδομένων. Αξίζει επίσης, να σημειώθει οτι κατα την διάρκεια εκπαίδευσης, τόσο στο meta-learning, όσο και στο fine-tuning για τον υπολογισμό του σφάλματος χρησιμοποιείται το cross entropy loss, που η μαθηματική του εξίσωση για ενα συγκεκριμένο tasks T_K δίνεται απο την σχέση

$$\mathcal{L}_{T_l}(f_\phi) = -\sum_{x_i, y_i \sim T_l} y_i \log(f_\phi(x_i)) + (1 - y_i) \log(1 - f_\phi(x_i)) \tag{4.17}$$

Για καθε επεισόδιο εκπαίδευσης δειγματολειπτούνται m τυχαια tasks και στην συνέχεια για κάθε task πραγματοποιείται εκπαίδευση στο μοντέλο. Πριν την εισαγωγή των δεδομένων στο μοντέλο το task χωρίζεται σε μικρότερα σύνολα δεδομένων τα οποία ονομάζουμε minibatches και είναι αυτά που εισάγουμε στο μοντέλο κατα την διαδικασία εκπαίδευσης. Πριν την εισαγωγή τους, όμως στο μοντέλο, πραγματοποιείται στα δεδομένα του minbatch μια απο τις τεχνικές σύνθετης επαύξησης δεδομένων (4.2.2, 4.2.2, 4.2.2) που αναλύσαμε προηγουμένως με πιθανότητα 50%.

Παρακάτω φαίνεται ο αλγόριθμος, που πραγματοποιεί την εκπαίδευση που περιγράψαμε προηγουμένως και ο αλγόριθμος που πραγματοποιεί τις τεχνικές augmentations.

Αλγοριθμος 4.2: *MetaMed[2]*

Είσοδος:
$D^{test}_{meta\_test}$,
a, ϵ εσωτερικοί και εξωτερικοι ρυθμοι εκμάθησης,
ζ παράμετρος κατανομής,
h αριθμος επαναληψεων κατα το fine-tuning,
N αριθμός tasks,
M αριθμός επεισοδίων
Έξοδος: Μέσο accuracy για N few shot tasks
/* Meta-training Stage */
for iteration = 1, 2, . . . M **do**
Επιλεξε m τυχαία tasks απο το $D_{meta-train}$ σύνολο δεδομένων
for iteration = 1, 2, . . . μ **do**
if augmentation **then**
$\lambda \sim$ Beta(ζ, ζ)
T = MetaTrainAug(T, augtype, λ)
end if
Υπολόγισε το σφάλμα $\mathcal{L}_T(f_\phi) = -\sum_{x_i, y_i \sim T} y_i \log(f_\phi(x_i)) + (1 - y_i) \log(1 - f_\phi(x_i))$
Ανανέωση των παραμέτρων ϕ με χρήση Adam optimizer και του loss $L_T(f_\phi)$ σε ϕ'
end for
Meta-Update: $\phi \leftarrow \phi + \beta \frac{1}{m} \sum_{i=1}^{m} (\phi'_i - \phi)$
end for
/* Meta-testing Stage */
for i = 1 το N **do**
Επέλεξε τυχαίο n-way, k-shot few-shot task T_l, με $T_l \in D_{meta-test}$ και χωρισε το σε $D^{train}_{meta\_test}, D^{test}_{meta\_test}$
Φόρτωσε τις παραμέτρους ϕ' που υπολογίστηκαν στο meta-train
for iteration = 1, 2, . . . η **do**
Υπολόγισε το loss πάνω στο $D^{train}_{meta\_test}$ και ανανέωσε τις παραμέτρους ϕ' σε ϕ'' με την βοήθεια ενος Adam Optimizer
end for
Φόρτωσε τις παραμέτρους ϕ'' και υπολόγισε το accuracy πάνω στο σύνολο δεδομένων $D^{test}_{meta\_test}$
end for

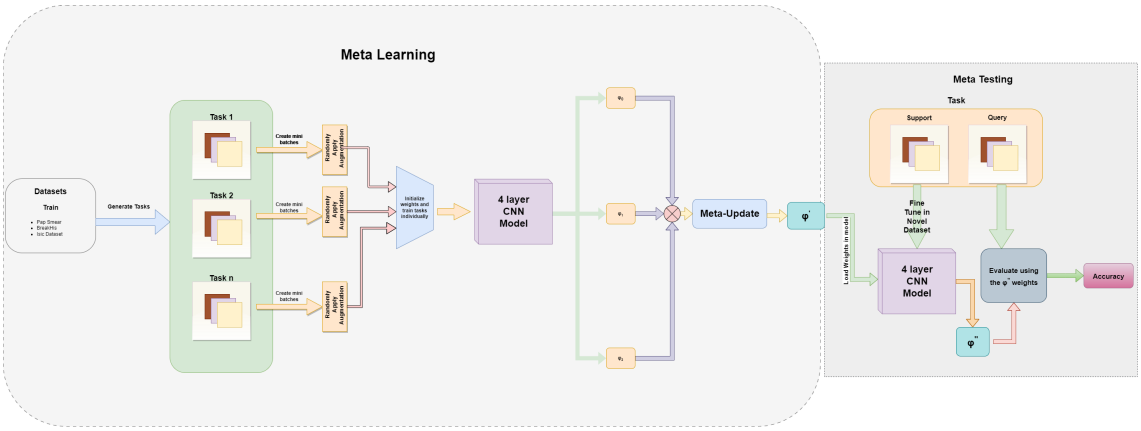
Όπως φαινεται ο αλγόριθμος (4.2) περιγράφει τόσο την διαδικασία του meta-learning όσο και την διαδικασία του fine-tuning. Αρχικά δίνονται τυχαίες αρχικοποιήσεις στις παραμέτρους του μοντέλου, στην συνέχεια εφαρμόζεται το for loop, που εκφράζει των αριθμό των επεισοδίων στο meta training. Για κάθε εκτέλεση αυτου του loop δειγματολειπτούνται τυχαία tasks. Αυτά τα tasks στην συνέχεια χωρίζονται σε minibatches και τους εφαρμόζονται οι τεχνικές augmentation που αναφέρθηκαν και προηγούμενως, η διαδικασία που ακολουθείται για το augmentation περιγράφεται απο τον αλγόριθμο (4.3). Τα αποτελέσματα στην συνέχεια που εξάγονται απο το μοντέλο συγκρίνονται με τα ground truth y και υπολογίζεται το cross entropy loss, το οποίο χρησιμοποιείται απο τον optimizer για την ανανέωση των βαρών. Τέλος μετα την ολοκληρωση της εκπαίδευσης εξάγωνται οι παράμετροι που υπολογίστηκαν για ολα τα tasks m που δειγματολειφθήσαν με σκοπό να υπολογιστεί το meta-update. Οταν πραγματοποιηθεί η εκπαίδευση για ολα τα επεισόδια (M) ακολουθεί το

βήμα του fine-tuning. Στο συγκεκριμένο βήμα εκτελούμε ενα for loop για εναν συγεκριμένο αριθμό few-shot tasks N, ο οποίος συμβολίζει τον αριθμό των tasks πάνω στον οποίο θα ελέγξουμε το μοντέλο με χρήση της μετρικής accuracy, Μέσα στο for loop δειγματολειπτούμε ενα k-way, n-shot task, στην συνέχεια φορτώνουμε τις παραμέτρους, που υπολογίστηκαν στο βήμα του meta-update και πραγματοποιούμε fine-tuning για εναν αριθμό βημάτων h, οταν ολοκληρωθεί αυτή η εκπαίδευση πραγματοποιείται ο έλεγχος για την εξαγωγή του accuracy

Αλγοριθμος 4.3: *MetaTrainAug[2]*

procedure MetaTrainAug(T, augtype, λ)
for each image, label pair $\{(x_i, y_i), i \leq |T|\}$ **do** ▷ Επανάληψη για κάθε task
  if augtype == cutout **then** ▷ CutOut επαύξηση
    Τυχαία δημιουργία H mask
    $x_m^i = H \odot x_i$
    $y_m^i = y_i$
  else
    Τυχαία επιλογή δειγμάτων $(x_m, y_m) \in T$
    if augtype == mixup **then**
      MixUp επαύξηση
      $x_m^i = \lambda x_i + (1 - \lambda) x_m^i$
      $y_m^i = \lambda y_i + (1 - \lambda) y_m^i$
    else ▷ CutMix επαύξηση
      Δημιουργία τυχαιάς μάσκας F
      $x_m^i = F \odot x_i + (1 - F) \odot x_m^i$
      $y_m^i = \lambda y_i + (1 - \lambda) y_m^i$
    end if
  end if
  $T_{\text{aug}} \leftarrow (x_m^i, y_m^i)$ ▷ Δημιουργία καινούριου Task με επαυξημένα δείγματα
end for
return augmented task(T_{aug})

Παρακάτω φαίνεται το pipeline του MetaMed που ακολουθήθηκε.



Σχήμα 4.19: *MetaMed pipeline*

4.4 PFEMED[3]

Το PFEMED[3] είναι ενα μοντέλο εμπνευσμένο απο το PFENET[4], που έχει ως σκοπό την επιλύση του προβλήματος του long tailed classification με την μέθοδο του few shot learning και του meta-learning που εξετάστηκε και απο το MetaMed[3]. Το συγκεκριμένο μοντέλο έχει ως σκοπο την εξαγωγή γενικων και ειδικών χαρακτηριστικών απο βιοιατρικές εικόνες. Ο τρόπος που πραγματοποιήθηκε η εξαγωγή αυτων των δεδομένων ήταν με χρήση μια δομής διπλού κωδικοποιητή, όπου ο ένας κωδικοποιητής αποτελείται απο ενα CNN μοντέλο το οποίο είχε εκπαιδευτεί στο ImageNet και ο δεύτερος κωδικοποιητής απο ενα CNN μοντέλο το οποίο εκπαιδεύεται στο dataset που περιέχει τα δεδομένα για τα οποία επιθυμούμε, να εξάγουμε συμπεράσματα. Επίσης εκτός απο αυτούς τους δυο encoder που εξάγουν πιο χρήσιμα χαρακτηριστικά χρησιμοποιήθηκε και ενας Variational Autoencoder (VAE), που αύξησε την ευρωστια των εξαγώμενων χαρακτηριστικών, που προκύπτουν απο την συνένωση των χαρακτηριστικών που εξάγονται απο τους δυο προαναφέροντες κωδικοποιητές. Τέλος χρησιμοποιήθηκε η μέθοδος του PT-MAP[5] ωστε να βπηθήσει στην αύξηση του accuracy,

4.4.1 PFENET[4]

Για τα support, query δεδομένα μπορούν, να εξαχθουν χαρακτηριστικά απο backbones και στην συνέχεια να τα επεξεργαστουν ώστε να εξάγουν παραμέτρους, που να είναι χρήσιμες στην ταξινόμηση[79, 80] υπολογίζοντας το cosine similarity[81, 82, 83] ή την συνέλιξη[84, 85, 86, 87, 88], για να εξάγουν την τελική πρόβλεψη. Η μέθοδος PFEMED για την δημιουργια σύνθετων χαρακτηριστικών που περιέχουν περισσότερες πληροφορίες και βοηθούν στην εξαγωγή πιο εύρωστων χαρακτηριστικών εμπνεύστηκε απο την τεχνική του PFENET που εφαρμόζει ενα σύνολο απο μετασχηματισμούς σε χαρακτηριστικά που εξάγει απο CNN μοντέλα. Αρχικά, το PFENET χρησιμοποιεί CNN μοντέλα, τα οποία ειχαν προεκπαιδευτεί πάνω στο Imagenet ως backbones με σκοπο την εξαγωγή χαρακτηριστικών απο το δικτύο πάνω στο σύνολο δεδομένων για το οποίο θελουμε να εξάγουμε προβλέψεις. Τα χαρακτηριστικά που εξάγονται απο το δίκτυο χωρίζονται σε high level και mid level τα πρώτα αφορούν δεδομένα τα οποία εξάγωνται απο τα τελευταία layers του CNN και τα δεύτερα απο μεσαία layer του CNN. Τα high level χαρακτηριστικά για τα support δεδομένα στην συνέχεια

περνάνε απο ενα binary masking, με σκοπο να αφαιρεθουν οι πληροφορίες του background και να κρατηθούν μόνο οι βασικές πληροφόρίες. Στην συνέχεια στα support δεδομένα που έχουν περάσει απο mask και στα query δεδομένα εφαρμόζεται μια συνάρτηση cosine similarity, με σκοπό να βρεθούν οι περιοχές που έχουν μεγάλη συσχετισή μεταξύ τους. Αυτό το σύνολο χαρακτηριστικών ονομάζεται Prior Mask και η διαδικασία που περιγράφηκε Prior Generation.

Η επόμενη μέθοδος που εφαρμόζεται για την ενίσχυση των χαρακτηριστικών είναι το Feature Enrichment Module. Σε αυτό εισάγονται χαρακτηριστικά medium level στο σύστημα query δεδομένα, τα οποία έχουν περάσει απο μια average pooling συνάρτηση. Επίσης εισάγονται και τα support δεδομένα, τα οποία έχουν περάσει απο ενα global pooling στο οποίο έχει χρησιμοποιηθεί και το Support Mask. Στην συνέχεια αυτά τα δεδομένα προσαρμόζονται σε διαφορετικές κλίμακες και συνενώνονται μεταξύ τους αλλά και με το Prior Mask. Έτσι προκύπτει ενας πίνακας χαρακτηριστικών για κάθε διαφορετική κλίμακα. Ολες οι κλίμακες εκτός απο την πρώτη χρησιμοποιούν χαρακτηριστικά απο τις προηγούμενες κλίμακες με τα οποία εννώνονται και προστίθενται με σκοπο την αποφυγή επανάληψης μη αναγκαίων πληροφοριών. Τέλος τα δεδομένα τροποποιούνται στην κατάλληλη κλίμακα και προστίθενται μεταξύ τους, με σκοπό να πάρουμε το καινούριο feature το οποίο θα προσφέρει περισσότερες πληροφορίες.

4.4.2 PT-MAP[5]

Εκτός απο το PFENET το PFEMED χρησιμοποιήσε και αυτό την μέθοδο του PT-MAP, η οποια δέχεται ως είσοδο ενα σύνολο απο χαρακτηριστικά που εξάγονται απο ενα μοντέλο και έχει ως σκοπό να τα κανονικοποιήσει σε μια Gaussian κατανομή με την χρήση της μεθόδου Power Transform και στην συνέχεια πραγματοποιεί την πρόβλεψη της κλάσης με τον υπολογισμό της maximum a posterior. Η συγκεκριμένη τεχνική αναπτύχθηκε λόγω της αυξανόμενης χρήσης μεθόδων transfer learning στην επίλυση προβλημάτων few shot learning, όπου χρησιμοποιούνται χαρακτηριστικά που εξήχθησαν απο μοντέλα, τα οποία είχαν εκπαιδευτεί σε μεγάλυτερα σύνολα δεδομένων. Το πρόβλημα με αυτές τις μέθοδους είναι οτι τα χαρακτηριστικά, που εξάγωνται θεωρείτε οτι ακολούθουν μια συγκεκριμένη κατανομή και αυτή η κατανομή ειναι παρόμοια με αυτή των καινούριων novel δεδομένων. Στην πραγματικότητα όμως οι κατανομές των χαρακτηριστικών είναι αρκετά πολύπλοκες. Έτσι η μέθοδος του PT-MAP προτείνει εναν τρόπο προεπεξεργασιας των χαρακτηριστικών, με σκοπό να τα προσαρμόσει σε μια κατάλληλη κατανομή και στην συνέχεια με μια μέθοδο υπολογισμού της maximum a posteriori προσπαθεί να αξιοποιήσει τις πληροφορίες, που παρέχει αυτή η κατανομή με σκοπο την πρόβλεψη πιο αξιόπιστων αποτελεσμάτων.

Όπως έχουμε αναφέρει και σε προηγούμενα κεφάλαια, το πρόβλήμα του few shot learning μπορεί να οριστεί απο δυο σύνολα δεδομένων, τα οποία δεν έχουν κοινά στοιχεια και κλάσεις μεταξύ τους και στόχος μας για την επίλυση αυτου του προβλήματος είναι η αποδοτική αξιοποιήση των δεδομένων στο training dataset, με σκοπό να εξαγούμε ικανοποιητικά αποτελέσματα στο novel dataset στο οποίο πραγματοποιείται περιορισμένη εκπαίδευση fine-tuning. Όπως περιγράψαμε και στο 2.5.1 χωρίζουμε τα δεδομένα σε support, query τα πρώτα είναι τα δεδομένα για τα οποία έχουμε ετικέτες και τα δεύτερα είναι αυτά πάνω στα

οποία θέλουμε να πραγματοποιήσουμε προβλέψεις. Εαν επίσης θεωρήσουμε w τις κλάσεις που επιλέγονται σε ενα task έχουμε ενα σύνολο απο w(s+q) δεδομένα. Επίσης θεωρούμε f_∂ το μοντέλο το οποίο χρησιμοποιείται ως backbone για την εξαγωγή χαρακτηριστικών, το μοντέλο πρέπει να περνάει απο μια RELU, ωστε να είναι ολοι οι παράμετροι των χαρακτηριστικών του θετικοί.

Αν θεωρήσουμε ως $\upsilon = f_\phi(x), x \in D_{novel}$, το οποίο συμβολίζει τα δεδομένα που εξήχθησαν απο το μοντέλο που επιλέχθηκε ως backbone. Η power transform που εφαρμόζεται περιγράφεται απο την παρακάτω εξίσωση.

$$f(\upsilon) = \begin{cases} \frac{(\upsilon+\epsilon)^\beta}{\|\upsilon+\epsilon\|_2^\beta} & \text{if } \beta \neq 0 \\ \frac{\log(\|\upsilon+\epsilon\|)}{\log(\|\upsilon+\epsilon\|)} & \text{if } \beta = 0 \end{cases} \tag{4.18}$$

με $\epsilon = 1\epsilon - 6$ έτσι ώστε $\upsilon + \epsilon$ να είναι θεττικό, το αποτελεί μια υπερπαράμετρο που επηρεάζει την κατανομή.

Αν θεωρήσουμε οτι τα δεδομένα που προεπεξεργαστήκαμε ακολουθούν μια Gaussian κατανομή για κάθε κλάση. Ετσι ειναι σημαντικό να οριστεί ενα καλά τοποθετημενο κέντρο με σκοπό να πραγματοποιηθούν καλές προβλέψεις. Ο τρόπος που πραγματοποιούνται οι προβλέψεις για την κατηγοριοποίηση των δειγμάτων είναι με την χρήση ενος αλγορίθμου Expectation-Maximization που θα υπολογίζει επαναληπτικά την maximum a posterior πιθανότητα των κέντρων των κλάσεων για κάθε δείγμα.

Η εκτίμηση αυτών των κέντρων μέσω του MAP μοιάζει με την ελαχιστοποίηση της Wasserstein απόστασης. Έτσι δημιουργείται μια μέθοδος, που υπολόγιζει αυτήν την απόσταση με την χρήση του Sinkhorn αλγόριθμου. Αυτή η μέθοδος βοηθάει στην εκτίμηση της βέλτιστης μεταφοράς απο την αρχική κατανομή των χαρακτηριστικών σε μια που θα αντιστοιχεί στην κατάλληλη Gaussian κατανομή για την επιλογή τυχαίων δειγμάτων.

Για την περιγραφή των εξισώσεων που χρησιμοποιούνται για την εκτίμηση των κέντρων των κατανομών για κάθε κλάση ορίζουμαι ως f_S, f_Q το σύνολο των χαρακτηριστικών, που απευθύνονται στα δεδομένα, τα οποία γνωρίζουμε σε πια κλάση ανήκουν και τα δεδομένα για τα οποία δεν ξέρουμε σε πια κλάση ανήκουν και θέλουμε να πραγματοποιήσουμε προβλέψεις πάνω σε αυτά. Για αυτά τα χαρακτηριστικά ορίζουμε l(f) την αντίστοιχη κλάση στην οποία θα ανήκουν. Επίσης συμβολίζουμε τα δεδομένα για τα οποία δεν έχουμε labels ως $f_Q = (f_i)_i, 0 < i \leq wq$ και ορίζουμε τα κέντρα ως $c_j, 0 < j \leq w$ με το j να συμβολίζει την καθε κλάση.

Για τον υπολογισμό των κέντρων ακολουθούμε τον παρακάτω αλγόριθμο

Αλγοριθμος 4.4: *PT-MAP[5]*

Παράμετροι: $w, s, q, \lambda, a, nsteps$
Αρχικοποιήση: $c_j = \frac{1}{s}\sum_{f \in fs, l(f)=j} f$
for $i = 1$ το $nsteps$ **do**

Υπολόγισε $L_{ij} = (f_i - c_j)^2, \forall i, j$

Υπολόγισε $M^* = \text{Sinkhorn}(L, p = 1_{wq}, q = q_{1w}, \lambda)$

Υπολόγισε $\mu_j = g(M^*, j)$

Ανανέωσε $c_j \leftarrow c_j + a(\mu_j - c_j)$
return $\hat{l}(f_i) = \arg\max_j(M^*[i, j])$

Ο πίνακας που χρησιμοποιείται για mapping, με σκοπο να ελαχιστοποιήσει την Wasserstein distance συμβολίζεται ως M^* που γράφεται ως

$$M^* = \text{Sinkhorn}(L, p, q, \lambda) = \arg\min_{M \in U(p,q)} \sum_{ij} M_{ij} L_{ij} + \lambda H(M) \tag{4.19}$$

με $U(p, q) \in \mathbb{R}^{wqxw}$, το οποίο αποτελείτε απο θεττικά στοιχεία με τις σειρές να αθροίζουν σε p και οι στήλες να αθροίζουν σε q. Το $U(p, q)$ γράφεται ως

$$U(p, q) = \{M \in \mathbb{R}^{wq \times w} \mid M1_w = p, M^T 1_{wq} = q\}$$

Το p αναφέρεται στην κατανομή του μεγέθους, που χρησιμοποιεί κάθε δείγμα για ταξινόμηση και το q αναφέρεται στον αριθμό των δειγμάτων, που είναι ταξινομημένα σε κάθε κλάση με αυτόν τον τρόπο περιέχονται ολοι οι τρόποι που μπορούν να ταξινομηθούνουν τα δεδομένα στις κλάσεις

Η συνάρτηση που περιγράφει το loss $L \in \mathbb{R}^{wqxw}$ 4.19 αποτελείται από την ευκλείδεια απόσταση μεταξύ των δεδομένων οι κλάσεις των οποίων είναι άγνωστες και των κέντρων των κλάσεων που έχουν υπολογιστεί, έτσι το L_{ij} συμβολίζει την ευκλείδεια απόσταση μεταξύ ενος δείγματος i και ενος κέντρου μιας κλάσης j. Για την ταξινόμηση των δειγμάτων θεωρούμε ότι ένα δείγμα μπορεί να κατηγοριοποιηθεί σε παραπάνω από μια κλάσεις. Ο δεύτερος όρος του σφάλματος ισοδυναμεί με την εντροπία του πίνακα mapping και περιγράφεται από την παρακάτω εξίσωση $H(M) = -\sum_{ij} M_{ij} \log(M_{ij})$ το οποίο κανονικοποιείται απο την υπερ-παράμετρο λ

Όπως φαίνεται και απο τον αλγόριθμο 4.4 στο πρώτο βήμα αρχικοποιούνται τα κέντρα των κατανομών των κλάσεων. Στην συνέχεια σε κάθε επανάληψη του αλγοριθμου τα κέντρα επανεκτιμώνται, αφου σε κάθε iteration επαναυπολογίζεται ο πίνακας mapping M^* πάνω στα δείγματα τα οποία δεν έχουν labels χρησιμοποιώντας το sinkhorn mapping. Τα δεδομένα τα οποία έχουν ετικέτες χρησιμοποιούνται και αυτά στον υπολογισμό των νέων κέντρων τα οποία δίνονται από τις παρακάτω εξισώσεις.

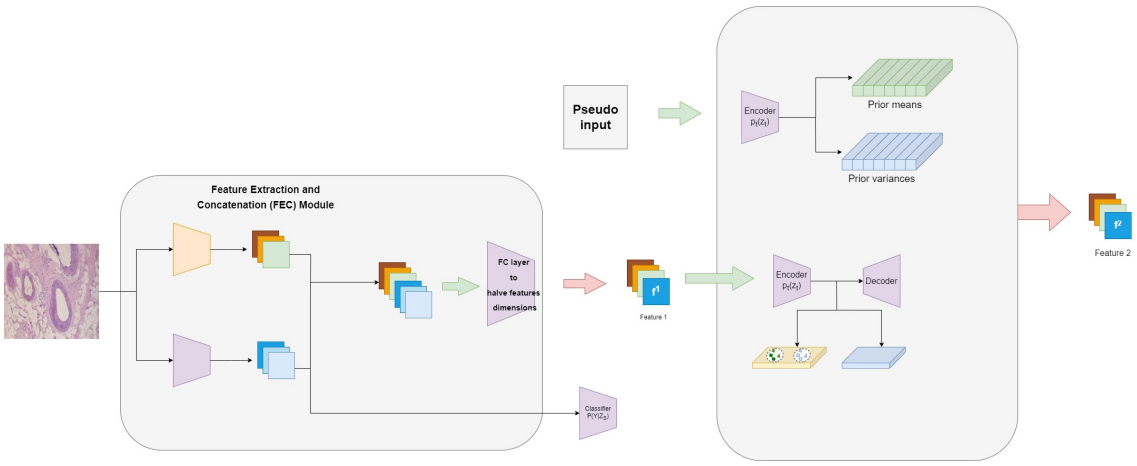
$$\mu_j = g(M^*, j) = \frac{\sum_{i=1}^{wq} M^*_{ij} f_i + \sum_{f \in fS, (f)=j} f}{s + \sum_{i=1}^{wq} M^*_{ij}} \tag{4.20}$$

Με τον τρόπο που προτείνεται για την ανανέωση των βαρών μπορεί σε αρχικές επαναλήψεις τα κέντρα να οδηγηθούν σε αποφάσεις, που ειναι μακρια απο τα βέλτιστα κέντρα, έτσι για την ανανέωση των βαρών χρησιμοποιείται μια παράμετρος που περιορίζει το πόσο μεγάλα είναι τα βήματα της ανανέωσης των βαρών. Αυτη η παράμετρος συμβολίζεται με a και μεταβάλλεται με την αύξηση του αριθμου των επαναλήψεων. Η εξίσωση της ανανέωσης των κέντρων δίνεται από την παρακάτω σχέση $c_j \leftarrow c_j + a(\mu_j - c_j)$

Μετα απο συγκεκριμένο αριθμό επαναλήψεων οι γραμμές του πίνακα $M*$ αντιπροσωπεύουν τις πιθανότητες του δείγματος να ανήκει σε κάθε κλάση. Απο αυτές τις τιμές επιλέγεται η μεγαλύτερη ως η τελική απόφαση ταξινόμησης του δείγματος στην αντίστοιχη κλάση.

4.4.3 Feature Extraction and Enhancement(FEE)[3]

Η τεχνική που χρησιμοποιείται για την ενίσχυση των χαρακτηριστικών στο PFEMED αν και βασίζεται στην μέθοδο που προτάθηκε απο το PFENET είναι διαφορετική κύριως στον τρόπο με τον οποίο συνδυάζει τα χαρακτηριστικά των support,query δειγμάτων. Παρακάτω φαίνεται το framework που ακολουθείται για την ενίσχυση των χαρακτηριστικών.

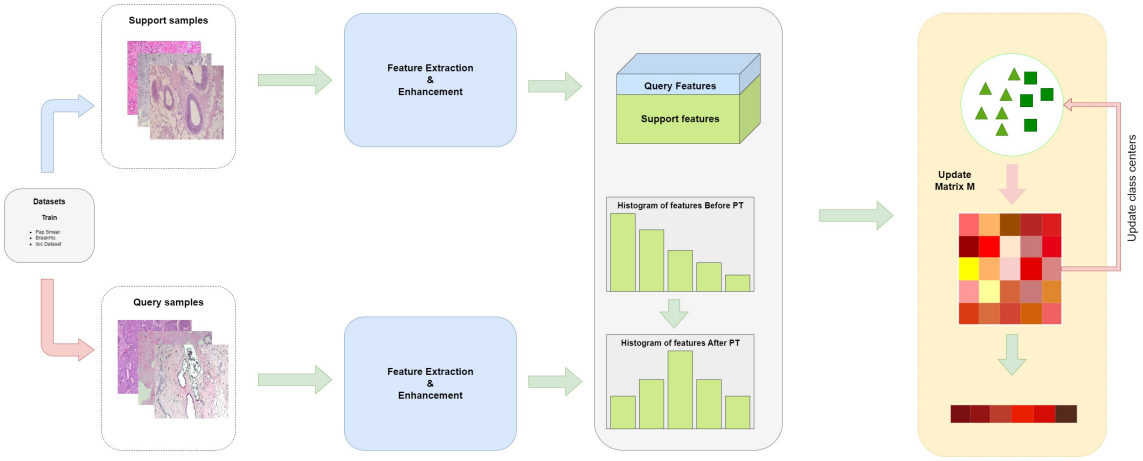


Σχήμα 4.20: *PFEMED feature enrichment*

Απο την (4.20) φαίνεται οτι για την εξαγωγή χαρακτηριστικών χρησιμοποιούνται δυο δίκτυα CNN ως encoder τα οποία έχουν την ίδια αρχιτεκτονική και στο μόνο πράγμα που διαφέρουν είναι οι τιμές των παραμέτρων τους. Το πρώτο δίκτυο είναι εκπαιδευμένο στο mini imagenet το οποίο αποτελεί ένα υποσύνολο του ImageNet και αυτό το δίκτυο χρησιμοποιείται για την εξαγωγη γενικών χαρακτηριστικών πάνω στο σύνολο δεδομένων για το οποίο θέλουμε να εξάγουμε συμπεράσματα. Εκτος απο αυτό το μοντέλο χρησιμοποιείται και ενα επιπλέον CNN δίκτυο στην μορφή encoder. Αυτό το δίκτυο εκπαιδεύεται πάνω στο train dataset και εξάγει χαρακτηριστικά τα οποία είναι πιο συγκεκριμένα και πιο κοντά στην κατανομή των δεδομένων πάνω στην οποία θέλουμε να πραγματοποιήσουμε προβλέψεις novel Dataset. Τα χαρακτηριστικά που εξάγονται απο αυτό το δίκτυο είναι πιο συγκεκριμένα και δίνουν πιο λεπτομερείς πληροφορίες για τα δεδομένα, τα οποία θέλουμε να προβλέψουμε. Αυτα τα

χαρακτηριστικά τα ονομάζουμε specific features, ενώ αυτα που εξήχθησαν απο το δίκτυο που εκπαιδεύτηκε στο mini imagenet ονομάζονται general features, στην συνέχεια αυτά τα δύο σύνολα χαρακτηριστικών specific, general που εξήχθησαν συνενώνονται. Αυτή η διαδικασία πραγματοποιείται ξεχωριστα για τα support και query δεδομένα.

Αφου τα specific, general χαρακτηριστικά ενωθούν σε έναν μεγαλύτερο πίνακα, στην συνέχεια περνάνε απο ένα fully connected layer, το οποίο μειώνει τις διαστάσεις του πίνακα χαρακτηριστικών. Αφου μειωθούν οι διαστάσεις του πίνακα χαρακτηριστικών στην συνέχεια τροφοδοτούνται ως είσοδοι σε ενα Variational Autoencoder (VAE). Ο VAE είναι ένα μοντέλο το οποίο αποτελείται από δομές encoder, decoder, οι οποίες με την σειράς τους αποτελούνται απο layers Fully connected δικτύων. Ο ρόλος του VAE είναι να τροφοδοτηθεί με τα δεδομένα και να τα απεικονίσει σε έναν χώρο κατάστασης ως κανονικές κατανομές με ένα συγκεκριμένο κέντρο και διασπορά. Αυτή η απεικόνιση προκύπτει απο την έξοδο του encoder και έχει ως σκοπό την δημιουργια κανονικοποιημένων δειγμάτων, ωστε να μπορει να επιλεχθεί ενα τυχαίο δείγμα και να ανακατασκευαστει μια ενισχυμένη αναπαράσταση απο τον decoder. Ο VAE πετυχαίνει να απεικονίσει αυτά τα δείγματα σε μικρότερους χώρους κατάστασης με τον υπολογισμό της posterior πιθανοτητας. Τα χαρακτηριστικά που εξάγονται απο αυτόν τον autoencoder συμβολίζονται με f^2 και στην συνέχεια αυτά τα χαρακτηριστικά εισαγονται στο PT-MAP, που εξηγήσαμε στο 4.4.2. Εκτός απο τον VAE που υπολογίζει τις posterior πιθανότητες των δειγμάτων χρησιμοποιείται ένα επιπλέον δίκτυο για την εκτίμηση των prior πιθανοτήτων. Αυτό το δίκτυο δέχεται ως είσοδο pseudo inputs, τα οποία είναι τένσορες που αποτελούνται μόνο απο την τιμή 1 και απο την υπόθεση οτι κάθε κλάση των βιοϊατρικών συνόλων δεδομένων έχει ένα κέντρο τον χώρο κατάσταση, που απεικονίζεται απο τον VAE κοντα στο οποίο βρίσκονται τα δείγματα που ανήκουν στην ίδια κλάση. Με αυτήν την υπόθεση μπορούμε να συμπεράνουμε οτι μετα απο συγκεκριμένο αριθμό βελτιστοποιήσεων και παρατηρήσεων οι προβλέψεις των prior θα συγκλίνουν στα βέλτιστα κέντρα των κλάσεων.



Σχήμα 4.21: *PFEMED pipeline*

Για την εκτίμηση του δικτύου χρησιμοποιείται gradient descent optimizer και η ανανέωση των βαρών γίνεται με ενα σύνθετο loss. Για την περιγραφή των εξισώσεων ορίζουμε τις κατανομές των posterior και των prior για μια i κλάση που συμβολίζονται ως $p_t(z_{(i)})$, $p_{\hat{\eta}}(z_{(i)})$ και περιγράφονται απο τις παρακάτω εξισώσεις.

$$p_t(z(i)) = q_\phi(z|x(i)) = \begin{cases} \mathcal{N}(z_{(i)}|u_{(i)}, \sigma_{(i)}), & z_{(i)} \in D_{\text{base}} \\ \mathcal{N}(z_{(j)}|v_{(j)}, \lambda_{(j)}), & z_{(j)} \in D_{\text{novel}} \end{cases}$$

(4.21)

$$p_\lambda(z_{(i)}) = q_\phi(z|d) = \begin{bmatrix} \mathcal{N}(z^*_{(1)}|u^*_{(1)}, \sigma^*_{(1)}) \\ \vdots \\ \mathcal{N}(z^*_{(l)}|u^*_{(l)}, \sigma^*_{(l)}) \\ \mathcal{N}(z^*_{(1)}|v^*_{(1)}, \lambda^*_{(1)}) \\ \vdots \\ \mathcal{N}(z^*_{(k)}|v^*_{(k)}, \lambda^*_{(k)}) \end{bmatrix} \tag{4.22}$$

Οπου d είναι τα pseudo inputs και $x_{(i)}$ είναι το i δείγμα, επίσης $u^*_{(l)}$, $\sigma^*_{(l)}$ και $v^*_{(1)}$, $\lambda^*_{(1)}$ είναι οι κατανομές των posterior στις γνωστές και άγνωστες κλάσεις i, j.

Το συνολικό loss αποτελείτε απο το L_class που αποτελεί το cross entropy loss

$$L_{\text{class}} = -\frac{1}{N}\sum_i \left(y_i \log(y_i) + (1 - y_i)\log(1 - y_i)\right) m_i$$

$$\text{s.t.} m_i = \begin{cases} 1, & \text{if } y_i \in D_{\text{base}} \\ 0, & \text{if } y_i \in D_{\text{novel}} \end{cases}$$

Επίσης χρησιμοποιείται ενα loss, το οποίο υπολογίζει πόσο ομοια είναι τα γενικά με τα ειδικά χαρακτηριστικά και προσπαθει να τα απομακρύνει μεταξύ τους, με σκοπό να μην υπαρχουν πλεονάζουσες πληροφορίες το οποίο συμβολίζεται ως L_{repul}.

$$L_{\text{repul}} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(1 - \cos(z_i^g, z_j^s))$$

Επιπλέον χρησιμοποιείται η μέση τετραγωνική απόσταση για τον υπολογισμό των διαφορων των καινούριων ενισχυμένων δειγμάτων με τα παλια που χρησιμοποιήθηκαν ως είσοδοι και συμβολίζεται με L_{recon}

$$L_{\text{recon}} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(f_1(i) - f_2(j))^2$$

Ακόμα υπολογίζεται το σφάλμα της κατανομής δεδομένων που ανήκουν σε γνωστες κλάσεις. Αυτό το σφάλμα συμβολίζεται ως $L_{\text{inner\_known\_post}}$

$$L_{\text{inner\_known\_post}} = \text{DKL}(p_t(x)||q(x)) \tag{4.23}$$
$$= E(p_t)[\log(p_t) - \log(q)] \tag{4.24}$$
$$= \frac{1}{2}\log\left|\frac{q}{p_t}\right| - \tag{4.25}$$
$$\frac{1}{2}E(p_t)\left((x-\mu_p)^T\Sigma_p^{-1}(x-\mu_p)\right) + \tag{4.26}$$
$$\frac{1}{2}E(p_t)\left((x-\mu_q)^T\Sigma_q^{-1}(x-\mu_q)\right) \tag{4.27}$$
$$= \frac{1}{2}\left(-\log|\Sigma p_t| - l + \mu_p^T p_t \mu_p + \text{Tr}\Sigma(p_t)\right) \tag{4.28}$$

Παρόμοια υπολογίζεται το loss μεταξυ των γνωστών και των άγνωστων χαρακτηριστικών που συμβολίζεται με $L_{\text{inner\_prior}}$

$$L_{\text{inner\_prior}} = \frac{1}{2}\left(-\log|p_{\hat{\jmath}}| - n + \mu_p^T p_{\hat{\jmath}}\mu_p + \text{Tr}(p_{\hat{\jmath}})\right) \tag{4.29}$$

Επίσης υπολογίζεται ενα σφάλμα μεταξυ των prior και των posterior πιθανοτήτων για τις γνωστές κλάσεις

$$L_{\text{intro\_known}} = \frac{1}{2}\log\left|\frac{\Sigma p_{\hat{\jmath}}}{\Sigma p_t}\right| - \frac{l}{2} + \frac{1}{2}(\mu_{pt} - \mu_{p\hat{\jmath}})^T\Sigma_q^{-1}(\mu_{pt} - \mu_{p\hat{\jmath}}) + \frac{1}{2}\text{Tr}(\Sigma_{p\hat{\jmath}}^{-1}\Sigma_{pt}) \tag{4.30}$$

Επίσης χρησιμοποιείται ενα triplet loss, με σκοπό να υπάρξει μεγαλύτερος διαχωρισμός μεταξύ των κλάσεων

$$L_{\text{triplet}} = \sum_{i=1}^{k}\sum_{j=1}^{l+k}\left(||\mu_a - \mu_p||_2^2 - ||\mu_a - \mu_n||_2^2 + a_w\right) \tag{4.31}$$

Συνδυάζοντας αυτά τα σφάλματα μεταξυ τους μπορούμε να πάρουμε το συνολικό σφάλμα για την κατανομή των χαρακτηριστικών $L_{distri\_1} = L_{inner\_known\_post} + L_{inner\_prior} + L_{intro\_known} + L_{intro\_unknown} + L_{triplet}$

Το συνολικό σφάλμα στην πρώτη φάση εκπαίδευσης υπολογίζεται ως $L_{total_1} = L_{class} + L_{recon} + L_{repul} + L_{distri\_1}$

Στην δεύτερη φάση της εκπαίδευσης παύει να χρησιμοποιείται το $L_{intro_unknown}, L_{triplet}$, διοτι μπορεί να δημιουργηθούν προβλήματα ασταθειας στην αντιστοιχια των posterior και prior. Ετσι το loss δίνεται ως

$$L_{distri\_1} = L_{inner\_known\_post} + L_{inner\_prior} + L_{intro\_known}$$

και

$$L_{total\_1} = L_{class} + L_{recon} + L_{repul} + L_{distri\_2}$$

Μετα το τέλος αυτής της εκπαίδευσης, τα χαρακτηριστικά εισάγωνται στην PT-MAP μεθοδο,

που αναλύθηκε στο 4.4.2

Ο αλγόριθμος που χρησιμοποιήθηκε για την εκπαίδευση φαίνεται παρακάτω

Αλγοριθμος 4.5: *Αρχικοποίηση εκπαίδευσης*

μ_i ($i \in [1, l+k]$): κέντρο της i κλάσης της posterior
μ_j^* ($j \in [1, l]$): κέντρο της j κλάσης της prior
υ_m ($m \in [1, k]$): κέντρο της m αγνωστης κλάσης της prior
Αρχικοποίηση των επαναλήψεων $n_e = 90$
Πρόλεψη του μ, μ^*, και του υ απο το προτεινόμενο μοντέλο
$[C(1), ..., C(k)]$ = K-means(υ)
for $m = 1$ to k **do**
    Αρχικοποίηση των κέντρων μ_{l+m} σε C_m
end for
for $n = 0$ το n_e **do**
    Πρόβλεψη του μ, μ^*, και υ απο το προτεινόμενο μοντέλο
    $[C1, ..., Ck]$ = K-means(υ)
    for $s = 1$ to k **do**
        $C_s = \arg\min(DKL(C||\mu_s^*))$
    end for
    Υπολογισμός του loss
    Ενημέρωση των παραμέτρων του μοντέλου
end for
    return νέες παραμέτρους του μοντέλου =0

4.5 Few Ture[6]

Τα προβλήματα του few shot learning απαιτούν την εκπαίδευση του δικτύου σε ένα μικρό συνολο απο δεδομένα, για αυτό τον λόγο προτιμούνται συνήθως μοντέλα, τα οποία έχουν μικρό αριθμό παραμέτρων με σκοπο να μην οδηγούνται σε overfitting κατα την διάρκεια εκπαίδευσης. Για αυτόν τον λόγω μοντέλα όπως οι transformers τείνουν να μην χρησιμοποιούνται στην επίλυση τέτοιων προβλημάτων. Το fewTure προτείνει έναν τρόπο εκπαίδευσης μοντέλων transformers, ο οποίος παράγει αποτελέσματα τα οποία μπορούν να συγκριθούν με τις state of the art μεθόδους σε dataset όπως το mini-imagenet, tierdimagenet κ.λ.π.

Πιο συγκεκριμένα το μοντέλο ακολουθεί ένα τρόπο εκπαίδευσης κατα τον οποίο χωρίζει την εικόνα εισόδου σε μικρότερες εικόνες και τις κωδικοποιεί με την βοήθεια vit-transformers, τις οποίες ονομάζει patches κατι το οποίο βοηθάει στην δημιουργία σημασιολογικών συνδέσεων μεταξύ τον τοπικών περιοχών πάνω στις εικόνες χωρίς να λαμβάνει υπόψη τις αντιστοιχες κλάσεις. Απο αυτές τις μικρότερες αναπαραστάσεις εικόνων που έχουν κωδικοποιηθεί επιλέγονται αυτές που παρέχουν τις περισσότερες πληροφορίες και χρησιμοποιούνται ως συνάρτηση του support set μέσω online optimization. Αυτή η τεχνική βασίζεται στην ανάπτυξη unsupervised μεθόδων εκπαίδευσης των δικτύων με χρήση εικόνων που τους εφαρμόζεται μια επικάλυψη. Αυτή η μοντελοποίηση έχει ως σκοπό να προσφέρει μια λύση στο πρόβλημα μη καλώς ορισμένων ετικετών μαθαίνοντας πιο γενικές στατιστικές δομές των δεδομένων αποφεύγοντας παράλληλα την ζημιογόνα ταύτιση χαρακτηριστικών που οδηγεί σε supervision collapse[89].

Τα περισσότερα σύνολα δεδομένων που αφορούν την ταξινόμηση εικόνων δίνουν ετικέτες στις εικόνες με βάση το κύριο χαρακτηριστικό τους, έτσι όταν το μοντέλο εκπαιδεύεται με μεθόδους gradient descent πάνω σε τέτοια δεδομένα μαθαίνει να επικεντρώνεται στην κύρια πληροφορία, που απεικονίζεται σε αυτά και αγνοεί πληροφορίες, οι οποίες φαίνονται ασήμαντες στην ταξινόμηση. Κάτι τέτοιο μπορεί να μην αποτελεί μεγάλο πρόβλημα σε σύνολα δεδομένων, τα οποία έχουν πολλά δείγματα, αλλά στις περιπτώσεις που εφαρμόζονται μέθοδοι όπως το few shot learning, όπου η εκπαίδευση βασίζεται σε λίγα δεδομένα και το σύνολο των κλάσεων στα δεδομένα εκπαίδευσης διαφέρει από τα δεδομένα πρόβλεψης αυτή η παράλειψη δεδομένων, τα οποία δεν θεωρούνται σημαντικά μπορεί να οδηγήσει σε δίκτυα, τα οποία δεν γενικεύουν ικανοποιητικά. Επίσης ένα άλλο σύνηθες πρόβλημα που προκύπτει σε μεθόδους όπως το few shot learning είναι οτι το μοντέλο δίνει μεγάλη σημασία σε χαρακτηριστικά και πρότυπα που προέκυψαν κατα την διάρκεια της εκπαίδευσης, τα οποία δεν συναντώνται στα δεδομένα πάνω στα οποία θέλουμε να πραγματοποιηθούν οι προβλέψεις, αυτό μπορεί να οδηγήσει σε χαμηλά αποτελέσματα κατα την διάρκεια της πρόβλεψης και ονομάζεται supervision collapse.

Για την αντιμετώπιση αυτου του προβλήματος εχουν αναπτυχθεί κάποιες τεχνικές. Οι περισσότερες από αυτές τις μεθόδους βασίζονται σε μεθόδους που συνδυαζουν σημασιολογικά τα δεδομένα των support, query ή μπορεί να εξάγουν χάρτες χαρακτηριστικών που απεικονίζουν την σημασιολογική σύνδεση μεταξύ των δεδομένων δίνοντας μεγαλύτερες τιμές σε περιοχές, όπου υπάρχει μεγαλύτερη συσχέτιση των χαρακτηριστικών. Άλλες μέθοδοι προτείνουν τεχνικές cross-attention μεταξύ των support και query χαρακτηριστικών που επικεντρώνεται στις περιοχές που είναι σημαντικές για ταξινόμηση. Τέτοιες μέθοδοι αν και δίνουν καλά αποτελέσματα επικεντρώνονται στις περιοχές, οι οποίες είναι σημαντικές για την ταξινόμηση στην αντίστοιχη κλάση, αρα οι πληροφορίες που παρέχουν σχετίζεται μόνο με την συγκεκριμένη κλάση και δεν παρέχουν πληροφορίες σχετικά με κοινά χαρακτηριστικά που μπορεί να συναντώνται μεταξύ δυο δειγματων που ανήκουν σε διαφορετικές κλάσεις.

Μια μέθοδος λοιπόν που μπορεί να παρεχει χαρακτηριστικά, που περιέχουν περισσότερες σημασιολογικές πληροφορίες για τα δεδομένα και να βρίσκει τις ομοιότητες αυτών με δεδομένα, που ανήκουν στην ίδια κλάση, αλλά και σε διαφορετικές μπορεί να φανεί πολύ χρήσιμη στην ανάπτυξη γενικότερων μοντέλων, που θα επιλύουν το πρόβλημα του few shot learning.

Με αφορμή τις παραπάνω παρατηρήσεις προτάθηκε μια μέθοδος, που χωρίζει την εικόνα σε μικρότερα τμήματα, τα οποία ονομάζονται patches και αποτελούν τοπικές περιοχές η οποίες έχουν μεγαλύτερη πιθανότητα να περιέχουν ένα χρήσιμο χαρακτηριστικό. Τα μοντέλα που επιλέγονται να εκπαιδευτούν πάνω σε αυτά τα δεδομένα είναι vision transformers και η τεχνική εκπαίδευσης που επιλέγεται είναι το self-supervised learning, λόγω της έλλειψης καλώς ορισμένων ετικετών. Επίσης χρησιμοποιείται επικάλυψη σε περιοχές των εικόνων, με σκοπό την εκπαίδευση πιο γενικών μοντέλων. Η ταξινόμηση βασίζεται στην εύρεση ομοιοτήτων μεταξυ των patches, που ορίσαμε για τα support δεδομένα. Για την εύρεση αυτών των νοηματικών ομοιοτήτων μεταξύ των patches δημιουργείται ενας χαρτης, που αποτελείται απο τις prior πιθανότητες. Ο τρόπος με τον οποίο δημιουργείται αυτός ο χάρτης που δείχνει την ομοιότητα των τοπικών περιοχών των support δεδομένων είναι με την χρήση παραμέτρων που δηλώνουν την συνεισφορα του κάθε token στην σωστή ταξινόμηση των δειγμάτων των

υπολοίπων support δεδομένων, κατι το οποίο ενισχύει τις τιμές που αντιπροσωπευουν ισχυρές ομοιότητες μεταξύ των κλάσεων και αποδυναμώνει τις περιοχές που μπορεί να υπάρχουν ομοιότητες μεταξύ δεδομένων διαφορετικών κλάσεων. Στην συνέχεια αυτή η παράμετροι σημαντικότητας που εξάγονται χρησιμοποιούνται ως βάση για την εξαγωγή για την ταξινόμηση των query δεδομένων. Σε αντίθεση με προηγούμενες μεθόδους στον συγκεκριμένο τρόπο εκπαίδευσης τα query δεδομένα δεν συμμετέχουν καθόλου στην διαδικασία εκπαίδευσης του μοντέλου και χρησιμοποιούνται αποκλειστικά στο βήμα του validation.

Για τον ορισμό του προβλήματος του few shot learning θα χρησιμοποιήσουμε τους συμβολισμούς, που είχαμε χρησιμοποιήσει και στις προηγούμενες μεθόδους που εξηγήσαμε. Για τα συνολά δεδομένα ορίζουμε τα D_{train}, D_{test}, τα οποία δεν έχουν κοινές κλάσεις. Ενώ για το episodic training ορίζουμε τα support sets ως $X_s = \{(x^s_{nk}, y^s_{nk}) \,|\, n = 1, \ldots, N; k = 1, \ldots, K; y^s_{nk} \in C_{\text{train}}\}$ με x^{nk}_s να είναι το $k-th$ δείγμα της κλάσης n με ετικέτα y^s_{nk} και τα query δεδομένα γράφονται $X_q = \{(x^n_q, y^n_q) \,|\, n = 1, \ldots, N$, όπου x^n_q είναι ενα query δείγμα της κλάσης n με ετικέτα y^n_q.

Απο το σχεδιάγραμμα 4.22 φαίνεται ότι η εικόνα πριν τροφοδοτηθεί στον transformer περνάει από ένα φίλτρο pixmix, το οποίο εφαρμόζει τυχαία τον μετασχηματισμό pixmix, που αναλύθηκε στο 4.2.2. Στην συνέχεια η εικόνα χωρίζεται σε patches, τα οποία εισάγονται στο σύστημα του transformer. Πιο συγκεκριμένα τα patches χωρίζονται σε αυτά του support set, τα οποία συμβολίζονται με p_s και αυτά του query set που συμβολίζονται με p_q, τα οποία εισάγονται στον transformer. Η έξοδος του transformer επιστρέφει τα embeddings των περιοχών των εικόνων, που εισήχθησαν και συμβολίζονται ως z_s, z_q για τα support, query αντίστοιχα. Για την ταξινόμηση των δεδομένων χρησιμοποιείται ο χάρτης που περιέχει τους prior υπολογισμούς και εκφράζει την νοηματική ομοιότητα των περιοχών των εικόνων που έχουν κωδικοποιηθεί στα query δεδομένα για τα οποία πραγματοποιείται η εκτίμηση, αλλά και για ολα τα support δείγματα. Αυτός ο χάρτης ομοιότητας των prior αναπαριστά τις συνδέσεις μεταξύ των δειγμάτων χωρίς να λαμβάνει υπόψη τις κλάσεις πράγμα, το οποίο μπορεί να επηρεάζει αρνητικά την ταξινόμηση. Για αυτόν το λόγο χρησιμοποιείται και ένας χάρτης, ο οποίος περιέχει τιμές που δείχνουν την συνεισφορά κάθε embedding στην σωστή ταξινόμηση.

4.5.1 Vision Transformers

Με την ανάπτυξη των transforms και την επιτυχία τους στην επίλυση προβλημάτων επεξεργασίας φωνής και φυσικής γλώσσας αρχισε να ερευνάτε η εφαρμογή τους και σε άλλους τομείς, όπως αυτος της επεξεργασίας εικόνας. Αντίστοιχα με τους transformers που δέχονται ως είσοδο μια ακολουθία από λέξεις, ετσι και στους ViT οι εικόνες χωρίζονται σε μικρότερα τμήματα(patches) και τοποθετουνται σειριακά, με σκοπό να δημιουργήσουν ένα διάνυσμα απο τα patches, το οποίο στην συνέχεια τροφοδοτείται στον Transformer. Ο transformer αποτελείται από μια σειρά από self-attention layers, τα οποία έχουν ως σκοπό να μάθουν συσχετίσεις μεταξύ μη γειτονικών patches. Οι έξοδοι των ViT αποτελούνται από διανύσματα embeddings για κάθε patch. Με την χρήση του μοντέλου των ViT μπορούν να διερευνηθούν συσχετίσεις των χαρακτηριστικών των εικόνων, οι οποίες δεν εξαρτώνται από την θέση τους πάνω στην εικόνα. Το βασικό μειονέκτημα των ViT είναι ότι έχουν πολλές παραμέτρους κατι

το οποίο κάνει την εκπαίδευση πιο χρονοβόρα.

4.5.2 Masked Image Modeling

Στο Fewture σύστημα χρησιμοποιήθηκε η τεχνική του Masked Image Modeling, η οποία δημιουργεί tokens με νοηματικές πληροφορίες, τα οποία στην συνέχεια χρησιμοποιούνται για την πρόβλεψη ενός patch μιας εικόνας, το οποίο έχει περάσει από κάποιο masking. Αυτή η ιδέα προήλθε απο το Masked Language Model, όπου στην αρχή κρύβει ένα τυχαίο κομματι απο την ακολουθία των tokens, το οποίο στην συνέχεια καλείται να ανακατασκευάσει απο τα υπόλοιπα tokens. Για την εκπαίδευση πραγματοποιήθηκε χρήση του Masked Image Modelling[90, 7, 91, 92, 93, 94, 95] καθώς τεχνικές αυτό-εκπαιδευσης έχουν δείξει, ότι μπορούν να μάθουν πιο γενικεύσιμα χαρακτηριστικά ακόμα και σε μικρότερου μεγέθους συνόλων δεδομένων[6].

Πιο συγκεκριμένα κάθε ακολουθία από tokens μιας εικόνας συμβολίζεται ως $x = x_i^N{}_{i=1}$ από αυτήν την ακολουθία tokens επιλέγεται τυχαία ένα δείγμα $m \in 0, 1^N$ στο οποίο εφαρμόζεται μια σειρά από augmentations, με σκοπό την δημιουργία corrupted αναπαραστάσεων αυτού του δείγματος που μπορεί να συμβολιστεί ως $\hat{x}, \{\hat{x}_i \,|\, (1 - m_i)x_i + m_i e[\text{MASK}]\}_{i=1}^N$. Σκοπός του μοντέλου εκπαίδευσης είναι να ανακτήσει τα masked tokens απο την εικόνα που της έχουν εφαρμοστεί μετασχηματισμοί, που έχουν ως σκοπό να την αλλοιώσουν.

4.5.3 Self-Distillation

Αυτή η μέθοδος ακολουθεί μια λογική εκπαίδευσης μαθητη-δασκαλου, όπου το μοντέλο teacher αποτελεί έναν transformer που έχει περισσότερες παραμέτρους απο το μοντέλο students και σκοπός είναι το μοντέλο students να αντλήσει πληροφορίες οχι μονο απο το cross entropy loss στις ground truth ετικέτες αλλά και απο τις προβλέψεις του μοντέλου teacher στα δεδομένα. Ο τρόπος που πραγματοποιείται αυτό είναι εφαρμόζοντας δύο διαφορετικές τεχνικές augmentation στα δεδομένα που παραγουν δυο διαφορετικές αναπαραστάσεις για την εικόνα, οι οποίες συμβολίζονται με u_t, u_s και στην συνέχεια τροφοδοτούνται στα μοντέλα teacher,student αντίστοιχα, τα οποία παράγουν τις αντίστοιχες προβλέψεις $u_t = P_{\partial'}(u)$ και $u_s = P_{\partial}(u)$. Αυτές οι προβλέψεις χρησιμοποιούνται για την εκπαίδευσή του student μοντέλου ελαχιστοποιώντας το cross entropy loss στις κατανομές που υπολογίστηκαν

$$\mathcal{L} = -P_{\partial'}(u)^T \log P_{\partial(u)} \tag{4.32}$$

4.5.4 iBOT[7]

Για την εκπαίδευση του ViT χρειάζεται να χρησιμοποιηθούν visual tokenizers για την συγκεκριμένη εκπαίδευση χρησιμοποιήθηκε το iBOT [7], το οποίο εφαρμόζει το MIM που αναλύθηκε στο 4.5.2 ως μια τεχνική απόσπασης πληροφοριών. Ο συγκεκριμένος tokenizer πετυχαινει να αποσπάσει νοηματικές πληροφορίες υψηλού επιπέδου εφαρμόζοντας τους χάρτες ομοιότητας μεταξύ των εικόνων για τα token των κλάσεων. Επιπλέον ο συγκεκριμένος tokenizer δεν χρειάζεται κάποια επιπλέον βήματα pre-training καθώς βελτιστοποιείται μαζί με το MIM.

Οι μέθοδοι που αναπτύχυθηκαν στο iBOT εμπνεύονται απο τις τεχνικές εκπαίδευσης που αναπτύχθηκαν στο DINO[96] που προτείνει το Self-Distillation το οποίο υπολογίζει το loss με χρήση της εξίσωσης 4.32 και το BEiT που έχει ως σκοπό να ελαχιστοποιήσει την εξίσωση4.33. Το iBOT χρησιμοποιεί το self-distillation loss για την δημιουργία token και πραγματοποιεί το MIM μέσω αυτού.

$$-\sum_{i=1}^{N} m_i \cdot P_{\phi}(x_i)^T \log P_{\partial}(\hat{x}_i) \tag{4.33}$$

Για την εκπαίδευση του μοντέλου, όπως αναφέρθηκε και προηγουμένως πραγματοποιούνται δύο τυχαία διαφορετικά augmentations, τα οποία παράγουν δύο διαφορετικές αναπαραστάσεις αυτής της εικόνας και συμβολίζονται με u, v. Στην συνέχεια αυτές οι εικόνες χωρίζονται σε patches και τους εφαρμόζεται blockwise masking. Έπειτα οι unmasked εκδοχές των εικόνων εισάγονται στο δίκτυο teacher, ενώ οι masked εκδοχές στο student που περιέχουν τα patches, τα οποία είναι προς ανακατασκευή. Οι αναπαραστάσεις των δεδομένων γράφονται ως $u_t = P_{\partial'}(u)$ και $u_s = P_{\partial}(u)$. Χρησιμοποιώντας αυτές τις κατανομές μπορεί να οριστεί το loss του MIM, το οποίο χρησιμοποιείται για την εκπαίδευση και δίνεται από τον τύπο

$$\mathcal{L}_{MIM} = -\sum_{i=1}^{N} m_i \cdot P_{\partial}(u_i)^T \log P_{\partial}(\hat{u}_i) \tag{4.34}$$

Το μοντέλο που χρησιμοποιείται για την εκπαίδευση έχει ως backbone τον vision transformer και χρησιμοποιείται επίσης ένα projection head, έτσι το μοντέλο network δημιουργεί κατανομές για κάθε patch, το οποίο περνάει από mask. Όπως αναφέρθηκε και προηγουμένως ο tokenizer, που χρησιμοποιείται στο iBOT είναι επίσης εκπαιδεύσιμος στο MIM χωρις την ανάγκη να εφαρμοστεί κάποιο pre-train σε αυτή. Το iBOT έχει ως σκοπό να ελαχιστοποιήσει δυο losses, το πρώτο είναι το $\mathcal{L}_{[cls]}$, το οποίο αφορά το σφάλμα μεταξύ των δύο διαφορετικών αναπαραστάσεων των των student,teacher transformers.

$$\mathcal{L}_{[CLS]} = -P_{\partial'}(v)^T \log P_{\partial}^{[CLS]}(u) \tag{4.35}$$

Αλγοριθμος 4.6: *iBOT[7] αλγόριθμος*

Είσοδος:
gs, gt ▷ Μοντέλα μαθητή και δασκάλου
C, C_0 ▷ Κέντρο cross view tokens για δάσκαλο και μαθητή
τ_s, τ_t ▷ Θερμοκρασία για normalization στα cross view tokens για δάσκαλο και μαθητή
$\tau_s^{'}, \tau_t^{'}$ ▷ Θερμοκρασία για normalization στα patches για δάσκαλο και μαθητή
l ▷ Παράμετρος ανανέωσης παραμέτρων για το δίκτυο
$m, m^{'}$ ▷ Παράμετρος ανανέωσης παραμέτρων για τα tokens και patches
$g_t.params = g_s.params$
for x in *loader* **do**
u, v = augment(x), augment(x) ▷ Παραγώγη ξεχωριστών αναπαραστάσεων για την ιδια εικόνα
$\hat{u}, m_u$ = blockwise mask(u) ▷ Τυχαια κάλυψη κάποιων tokens
$\hat{v}, m_v$ = blockwise mask(v) ▷ Τυχαια κάλυψη κάποιων tokens
$\hat{u}_s^{[CLS]}, u_{\pi\alpha\tau\sigma\eta}^s = g_s(\hat{u}, \text{return all tok=true})$ ▷ [n, K], [n, S^2, K]
$\hat{v}_{[CLS]}^s, \hat{v}_{\text{patch}}^s = g_s(\hat{v}, \text{return all tok=true})$ ▷ [n, K], [n, S^2, K]
$\hat{u}_t^{[CLS]}, \hat{u}_t^{[patch]} = g_t(\hat{u}, \text{return all tok=true})$ ▷ [n, K], [n, S^2, K]
$\hat{v}_{[CLS]}^t, \hat{v}_{[patch]}^t = g_t(\hat{v}, \text{return all tok=true})$ ▷ [n, K], [n, S^2, K]
$\mathcal{L}_{[CLS]} = \frac{H(\hat{u}_{[CLS]}^s, \hat{v}_{[CLS]}^t, C, \tau_s, \tau_t)}{2} + \frac{H(\hat{v}_{[CLS]}^s, \hat{u}_{[CLS]}^t, C, \tau_s, \tau_t)}{2}$
$\mathcal{L}_{MIM} = \frac{(m_u \cdot H(\hat{u}_s^{[patch]}, \hat{u}_t^{[patch]}, C^{'}, \tau_s^{'}, \tau_t^{'})).\text{sum(dim=1)}}{m_u.\text{sum(dim=1)}}/2 + \frac{(m_v \cdot H(\hat{v}_{[patch]}^s, \hat{v}_{[patch]}^t, C^{'}, \tau_s^{'}, \tau_t^{'})).\text{sum(dim=1)}}{m_v.\text{sum(dim=1)}}/2$
$(L_{[CLS]}.\text{mean}() + L_{MIM}.\text{mean}()).\text{backward}()$
update(g_s) ▷ ανανέωση δασκάλου και μαθητή
$g_t.params = l \cdot g_t.params + (1 - l) \cdot g_s.params$
$C = m \cdot C + (1 - m) \cdot \text{cat}([u_t^{[CLS]}, v_t^{[CLS]}]).\text{mean(dim=0)}$
$C = m^{'} \cdot C^{'} + (1 - m^{'}) \cdot \text{cat}([u_t^{patch}, v_t^{patch}]).\text{mean(dim=0,1)}$
end for
procedure ($H(s, t, c, \tau_s, \tau_t)$)
$t = t.\text{detach}()$;
$s = \text{softmax}(s/\tau_s, dim = 1)$;
$t = \text{softmax}((t - c)/\tau_t, dim = 1)$;
return $-(t \cdot \log(s).\text{sum(dim = -1)}$;

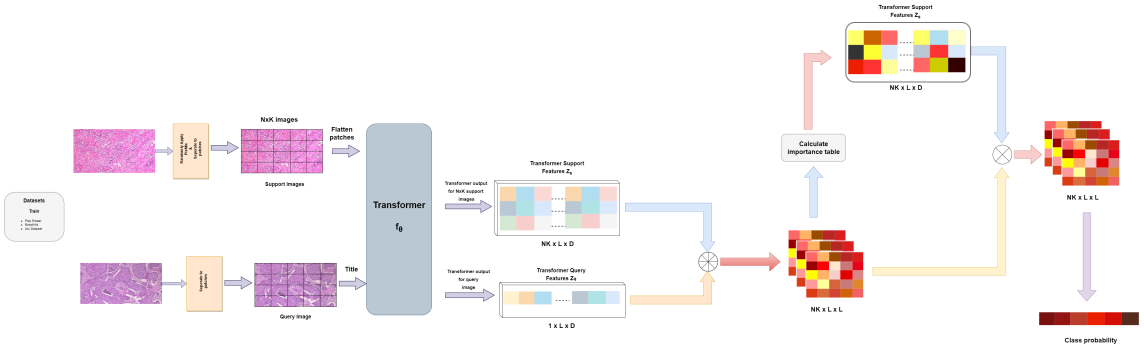
4.5.5 Pre-Train

Εμπνευσμένοι από το [7] και απο το [3] για την εκπαίδευση του transformer πάνω στα train δεδομένα η εκπαίδευση αντι να ξεκινήσει από την αρχή χρησιμοποιείται ένα δίκτυο, το οποίο είχε προεκπαιδευτεί στο mini-imagenet και αποτελεί ενα υποσύνολο του ImageNet[97] με 100 κλάσεις, που αφορούν γενικότερες κατηγορίες αντικειμένων. Αφου το δίκτυο εκπαίδευτει σε αυτά τα δεδομένα στην συνέχεια φορτώνονται οι παράμετροι του και πραγματοποιείται fine-tuning πάνω στα train δεδομένα του few shot learning.

4.5.6 Ταξινόμηση με επαναυπολογισμό των παραμέτρων της ομοιότητας των tokens

Οι εικόνες πριν την εισαγωγή τους στους transformers χωρίζονται σε patches, ο αριθμός των οποίων μπορεί να εκφραστεί απο την σχέση $M = \frac{HxW}{P^2}$ και συμβολίζεται ως $p = p_{i=1}^{iM}$. Στην

συνέχεια ολα αυτά τα patches τοποθετούνται ακολουθιακά σε ενα διάνυσμα μιας διάστασης και στην συνέχεια τα support αλλα και τα query αυτά διανύσματα εισάγονται στο transformer δίκτυο, όπου σαν έξοδο επιστρέφονται τα embeddings, τα οποία συμβολίζονται με Z_s, z_q, οπου καθε ενα απο αυτά τα embeddings αποτελείται απο tokens, τα οποία επλέγούμε να τα συμβολίσουμε ως εξής $Z_s = \{z^s_{nk} \,|\, n = 1, \ldots, N; k = 1, \ldots, K\}$. Καθέ ενα απο αυτά τα token μπορεί να εκφραστεί ως $z^s_{nk} = \{z^s_{nkl} \,|\, l = 1, \ldots, L; z^s_{nkl} \in \mathbb{R}^D\}$ και τα query tokens εκφράζονται ως $z_q = \{z_{lq} \,|\, l = 1, \ldots, L; z_{lq} \in \mathbb{R}^D\}$. Στις παραπάνω εξισώσεις το n αναπαριστά των αριθμό των κλάσεων, το k αναπαριστά τον αριθμό των δειγμάτων και το l εκφράζει των αριθμό των patches.



Σχήμα 4.22: *FewTURE important generation*

Αφου εξαχθούν τα embeddings για όλα τα support δεδομένα αλλά και για κάθε query δημιουργείται μια νοηματική σύνδεση υπολογίζοντας την ομοιότητα κάθε patch των support δεδομένων με κάθε patch των query δεδομένων. Με αυτόν τον τρόπο δημιουργείται ενας πίνακας που παρουσιάζει την ομοιότητα μεταξύ των δειγμάτων που συμβολίζεται ως $S \in \mathbb{R}^{N \cdot K \cdot LxL}$, όπου κάθε στοιχείο S υπολογίζεται απο μια συνάρτηση ομοιότητας και συνήθως χρησιμοποιείται το cosine similarity. Όπως αναλύθηκε και προηγουμένως η εύρεση ομοιοτήτων σε tokens μπορεί να οδηγήσει σε λάθος συμπεράσματα καθώς μπορεί να βρεθούν κοινά στοιχεία σε δείγματα, τα οποία ανήκουν σε διαφορετικές κλάσεις. Για αυτόν τον λόγο εκτός απο τα similarity maps υιοθετείται και ενας πίνακας, ο οποίος είναι task specific και έχει τιμές που εκφράζουν την σημαντικότητα που έχουν τα tokens στην σωστή ταξινόμηση των δειγμάτων. Αυτός ο πίνκας συμβολίζεται ως $u \in \mathbb{R}^{N \cdot K \cdot Lx1}$ και ανανεώνεται με μεθόδους optimization, που βασίζονται στην ταξινόμηση των support δειγμάτων. Στην συνέχεια αυτός ο πίνακας προστίθεται στον χάρτη ομοιότητων S με την μέθοδο broadcasting ως προς στήλη $\tilde{S} = S + [u \cdot 1^{1xL}]$, όπου κάθε αντικείμενο συμβολίζεται ως s^{l_s,l_q}_{nk}. Για την ταξινόμηση των δειγμάτων εφαρμόζεται μια κανονικοποιήση των στοιχείων του πίνακα S και στην συνέχεια εφαρμόζεται λογαριθμικό άθροισμα στα tokens, που ανήκουν στην ίδια κλάση των support συνόλων. Αφού εφαρμοστεί αυτό το άθροισμα και προκύψουν N τιμές, που εκφράζουν τον αριθμό τον κλάσεων εφαρμόζεται μια softmax συνάρτηση και το query δείγμα κατατάσεται στην κλάση, που έχει την υψηλότερη τιμή απο την έξοδο softmax. Ο υπολογισμός που περιγράψαμε παραπάνω μπορεί να εκφραστεί απο την εξίσωση

$$\hat{y}_q = \text{softmax}\left(\left(\log \sum_{k=1}^{K} \sum_{lq=1}^{L} \sum_{ls=1}^{L} \exp\left(\frac{s^{nk}_{ls,lq}}{\tau_S}\right)\right)_{n=1}^{N}\right) \tag{4.36}$$

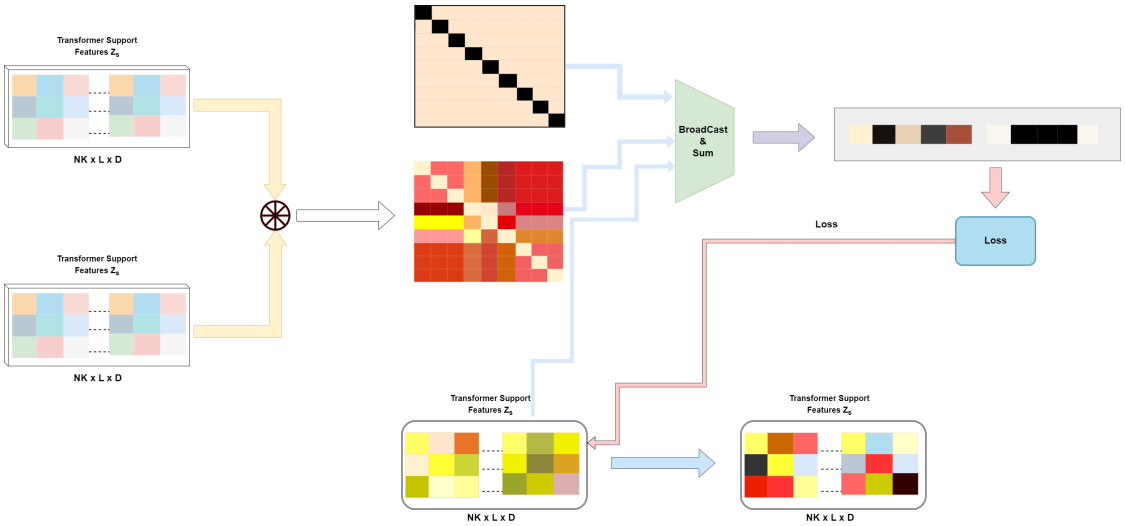
4.5.7 Υπολογισμός τιμών πίνακα σημαντικότητα

Για τον υπολογισμό των τιμών της σημαντικότητας των tokens χρησιμοποιούνται όλα τα δείγματα του support συνόλου του task. Η συγκεκριμένη μέθοδος έχει ως σκοπό να ταξινομήσει τα support δείγματα αντι για τα query δειγματα. Για την εκπαίδευση αυτής της μεθόδου χρησιμοποιείται ο πίνακας με τα support tokens Z_s, που ορίσαμε στα προηγούμενα βήματα και επίσης χρησιμοποιείται και ενα πίνακας Z_{sq} που αποτελείτε ακριβώς απο τα ίδια tokens με τον Z_q, αλλά για τον οποίο δεν γνωρίζουμε τις ετικέτες. Στην συνέχεια με την ιδια μέθοδο που εφαρμόστηκε προηγουμένως πραγματοποιούμε και εδω pair wise similarity μεταξύ των Z_s, Z_{sq} και προκύπτει ο πίνακας $S_S \in \mathbb{R}^{N\cdot K\cdot LxN\cdot K\cdot L}$. Ο πίνακας με την σημαντικότητα των tokens αρχικοποιείται ως $u^0 = 0 \in \mathbb{R}^{N\cdot K\cdot Lx1}$ και στην συνέχεια προστίθεται ως προς στήλη στον S και προκύπτει ο $\tilde{S} = S + [u \cdot 1^{1xL}]$. Σκοπός αυτής της μεθόδου είναι η εύρεση των tokens, που συνεισφέρουν θεττικά στην σωστή ταξινόμηση και στην εύρεση των tokens, που συνεισφέρουν αρνητικά στην ταξινόμηση. Βεβαια σε αυτήν την περίπτωση τα support δεδομένα και τα pseudo query δεδομένα είναι τα ίδια οπότε εαν απλά εφαρμοστεί το similarity ως προς τα patches τότε αυτά που είναι ίδια θα έχουν πολύ υψηλό similarity και θα ταξινομούνται σωστά μόνα τους, για αυτό τον λόγο εφαρμοζεται ενα διαγώνιο masking με blocks διαστάσεων LxL στα δεδομένα, που προκύπτουν απο τον χάρτη των ομοιότητων με σκοπό η ταξινόμηση να βασίζεται σε πληρόφορίες που προκύπτουν απο τις υπόλοιπες εικόνες. Στην συνέχεια εφαρμόζεται μια κανονικοποίηση στις τιμές, που προκύπτουν από τον χάρτη και εφαρμόζεται ένα λογαριθμικό άθροισμα για όλα τα δεδομένα που ανήκουν στις ίδιες κλάσεις και στην συνέχεια εφαρμόζεται μια συνάρτηση softmax, όπου η μεγαλύτερη από αυτές τις τιμές είναι και η κλάση που επιλέγεται στην ταξινόμηση. Στην συνέχεια κατα την διαδικασία της εκπαίδευσης υπολογιζεται το cross entropy loss, όπου και η ελαχιστοποιηση αυτού του σφάλματος αποτελεί τον στόχο της εκπαίδευσης. Την ελαχιστοποίηση αυτού του σφάλματος για όλα τα δείγματα μπορούμε να την εκφράσουμε ως εξής

$$\arg\min_{u} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathcal{L}_{\text{CE}}(y_s^{nk}, \hat{y}_s^{nk}(u)) \tag{4.37}$$

Με την χρήση της πρόσθεσης ανα στηλη στον πίνακα σημαντικότητας των tokens οι τιμές σημαντικότητας των παραμέτρων διαμοιράζονται σε όλα pseudo query tokens και ετσι το μοντέλο ενθαρρύνεται να εκπαιδευτεί στα δεδομένα χρησιμοποιώντας όλες τις διαθέσιμές πληροφορίες. Με αυτον τον τρόπο ενισχύονται οι σχέσεις, που απεικονίζουν τις ομοιότητες των δειγματων μεταξύ δεδομένων της ίδιας κλάσης, ενώ οι ομοιότητες που εμφανίζονται σε δεδομένα διαφορετικών κλάσεων αποθαρρύνονται.

Η συγκεκριμένη μέθοδος χρησιμοποιεί καινούρια στοιχεία, τα οποία δεν έχουν εφαρμοστεί σε μεγάλο βαθμό στην επίλυση προβλημάτων few shot learning, όπως η εφαρμογή του self supervised learning, που φαινεται [6] να αποδίδει αρκετά καλύτερα σε σύγκριση με supervised τεχνικές σε σύνολα δεδομένων, όπως το (mini-imagenet, fc-100, cifar, tieredimagenet). Επίσης αντι για τις τεχνικές που χρησιμοποιούσαν ως backbone CNN μοντέλα και εξήγαγαν χαρακτηριστικά πάνω στα οποία βασίζονταν για την ταξινόμηση των δεδομένων ή σε τεχνικές optimization, που αναλύθηκαν στο (4.3.3,4.3.3) στην συγκεκριμένη περίπτωση χρησιμοποιήθηκαν transformers, οι οποίοι συχνα αποφεύγονται λόγω του μεγάλου όγκου



Σχήμα 4.23: *FewTURE important generation*

δεδομένων που χρειάζονται στην εκπαίδευση.

Κεφάλαιο 5

Υλοποίηση

Στο συγκεκριμένο κεφάλαιο περιγράφονται πιο συγκεκριμένα οι τρόποι υλοποίησης των αλγορίθμων καθώς και πληροφορίες σχετικά με τις υπερπαραμέτρους.

5.1 Βασικές Μέθοδοι Αναπτυξής των Αλγορίθμών και επεξεργασίας των Δεδομένων

Στην ενότητα αυτή παρουσιάζονται οι βασικοί μέθοδοι στους αλγόριθμους που χρησιμοποιήθηκαν, οι υπερπαράμετροι που επιλέχθησαν αλλά και τα πειράματα που πραγματοποιήθηκαν.

5.1.1 Ανάλυση και Προεπεξεργασια Δεδομένων

Τα σύνολα δεδομένων που επιλέχθηκαν είναι το Pap Smear το οποίο αποτελείται απο 7 κλάσεις που περιέχουν εικόνες που έχουν ληφθεί απο μικροσκόπιο και απεικονίζουν κυτταρα τα οποία έχουν δειγματοληπτειθεί απο test Pap που είχαν πραγματοποιηθεί στο Herlev University Hospital. Παρατηρούμε οτι οι κλάσεις αυτου του συνόλου δεδομένων είναι ανισοκατανεμημένες. Απο ολές τις κλάσεις επιλέγονται ως train δεδομένα, οι κλάσεις που έχουν τα περισσότερα δείγματα, ενω για test δεδομένα novel Dataset επιλέγοναι οι κλάσεις που έχουν τον μικρότερο αριθμό δειγματών. Για τα σύνολα δεδομένων BreakHis, ISIC2018 ακολουθηθηκέ παρόμοια διαδικασία.

Τα δείγματα του Pap Smear περιέχουν εικόνες διαφορετικών διαστάσεων έτσι για την εύκολη εισαγώγη τους στα μοντέλα εκπαίδευσης (όπως CNN, Transformers, κ.λ) πραγματοποιείται μια ρύθμιση του μεγέθους τους σε διαστάσεις 84x84. Επίσης παρατηρήθηκε οτι για τα δείγματα του Pap Smear και του Isic2018 η κύρια πληροφορία για την εξαγωγή συμπερασμάτων στην εικόνα βρίσκεται στο κέντρο αυτής. Ετσι πραγματοποιείται ενα center crop σε αυτά τα δεδομένα για να επικεντρωθεί το σύστημα στην εξαγωγή χαρακτηριστικών απο τα σημαντικότερα σημεια της εικόνας. Για το BreakHis dataset είναι πολύ δύσκολη η εξαγωγή συμπερασματών σχετικά με την σημαντικότητα των πληροφοριών πάνω στην εικόνα για ενα άτομο το οποίο δεν διαθέτει εξειδικευμένες γνώσεις.

Για το σύνολο δεδομένων BreakHis όλες οι εικόνες έχουν τις ίδιες διαστάσεις οι οποίες είναι (700x460). Επειδή αυτές οι διαστάσεις είναι πολύ μεγάλες και δημιουργούν προβλήματα με την περιορισμένη διάθεση μνήμης ram πραγματοποιείται resize. Στην συγκεκριμένη

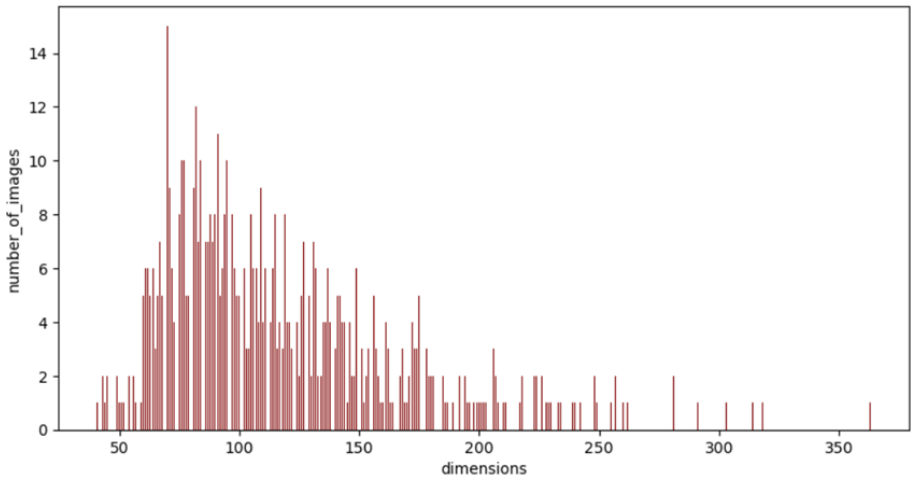
περίπτωση οι εικόνες μετατρέπονται σε (84x84) σε άλλες περιπτώσεις είδαμε ότι μπορούν να μετατραπούν σε (224 x 224). Το ISIC2018 αποτελείται από εικόνες που έχουν όλες διαστάσεις (600x450). Για τον ίδιο λόγο που μετατρέψαμε τις εικόνες του BreakHis σε μικρότερες διαστάσεις μετατρέπουμε και τις εικόνες του ISIC2018 σε (84x84)

Dataset	Classes	Number of samples	Domain(train or test)	Dimensions	Image Format
BreakHis	adenosis	113	novel	700 x 460	png
	tubular_adenoma	121	novel	700 x 460	png
	phyllodes_tumor	142	novel	700 x 460	png
	papillary_carcinoma	150	train	700 x 460	png
	lobular_carcinoma	170	train	700 x 460	png
	mucinous_carcinoma	222	train	700 x 460	png
	fibroadenoma	260	train	700 x 460	png
	ductal_carcinoma	903	train	700 x 460	png
ISIC2018	dermatofibroma	115	novel	600 x 450	jpg
	vascular_lesion	142	novel	600 x 450	jpg
	actinic_keratosis	327	novel	600 x 450	jpg
	basal_cell_carcinoma	514	train	600 x 450	jpg
	benign_keratosis	1099	train	600 x 450	jpg
	melanoma	1113	train	600 x 450	jpg
	melanocytic_nevus	6075	train	600 x 450	jpg
Pap-Smear	normal_superficiel	74	novel	—	BMP
	normal_intermediate	70	novel	—	BMP
	normal_columnar	98	novel	—	BMP
	severe_dysplastic	196	train	—	BMP
	moderate_dysplastic	146	train	—	BMP
	light_dysplastic	182	train	—	BMP
	carcinoma_in_situ	110	train	—	BMP

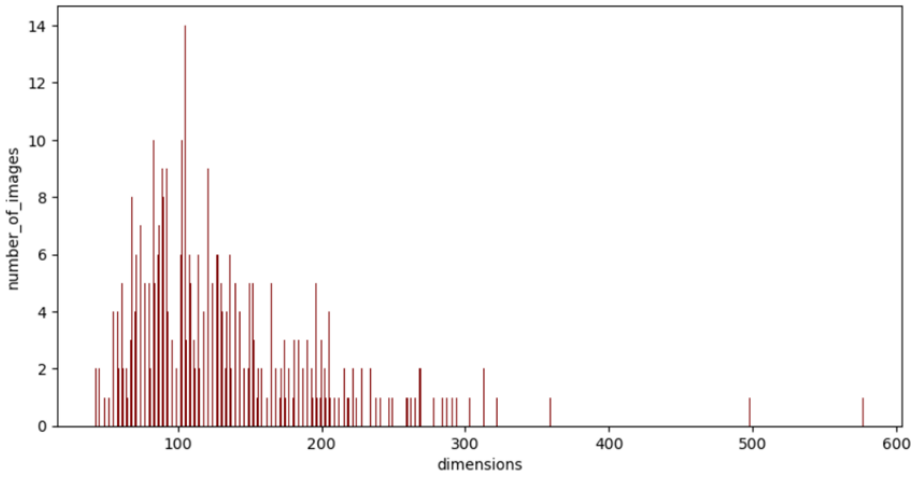
Table 5.1: *Information about datasets*

5.1.2 Αλγόριθμοι υλοποιήσης MetaMed

Παρακάτω φαίνονται και τα διαγράμματα που δείχνουν τις διαφορετικές διαστάσεις των εικόνων του Pap-Smear για το ύψος και το πλάτος των εικόνων αντίστοιχα.



Σχήμα 5.1: *Pap Smear width dimensions*



Σχήμα 5.2: *Pap Smear height dimensions*

5.1.3 Τεχνικές υλοποιήσης Transfer Learning

Για την τεχνική του Transfer learning το CNN δίκτυο αρχικά εκπαιδεύτηκε στα train δεδομένα των datasets Pap-Smear, Isic2018, BreakHis. Το δίκτυο CNN αποτελείται από 4 layer το κάθε ένα από αυτά τα layers περιέχει το CNN block το οποίο ορίζεται να περιέχει 32 φιλτρα στην συνέχεια αυτά περνάνε από ένα max polling με stride 2, έπειτα εφαρμόζεται μια συνάρτηση Relu και τέλος κατα την διάρκεια της εκπαίδευσης εφαρμόζεται και Batch Normalization.

Στα δεδομένα εφαρμόστηκαν πολλές διαφορετικές τεχνικές προεπεξεργασίας πριν εισαχθούν στο μοντέλο για εκπαίδευσης. Αρχικά τα δεδομένα τροφοδοτήθηκαν στο σύστημα χωρίς να του εφαρμοστεί κάποιο augmentation η μόνη αλλαγή που πραγματοποιήθηκε ήταν η αλλαγή του μεγέθους τους σε διαστάσεις $84x84$ και οι τιμές των εικόνων τροποποιήθηκαν με εφαρμογή ενος normalization, για να είναι μεταξύ του $[0,1]$. Στην συνέχεια πραγματοποιήθηκε το pre-training για 1000 εποχές. Πριν απο την διαδικασια της εκπαίδευσης τα train δεδομένα χωρίστηκαν σε train και validation, μετα την εκπαίδευση το μοντέλο δοκιμάστηκε στα validation δεδομένα όπου τα αποτελέσματα ήταν κοντά στο 60%. Αυτό δείχνει ότι τα δεδομένα για το train είναι πολύ λίγα για την εκπαίδευση του δικτύου. Επίσης τα validation δεδομένα χρησιμοποιήθηκαν για την εφαρμογή μιας μεθόδου Early Stopping όπου ελέγχει το loss των μοντέλων και επιλέγει να κρατήσει αυτο με το χαμηλότερο loss για το evaluation.

Στην συνέχεια δοκιμαστηκαν διαφορετικές τεχνικές augmentation για τον εμπλουτισμό του dataset πιο συγκεκριμένα εφαρμόστηκαν μέθοδοι Random Crop, Horizontal Flip, Vertical Flip μετα την εφαρμογή των augmentations και την εκπαίδευση αυτου του δικτύου δοκιμαστηκαν πάλι τα αποτελέσματα στα validation δεδομένα στα οποία παρατηρήθηκε αύξηση 10%. Τέλος χρησιμοποιήθηκαν και οι συνθέτες τεχνικές επαύξησης που αναλύθηκαν στα (4.2.2,4.2.2,4.2.2) και τα αποτελέσματα έδειξαν ότι το μοντέλο πέτυχε καλύτερη απόδοση με την εφαρμογή και αρχικών augmentation στην εικόνα αλλά και των σύνθετων. Το οποίο δείχνει ότι με αυτη την δημιουργία πιο σύνθετών εικόνων η εκπαίδευση με την μέθοδο του transfer learning μπορεί να παράξει μοντέλα με μεγαλύτερη ικανότητα γενίκευσης.

Αφου καταλήξαμε στο οτι τα καλύτερα αποτελέσματα τα παίρνουμε με εφαρμογή απλών augmentations και στην συνέχεια με εφαρμογή των σύνθετων (με πιθανότητα 50% και a =

0.25) augmentations προχωρήσαμε στην υλοποίηση της μεθόδου του few shot learning, όπου δημιουργήθηκε μια κλάση data generator που χώριζε τα δεδομένα ανα κλάση και στην συνέχεια για κάθε σύνολο δεδομένων support, query, test που ορίσαμε ανα κλάση δώσαμε μοναδικές τιμές ανα task. Αυτή η διαδικασία πραγματοποιήθηκε για 400 διαφορετικά task τα οποία τα χρησιμοποιούμε για το fine tuning και το τελικό evaluation του μοντέλου. Στην αρχή της εκπαίδευσης φορτώνουμε τις παραμέτρους που προέκυψαν απο το προηγούμενο βήμα που αναλύθηκε και ξεκινάμε το fine tuning στα train δεδομένα το οποίο το εκτελούμε για 50 επαναλήψεις πριν απο το evaluation. Αξίζει να σημειωθεί οτι σε αυτό το σημείο δεν προστέθηκε κάποιο από τα σύνθετα augmentations (cutout, cutmix, mixup) παρα μόνο τα αρχικά augmentations. Αυτό που παρατηρήθηκε στην συγκεκριμένη διαδικασία ηταν οτι πολλές φορές το μοντέλο οδηγούνταν σε overfitting κατα την διάρκεια της εκπαίδευσης, για την αποφυγή αυτου προστέθηκε μια διαδικασία Early Stopping. Όπου το μοντέλο ελέγχει τα αποτελέσματα του μέσω των query δεδομένων τα οποία χρησιμοποιούνται ως validation και αν το σφάλμα αυξάνεται πάνω σε αυτά για 10 συνεχόμενες επαναλήψεις τότε η διαδικασία εκπαίδευσης σταματά και το evalutation στα test δεδομένα πραγματοποιείται για τις παραμέτρους που πέτυχαν το χαμηλότερο loss. Με εφαρμογή αυτής της μεθόδου παρατηρούμε οτι τα αποτελέσματα που επιτεύχθηκαν ηταν υψηλότερα σε σχέση με την απλή εφαρμογή του transfer learning και σε κάποιες περιπτώσεις πλησίασαν αποτελέσματα πιο σύνθετων μεθόδων όπως το meta-learning.

Dataset	Few-shot task	Transfer-learning	Modified Transfer Learning
2-way			
Pap-Smear	3 shot	80.12	82.24
	5 shot	84.88	86.47
	10 shot	90.38	89.52
3-way			
Pap-Smear	3-shot	71.50	74.50
	5-shot	77.92	80.40
	10-shot	83.58	82.95

Table 5.2: *PapSmear Transfer Learning model accuracy*

Παρατηρώντας τα αποτελέσματα βλέπουμε ότι το μοντέλο που εκπαιδεύτηκε στις πιο σύνθετες εικόνες του συνόλου δεδομένων παράγει χαμηλότερο accuracy στις περιπτώσεις που το fine-tuning πραγματοποιείται σε μεγαλύτερο σύνολο δεδομένων, όπως οι περιπτώσεις του 10 shot σε σχέση με το μοντέλο transfer learning που χρησιμοποιεί απλές εικόνες. Σε αντίθεση το σύστημα σύνθετων εικόνων δίνει καλύτερα αποτελέσματα σε περιπτώσεις που τα δείγματα είναι λιγότερα όπως το 5-shot, 3-shot. Απο τα παραπάνω αποτελέσματα μπορούμε να καταλήξουμε στο συμπέρασμα ότι η μέθοδος των πιο σύνθετων augmentatins παράγει μοντέλα που δίνουν πιο γενικεύσιμα αποτελέσματα, αφου μπορούν να πετύχουν υψηλότερο accuracy σε περιπτώσεις που εκπαιδεύονται σε λιγότερα δείγματα. Παρόμοια αποτελέσματα παρατηρήθηκαν και για τα σύνολα δεδομένων (ISIC2018, BreakHis).

Dataset	Few-shot task	Transfer-learning	Modified Transfer Learning
2-way			
ISIC 2018	3 shot	66.88	70.64
	5 shot	73.88	74.70
	10 shot	80.38	81.44
3-way			
ISIC 2018	3-shot	55.67	56.44
	5-shot	59.67	60.33
	10-shot	65.92	66.50

Table 5.3: *ISIC2018 Transfer Learning model accuracy*

Κατα την εκπαίδευση του Transfer learning χρησιμοποιήθηκαν ξεχωριστα οι τεχνικές των advanced augmentation και δοκιμάστηκαν διαφορετικές τιμές για την πιθανότητα εφαρμογής τους. Τα καλύτερα αποτελέσματα παρατηρήθηκαν οταν για τα augmentations του cutmix, mixup επιλέχθηκε πιθανότητα 50%. Επίσης επειδή αυτά τα augmentations αναμειγνύουν με δεδομένα διαφορετικών κλάσεων χρησιμοποιήθηκαν one hot labels για τις κλάσεις και το loss που υπολογίστηκε ήταν το cross entropy loss. Επιπλέον δοκιμάστηκαν διαφορετικές τεχνικές optimizations όπως ο Adam, SGD σε αυτές τις δοκιμές παρατηρήθηκε οτι ο Adam optimizer παράγει καλύτερα αποτελέσματα και το learning rate που χρησιμοποιήθηκε ήταν 0.001.

Dataset	Few-shot task	Transfer-learning	Modified Transfer Learning
2-way			
BreakHis x40	3 shot	76.12	77.90
	5 shot	79.00	81.10
	10 shot	82.25	84.80
3-way			
BreakHis x40	3-shot	63.17	64.00
	5-shot	65.08	68.80
	10-shot	69.50	72.90
2-way			
BreakHis x100	3 shot	78.50	74.83
	5 shot	78.62	78.50
	10 shot	82.00	83.60
3-way			
BreakHis x100	3-shot	62.42	63.50
	5-shot	64.33	65.40
	10-shot	66.50	68.95
2-way			
BreakHis x200	3 shot	73.62	71.24
	5 shot	78.00	77.47
	10 shot	79.75	80.52
3-way			
BreakHis x200	3-shot	62.50	61.34
	5-shot	64.92	65.47
	10-shot	70.08	71.85
2-way			
BreakHis x400	3 shot	72.88	71.50
	5 shot	74.62	76.40
	10 shot	77.50	80.62
3-way			
BreakHis x400	3-shot	56.75	57.50
	5-shot	61.75	60.28
	10-shot	63.50	64.76

Table 5.4: *BreakHis Transfer Learning model accuracy*

5.1.4 Τεχνικές υλοποιήσης Meta Learning

Στην τεχνική εκπαίδευσης του meta learning ακολουθήσαμε παρόμοια διαδικασία για το fine tuning με αυτη που περιγράφηκε παραπάνω για την μέθοδο transfer learning. Η κύρια διαφορά αυτών των δύο ήταν στην διαδικασία του pre-training του μοντέλου όπου εκεί εφαρμόστηκαν μέθοδοι που βασίζονται σε optimization του μοντέλου για προσαρμογή σε νέα δεδομένα τα οποία δεν έχει ξαναδεί. Ο αλγόριθμος που χρησιμοποιήθηκε για να πετυχουμε ενα γενικεύσιμο και εύκολα προσαρμόσιμο μοντέλο ηταν ο Reptile. Για την εφαρμογή αυτού του αλγορίθμου έπρεπε να χωρίσουμε τα train δεδομένα σε tasks επίσης έπρεπε να ορίσουμε τον αριθμό των εσωτερικών επαναλήψεων που θα πραγματοποιούνταν στο δίκτυο. Τέλος ήταν αναγκαίο να βρεθεί η κατάλληλη τεχνική επαυξησης δεδομένων για κάθε dataset.

Αρχικά για τον αριθμό των task δοκιμάστηκαν μεγάλες τιμές, όπως το 40 οι οποίες δεν πρόσφεραν τόσο καλά αποτελέσματα για μικρά dataset. Αυτό συμβαίνει διότι για την δημιουργία πολλών task πολλά δεδομένα θα χρειαστει να επαναληφθούν σε ενα επεισοδιο εκπαιδευσης. Με αυτόν τον τρόπο κατα την ανανέωση των γενικών παραμέτρων στο βήμα του meta update, τα δεδομένα τα οποία θα επαναλαμβάνονται θα επηρεάζουν σε μεγαλύτερο βαθμό τις γενικές παραμέτρους σε σχέση με άλλα, έτσι είναι δυνατόν να δημιουργηθεί ενα bias προς αυτά τα δεδομένα.

Για τον αριθμό των εσωτερικών επαναλήψεων που θα εκτελεστούν με την χρήση του Adam optimizer δοκιμάστηκαν διαφορετικές τιμές. Αρχικά θέσαμε την τιμή 30. Κατα την εκτελεση του αλγορίθμου Reptile παρατηρήθηκε οτι ειδικά για τις περιπτώσεις που είχαμε μικρό αριθμό δειγμάτων για εκπαίδευση (5-shot,3-shot) το μοντέλο οδηγούνταν γρηγορά σε overfitting και οι παράμετροι που επιστρέφονταν στο στάδιο του meta update δεν βοηθούσαν στην δημιουργία ενός μοντέλου το οποίο θα μπορούσε να προσαρμοστεί σε νεα δεδομένα. Μετα απο μείωση των επαναλήψεων σε χαμηλότερο αριθμό παρατηρήσαμε ότι το μοντέλο ήταν πιο αποδοτικό για μικρότερο αριθμό εσωτερικών επαναλήψεων αλλά για μεγαλύτερο αριθμό εκτέλεσης επεισοδίων.

Για τις τεχνικές augmentation δεν παρατηρήθηκε κάποια που να ξεχωρίζει από τις άλλες. Συνήθως καλύτερα απέδιδαν οι τεχικές cutmix και mixup, αλλά σε μερικές περιπτώσεις το cutout και το απλό augmentation πέτυχαν καλύτερα αποτελέσματα.

Για την εκπαίδευση του μοντέλου χρειαζόμαστε παραμέτρους οι οποίες να μπορούν να προσαρμοστούν γρήγορα σε καινούργια δεδομένα. Έτσι κατα το validation του μοντέλου αντι να πραγματοποιείται απλός έλεγχος του accuracy στην ταξινόμηση των δεδομένων, πραγματοποιείται αρχικά ενα στάδιο fine tuning, όπου το μοντέλο προσαρμόζεται σε δεδομένα που θεωρούμε ως train και στην συνέχεια πραγματοποιείται το evaluation.

Αξίζει σε αυτό το σημείο να σημειωθεί ότι για κάθε task αρχικοποιείται ενας διαφορετικός Adam optimizer με σκοπο να μην υπάρξει Leakage μεταξύ των δεδομένων κατι που μπορεί να οδηγήσει στην εξαγωγή λάθος συμπερασμάτων για το μοντέλο.

Ένα ακόμα σημείο το οποίο αξίζει να αναφέρουμε είναι ότι κατα τον χωρισμό των δεδομένων σε tasks και πριν την εισαγωγή τους στο μοντέλο για την εκπαίδευση ή ακόμα και στο evaluation χωρίζονται σε minibatches, ο χωρισμός αυτός σε μικρά batches οδηγεί στην εξαγωγή καλύτερων αποτελεσμάτων σε σχέση με την εισαγωγή όλων των δεδομένων μαζι σαν ενα μεγάλο batch.

Το μοντέλο εκπαιδεύτηκε για 30000 επαναλήψεις επεισοδίων. Κατα την διάρκεια της εκπαίδευσης με τον αλγόριθμό Reptile παρατηρήθηκε οτι το μοντέλο στην αρχή της εκπαίδευσης ξεκινάει από χαμηλότερα accuracies και μετα από συγκεκριμένο αριθμό επαναλήψεων το accuracy αυξάνεται αλλά μετα απο την 10000 επανάληψη αρχίζει να μειώνεται μεχρι που μετα απο καποιες επαναλήψεις γύρω στην 15000 αρχίζει και αυξάνεται ξανα.

Τέλος αξίζει να σημειωθεί οτι για την εφαρμογή τέτοιων τεχνικών εκπαιδευσης που ειναι optimization based και βασίζονται σε πολλαπλές επαναλήψεις εκπαίδευσης και συχνή ανανέωση των παραμέτρων του μοντέλου είναι σημαντικό να επιλεγούν μοντέλα τα οποία δεν είναι πολύ περίπλοκα και δεν έχουν μεγάλο αριθμό παραμέτρων έτσι ωστε η εκπαίδευσή να μην κρατάει πάρα πολύ χρόνο αλλά και το μοντέλο να μην οδηγείται σε overfitting.

5.1.5 Τεχνικές υλοποιήσης FewTURE αλγορίθμου

Για την εκπαίδευση του αλγόριθμου στην μέθοδο του FewTURE που χρησιμοποιεί transformers χρειάστηκε μια προεπεξεργασία των δεδομένων πιο συγκεκριμένα η δημιουργία augmentations για τις εικόνες και ο διαχωρισμός τους σε μικρότερα patches. Για την εφαρμογή των augmentations ορίστηκε μια κλάση η οποία δημιουργεί δύο διαφορετικές ακολουθίες απο augmentations στις οποίες εφαρμόζονται τυχαία οι μέθοδοι της περιστροφής, αλλοίωσης χρώματος και η εφαρμογή Gaussian θορύβου. Αφου δημιουργηθούν αυτές οι ακολουθίες η πρώτη σειρά μετασχηματισμών εφαρμόζεται στην εικόνα που καλείται απο το dataset και αυτή η εικόνα αποθηκεύεται σε μια λίστα. Στην συνέχεια ανάλογα με τον αριθμό των διαφορετικών αναπαραστάσεων της εικόνας που έχει οριστεί ως υπερ παράμετρος, εφαρμόζεται η δεύτερη ακολουθία μετασχηματισμών τόσες φορές, όσες έχει οριστεί ο αριθμός μείον μια φορά που απεικονίζει την αρχική αναπαράσταση. Όλες οι εικόνες που έχουν προκύψει στην συνέχεια τοποθετούνται ακολουθιακά σε μια λίστα. Επίσης ανάλογα με τον αριθμό των τοπικών αναπαραστάσεων που έχουν οριστεί πραγματοποιείται τυχαία περικοπή στις εικόνες και εφαρμόζονται τεχνικές augmentation. Στην συνέχεια για τις εικόνες που έχουν προκύψει εφαρμόζεται με πιθανότητα 50% ο μετασχηματισμός Pixmix ο οποίος αναλύθηκε στο 4.2.2.

Κατα την εκτέλεση του pixmix augmentation επιλέγεται μια τυχαία εικόνα από το σύνολο δεδομένων fractals. Στην συνέχεια ελέγχεται η συνθήκη για το αν θα πραγματοποιηθεί τελικά το pixmix. Σε περίπτωση που δεν πραγματοποιηθεί το augmentation εφαρμόζεται απλά μια κανονικοποίηση των δεδομένων και στην συνέχεια μετατρέπονται σε τένσορες. Αν εφαρμοστεί η κανονικοποίηση τότε πραγματοποιείται ενα augmentation στην εικόνα απο τα (rotate, solarize, contrast, brightness, sharpness) και στην συνέχεια η εικόνα που προκύπτει αναμειγνύεται με την τυχαία εικόνα που επιλέχθηκε απο τα fractals. Η ανάμειξη γίνεται επιλέγοντας τυχαία μεταξύ της πρόσθεσης των δυο εικόνων ή των πολλαπλασιασμό αυτών. Αυτή η διαδικασία πραγματοποιείται επαναληπτικά ανάλογα με τον αριθμό της υπερπαραμέτρου που έχουμε ορίσει. Μετά την ολοκλήρωση αυτής της επαναληπτικής διαδικασίας εφαρμόζεται κανονικοποίηση των δεδομένων και στην συνέχεια μετατρέπονται σε τένσορες.

Μετα την δημιουργία των αναπαραστάσεων των εικόνων ορίζεται μια κλάση η οποία χωρίζει τα δεδομένα σε patches και στην συνέχεια εφαρμόζει Masking στα δεδομένα α-

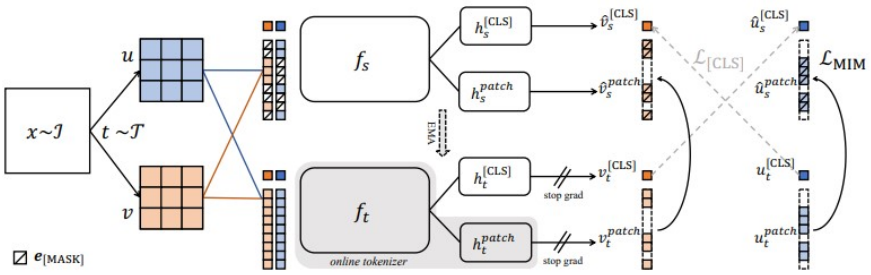
νάλογα με ποσοστό πρόβλεψης το οποίο έχει δοθεί. Το pred_ratio αποτελεί τον αριθμό που καθορίζει το ποσοστό του masking που θα εφαρμοστεί και είναι μια τιμή υπερπαραμέτρου η οποία επλέγεται μεταξύ $(0.0 - 1.0)$, επίσης επιλέγεται τιμή για το pred_ratio_var μεταξύ του $(0.0 - 1.0)$. Αν οι τιμές του pred_ratio και pred_ratio_var είναι απλοί δεδκαδικοί αριθμοί τότε για την τελική τιμή του pred_ratio επιλέγεται μια τιμή μεταξύ των $(pred\_ratio - pred\_ratio\_var, pred\_ratio + pred\_ratio\_var)$. Ενώ στην περίπτωση που το pred_ratio και το pred_ratio_var αποτελούν λίστες, πραγματοποιείται η ίδια διαδικασία που περιγράφηκε προηγουμένως για κάθε ζευγάρι των στοιχείων και απο τις τιμές που προκύπτουν επιλέγεται μια απο αυτές. Αφου επιλεχθεί η τιμή του pred_ratio υπολογίζεται η τιμή high που μας δίνει το άνω όριο των masks που θα πραγματοποιηθούν. Το high υπολογίζεται ως το γινόμενο $pred\_ratio \cdot Height \cdot Width$. Τα σχήματα για τα masks τα επιλέγουμε ως blocks ο ορισμός των θεσεων τους,του μέγεθους τους και του αριθμού τους πραγματοποιείται ακολουθώντας την μέθοδο που παρουσιάστηκε στο BEiT[90]. Πιο συγκεκριμένα η υλοποίηση βασίζεται στο συγκεκριμένο κομμάτι [98]. Αναφορικά ο υπολογισμός πραγματοποιείται παίρνοντας την μικρότερη απο τις διαστάσεις των εικόνων στην οποία εφαρμόζεται ευκλέιδια διαίρεση με το 3 και στην συνέχεια υψώνεται η τιμή που προκύπτει στο τετράγωνο. Έτσι επιστρέφεται το low boundary για την επιλογή του mask. Επιπλέον μέσω τις τιμής του high που υπολογίσαμε προηγουμένως πραγματοποιείται η εκτίμηση του μέγιστου αριθμού των patches που θα εφαρμοστούν και υπολογίζεται ως η διαφορά του high - mask_count. Έπειτα υπολογίζεται η κλίμακα των δύο διαστάσεων ως το $e^{\log aspect\_ratio}$. Τέλος υπολογίζονται οι τιμές h, w ως $h = round\left(\sqrt{target\_area \cdot aspect\_ratio}\right)$, $w = round\left(\sqrt{\frac{target\_area}{aspect\_ratio}}\right)$ οι οποίες δηλώνουν τις διαστάσεις του κάθε mask. Με χρήση αυτών των τιμών ορίζεται το ανω και το αριστερό άκρο του mask. Αφου υπολογιστούν οι θέσεις και το μέγεθος των masks επιστρέφονται μαζι με τις εικόνες.

Pretraining

Στην συνέχεια ορίζονται τα μοντέλα τα οποία χρησιμοποιήθηκαν στην εκπαίδευση. Αρχικά ορίζεται το μοντέλο student στο οποίο εφαρμόζεται το masked image modeling, ενώ στο μοντέλο του teacher δεν εφαρμόζεται MiM, επίσης δεν εφαρμόζεται drop_rate. Το μοντέλο ViT transformer ορίζεται ως 12 blocks τα οποία αποτελούνται απο ενα Normalization layer, ενα Attention layer, ενα δεύτερο Normalization layer και ενα mlp layer. Πριν την εισαγωγή των δεδομένων στο ViT Transformers εφαρμόζεται το PatchEmbed το οποίο επεξεργάζεται τις εικόνες και τις χωρίζει σε patches με την χρήση της μεθόδου Conv2d. Στην συνέχεια ορίζονται τα cls_tokens τα οποία αποτελούν πίνακες με μηδενικές τιμές, οι οποίες θα χρησιμοποιηθούν στην εκπαίδευση για την εξαγωγη γενικότερων αποτελεσμάτων. Ακόμα θα χρησιμοποιηθούν και positional embedding για να αναδιοργανώσουν τα patches στην σωστή σειρά. Αυτά τα positional embeddings εκπαιδεύονται κατα την διάρκεια του pre-training και σε ορισμένες φορές κατα την διάρκεια του fine-tuning. Όπου κατα την διάρκεια της εκπαίδευσης αυτά τα embeddings συγκλίνουν σε έναν διανυσματικό χώρο όπου δείχνουν μεγάλη ομοιότητα με τα γειτονικά τους positional embeddings. Αφου έχουν δημιουργηθεί τα embeddings στην συνέχεια εισάγονται στο σύστημα του Transformer όπου το πρώτο αντικείμενο στο οποίο τροφοδοτούνται είναι το Attention module το οποίο ελέγχει τις

ομοιότητες μεταξύ των tokens με την διαδικασία που εξηγήθηκε στο 2.2.5. Στην συνέχεια εφαρμόζονται μέθοδοι κανονικοποιήσης και τα embeddings περνούν από το στάδιο του Mlp. Τέλος αν έχει οριστει fc normalization εφαρμόζεται αλλιώς επιστρέφονται τα δεδομένα. Το μοντέλο του teacher ορίζεται με τον ίδιο τρόπο και ακολουθεί την ίδια αρχιτεκτονική χωρις όμως να εφαρμόζεται σε αυτό η μέθοδος του MIM. Για την εκπαίδευση του μοντέλου όπως αναφέρθηκε και προηγουμένως χρησιμοποιείται το iBOT[7].

Το iBOT όπως περιγράψαμε ακολουθεί μια self-supervised εκπαίδευση έτσι μετα την εξαγωγή των προβλέψεων απο το μοντέλο ορίζονται τα κέντρα για το [cls] και για τα patch που θα χρησιμοποιηθούν για τον υπολογισμό των losses $\mathcal{L}_{cls}, \mathcal{L}_{MiM}$. Ο υπολογισμός του loss πραγματοποιείται με την διαδικασία που περιγράφεται στο 4.6 όπου για το cls εφαρμόζεται μια συνάρτηση softmax στις προβλέψεις που έχουν προκύψει για το teacher δίκτυο. Στην συνέχεια αυτές χωρίζονται σε τμήματα ανάλογα με τον αριθμό των αναπαραστάσεων που έχει οριστεί ως υπερπαράμετρος, για τα patches πραγματοποιείται η ίδια διαδικασία. Στην συνέχεια κατα τον υπολογισμό του loss παίρνουμε τις προβλέψεις των μοντέλων του student, teacher. Αν η πρόβλεψη αφορά ενα patch στο οποίο έχει πραγματοποιηθεί masking τότε ο υπολογισμός του loss πραγματοποιείται σύμφωνα με την σχέση 4.34 ενω για την περίπτωση που η πρόβλεψη αφορά patches τα οποία δεν είναι Masked χρησιμοποιείται το loss 4.35. Αφου υπολογιστούν αυτά τα σφάλματα στην συνέχεια πραγματοποιείται ανανέωση των κέντρων για τα patches και για τα [CLS]. Το τελικό σφάλμα υπολογίζεται ως το άθροισμα αυτών των δυο. Το οποίο στην συνέχεια χρησιμοποιείται για την ανανέωση των παραμέτρων μέσω του optimizer.



Σχήμα 5.3: *Ibot framework*

Αυτή η διαδικασία αποτελεί το κομμάτι του pre-training του μοντέλου και εκτελέστηκε για 1600 εποχές. Για την εκπαίδευση του transformer. Κατα την διάρκεια εκτέλεσης αυτής της διαδικασίας χρησιμοποιήθηκαν μόνο τα train δεδομένα απο τον διαχωρισμό του dataset και δεν χρησιμοποιήθηκε κανένα απο τα δείγματα του novel dataset. Το επόμενο στάδιο που ακολουθείται για την προσαρμογή του μοντέλου στα novel δεδομένα αποτελεί το στάδιο του Meta train και αφορά τα maps τα οποία παρουσιάστηκαν και στο (4.23,4.22).

Στα αρχικά στάδια του pre-training του few_TURE μοντέλου η εκπαίδευση πραγματοποιήθηκε με τυχαία αρχικοποίηση των παραμέτρων και στην συνέχεια η εκπαίδευση αυτων πάνω στα δεδομένα του train. Κάτι τέτοιο λόγο του μικρού αριθμού των διαθέσιμων δεδομένων για εκπαίδευση οδήγησε το μοντέλο γρήγορα στο overfitting. Για την αντιμετώπιση αυτού του προβλήματος δοκιμαστηκαν να εφαρμοστουν σύνθετες τεχνικές augmentation μια από αυτές ήταν η τεχνική του pixmix 4.2.2 η οποία όπως περιγράφεται συνδύαζε τις εικόνες με

fractals για να δημιουργήσει πιο σύνθετες αναπαραστάσεις. Με την χρήση τέτοιων τεχνικών καταφέραμε να δημιουργήσουμε πιο σύνθετα δείγματα εικόνων και να και να αποφύγουμε σε μεγάλο βαθμό το overfitting.

Meta Training

Αφου ολοκληρώθει η διαδικασία του pre-train το επόμενο σταδιο είναι η προσαρμογή του μοντέλου στο domain του novel dataset. Στο στάδιο του Meta train χρησιμοποιούμαι επίσης το PixMix για τα δεδομένα πριν τα χωρίσουμε σε tasks με τον ίδιο τρόπο με τον οποίο το χρησιμοποιήσαμε στο στάδιο του pre-train. Το πρώτο στάδιο αποτελεί τον χωρισμό των δεδομένων σε tasks, η διαδικασία που ακολουθείται είναι παρόμοια με αυτή που περιγράφηκε στα προηγούμενα βήματα. Αρχικά ορίζεται ο αριθμός των δειγματών που θα υπάρχουν σε κάθε κλάση ενός task. Στην συνέχεια για το task οριζεται ο αριθμός των batches στα οποία θα χωριστεί το task. Ετσι για κάθε batch δειγματοληπτείται ένας αριθμός από κλάσεις στις οποίες πραγματοποιείται επαναληπτικά η επιλογή δειγμάτων, μολις ολοκληρωθεί αυτή η διαδικασια με την εντολή yield επιστρέφεται το batch το οποίο θα τροφοδοτηθεί στην συνέχεια στο σύστημα. Έπειτα αναταξινομούνται τα δεδομένα με σκοπό να είναι οργανωμένα ακολουθιακα δηλαδή πρώτα τα query της πρώτης κλάσης, στην συνέχεια αυτά της δεύτερης κ.ο.κ., το ίδιο πραγματοποιείται και για τα labels. Στο επόμενο βήμα της εκπαίδευσης του Meta-train φορτώνονται τα μοντέλα που εκπαιδεύτηκαν στο στάδιο του pre-train και απο αυτά εξήχθησαν τα embeddings τα οποία στην συνέχεια χωρίστηκαν στα support και στα query δεδομένα τα οποία αναταξινομούνται, ωστε οι σειρές του πίνακα να αποτελούν τον αριθμό των δειγμάτων. Αφου αποθηκευτούν τα embedings για τα support, query δεδομένα και βρεθεί ο αριθμός της ακολουθίας των patches για τα δυο σύνολα δεδομένων ακολουθεί η δημιουργια του masking για τα support δεδομένα, ώστε να μην στηρίζονται στο ίδιο δείγμα για την ταξινόμηση, οταν πραγματοποιείται η πρόβλεψη των pseudo-query δεδομένων. Στην συνέχεια πραγματοποιείται μια επαναληπτική ακολουθία για την οποία υπολογίζεται ο πίνακας σημαντικότητας των παραμέτρων. Το πρώτο βήμα είναι ο υπολογισμός του optimization του πίνακα με τις παραμέτρους σημαντικότητας. Έπειτα πραγματοποιούνται επαναλήψεις, όπου εξάγεται η πρόβλεψη του μοντέλου για τα support δεδομένα και υπολογίζεται το loss με βάση τα ground truth labels, όπου στην συνέχεια χρησιμοποιείται για την ανανέωση των παραμέτρων. Μετα την εκπαίδευση του διανύσματος σημαντικότητας των παραμέτρων υπολογίζεται η πρόβλεψη των query δεδομένων. Τα βήματα που ακολουθούνται για την εξαγωγή αυτών των προβλέψεων είναι ο υπολογισμός του cosine similarity μεταξύ των support και query δεδομένων, η πρόσθεση του διαγώνιου block masking και η πρόσθεση του διανύσματος σημαντικότητας μέσω broadcasting. Τέλος εξάγεται το prediction που πραγματοποιείται μέσω της εφαρμογής του temperature scaling και της λογαριθμικής πρόσθεσης των patches που σχετίζονται με τις query εικόνες. Στην συνέχεια αυτές οι προβλέψεις συγκρίνονται με τα ground truths και το loss που προκύπτει χρησιμοποιείται για ανανέωση των κύριων παραμέτρων του μοντέλου.

Για την εξαγωγή των αποτελεσμάτων στα δεδομένα πρόβλεψης το διάνυσμα σημαντικότητας των παραμέτρων εκπαιδεύεται στα support δεδομένα του dataset που χρησιμοποιείται για train, αφού πραγματοποιηθεί η εκπαίδευση με τον ίδιο τρόπο που περιγράφηκε

και προηγουμένως για τα βήματα του adaptation που έχουν οριστεί, υπολογίζεται το cosine similarity μεταξύ των query, support δεδομένων και πραγματοποιείται πρόσθεση μέσω broadcasting με το διάνυσμα που αποκρύπτει τα διαγώνια στοιχεία και το διάνυσμα που δείχνει την σημαντικότητα των παραμέτρων.

Pipeline εκπαίδευσης

Η εκπαίδευση κατα την διαδικασία του pre-training πραγματοποιείται για 1600 εποχές, η διαδικασία που χρησιμοποιήθηκε είναι αυτή που περιγράφηκε στο 5.1.5 και οι υπερ παράμετροι που χρησιμοποιήθηκαν είναι οι εξής

image size	84
global corps number	2
local corps number	10
patch size	16
number of validations	400
embeding dimnsion	1024
prediction ratio	[0, 0.3]
prediction ratio variance	[0, 0.2]
masking shape	block
batch size	256
epochs	1600
Optimizer	adamw

Table 5.5: *Pretrain parameters*

Κατα την διάρκεια της εκπαίδευσης παρατηρήθηκε οτι τα τρία datasets συνέκλιναν στις μέγιστες τιμές εκπαίδευσης του κοντά στην εποχή 800. Επιπλέον παρατηρήθηκε οτι με την αύξηση του batch size τα αποτελέσματα άρχισαν να αυξάνονται πράγμα το οποίο οφείλεται στην μεγαλύτερη εισαγωγή δεδομένων κατα την εκπαίδευση του μοντέλου. Επιπλέον για την εκπαίδευση χρησιμοποιήθηκε adam optimizer. Η διαδικασία της εκπαίδευσης πραγματοποιήθηκε με χρήση κάρτας γραφικών nvidia GTX1080ti.

Αφού ολοκληρώθηκε η διαδικασία του pretraining το επόμενο βήμα ήταν αυτό του Meta-training για αυτήν την εκπαίδευση ορίστηκαν 100 εποχές episodic training και οι υπερ παράμετροι που χρησιμοποιήθηκαν φαίνονται παρακάτω

image size	84
n way	2,3
k shot	3, 5, 10
query	15
number of episodes for 1 epoch of training	100
number of validation episodes	400
meta learning rate	0.0002
similarity temperature	0.051
optimizer	SGD

Table 5.6: *Meta train hyperparameters*

Η εκπαίδευση πραγματοποιήθηκε για όλους τους συνδυασμούς n-way, k-shot, μετα

την εκπαίδευση πραγματοποιήθηκε evaluation του μοντέλου για τα αποτελέσματα του στο dataset που θέλουμε να πραγματοποιηθεί το adaptation παίρνοντας το μέσο accuracy σε όλα για 400 episodes.

Τέλος παραθέτονται οι παράμετροι που χρησιμοποιήθηκαν για το Pixmix augmentation και για την περίπτωση του pretrain αλλά και για το meta train.

mixing parameter(beta)	4
severity of augmentation	1
number of mixing iterations	4
access to augmentations	all-ops
probability to apply pixmix	0.5
probability to apply augmentation	0.5

Table 5.7: *PixMix hyperparameters*

Metrics

Η μετρική που χρησιμοποιήθηκε για την εξαγωγή των αποτελεσμάτων ηταν αυτή του accuracy η οποία υπολογίστηκε ως:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.1}$$

Για την εξαγωγή των αποτελεσμάτων πραγματοποιήθηκε evaluation για 400 διαφορετικά tasks και από αυτά υπολογίστηκε ο μέσος όρος του accuracy με τον ίδιο τρόπο με τον οποίο πραγματοποιήθηκε στα MetaMed, PFEMED[3]. Στην αρχή η εκπαίδευση του δικτύου πραγματοποιήθηκε ξεκινώντας από τυχαίες παραμέτρους. Η εκπαίδευση με αυτήν την μέθοδο δεν κατάφερε να δώσει αποτελέσματα τα οποία ήταν συγκρίσιμα με αυτά του PFEMED, καθώς κυμαινόντουσαν γύρω στο 86% στο 2-way 10-shot task. Για την επίλυση αυτου του προβλήματος δοκιμάστηκαν διαφορετικοί τρόποι και ο πιο αποτελεσματικός απο αυτους, που είχε βοηθήσει και στην επίλυση διαφορετικών προβλημάτων, ήταν η έναρξη της εκπαίδευσης με παραμέτρους οι οποίες είχαν προκύψει απο προεκπαίδευση του δικτύου σε μεγαλύτερα dataset. Ετσι και στην συγκεκριμένη περίπτωση χρησιμοποιήθηκαν παράμετροι του δικτύου οι οποίες είχαν προεκπαιδευτεί στο mini-imagenet το οποίο αποτελεί ενα υποσύνολο του ImageNet, αυτό το υποσύνολο επιλέχθηκε έναντι ολόκληρου του dataset κυρίως λόγω των περιορισμένων πόρων που ήταν διαθέσιμοι. Επίσης, το ίδιο dataset είχε χρησιμοποιηθεί και στο PFEMED το οποίο είχε δώσει state of the art αποτελέσματα για το Pap-Smear.

Με την εφαρμογή αυτών των παραμέτρων και μετά με fine-tune αυτών στο train σύνολο των δεδομένων μας και τέλος εκτέλεση του βήματος του meta train για προσαρμογή στα novel δεδομένα προέκυψαν τα τελικά αποτελέσματα, τα οποία ήταν συγκρίσιμα με το PFEMED και στο 2-way 10-shot task οριακά το ξεπερνούσαν.

Dataset	Few-shot setting	Transfer learning	Meta-Med				PFEMED	Few-TURE
			Normal Augmen-tation	Cutout	Mixup	CutMix		
				2-way				
Pap-Smear	3 shot	80.12	85.37	84.12	90.12	86.75	95.53	**94.36±0.98**
	5 shot	84.88	86.50	87.87	91.12	87.87	95.87	**95.81±0.58**
	10 shot	90.38	89.37	91.50	93.00	93.37	96.00	**96.48±0.52**
				3-way				
Pap-Smear	3 shot	71.50	70.58	75.08	74.00	73.33	92.42	**89.75±0.62**
	5 shot	77.92	72.42	79.25	78.67	80.00	92.48	**91.96±0.42**
	10 shot	83.58	83.00	69.25	82.00	84.08	92.68	**92.63±0.34**

Πίνακας 5.8: *Pap-Smear Accuracy for all models*

Το επόμενο dataset πάνω στο οποίο εκπαιδεύτηκε το μοντέλο και για το οποίο εξήγαγε αποτελέσματα ήταν αυτό του ISIC το συγκεκριμένο σύνολο από δεδομένα αποτελείται από κλάσεις που εκφράζουν διαφορετικές παθήσεις που εμφανίζονται στο δέρμα. Οι εικόνες που περιέχει το συγκεκριμένο σύνολο από δεδομένα είναι φυσικές εικόνες και δεν έχουν εξαχθεί από κάποιο συγκεκριμένο ιατρικό μηχάνημα. Οι εικόνες εστιάζουν γενικά στο σημείο του δέρματος που εμφανίζεται η μη ομαλότητα. Η διαδικασία εκπαίδευσης που πραγματοποιήθηκε είναι ακριβώς η ίδια με αυτή που περιγράφηκε παραπάνω και για το Pap-Smear. Τα αποτελέσματα που προέκυψαν φαίνεται να είναι χαμηλότερα από αυτά του PFEMED, ωστόσο η απόσταση τους δεν είναι πολύ μεγάλη.

Train data results		
Dataset	Few-shot setting	Few TURE
	2-way	
ISIC 2018	3-shot	**82.14±1.35**
	5-shot	**85.98±1.21**
	10-shot	**86.50±1.06**
	3-way	
ISIC 2018	3-shot	**75.33±1.20**
	5-shot	**79.016±1.04**
	10-shot	**82.05±1.02**

Table 5.9: *ISIC train results*

Όπως φαίνεται και στον 5.10 πίνακα, το accuracy για την υλοποίηση με Transformer φαίνεται να επιτυγχάνει τα μεγαλύτερα ποσοστά στα tasks του 2-way,3-way 10-shot ενώ στα υπόλοιπα tasks, όπου το σύνολο των δειγμάτων προς εκπαίδευση γίνεται αρκετά μικρότερο, το accuracy του μοντέλου φθίνει με μεγαλύτερο ρυθμό από οτι αυτό του PFEMED. Αυτό πιθανότατα οφείλεται στην ανάγκη του Transformer για περισσότερα δεδομένα ακόμα και στο στάδιο του fine-tuning.

Το τελευταίο σύνολο δεδομένων πάνω στο οποίο πραγματοποιήθηκε η εκπαίδευση των διαφόρων μοντέλων και εξηχθησαν προβλέψεις ήταν το BreakHis. Το συγκεκριμένο dataset αποτελείται από τέσσερα υποσύνολα τα οποία περιέχουν εικόνες που αφορούν δείγματα απο μαστογραφίες. Οι διαφορές των δειγμάτων είναι οτι χρησιμοποιούνται διαφορετικά

Dataset	Few-shot setting	Transfer learning	Meta-Med				PFEMED	Few-TURE
			Normal Augmen-tation	Cutout	Mixup	CutMix		
				2-way				
ISIC 2018	3 shot	66.88	72.75	70.37	75.37	73.25	81.69	**73.80±1.41**
	5 shot	73.88	75.62	77.62	78.25	76.87	83.87	**81.70±1.22**
	10 shot	80.38	81.37	81.87	84.25	80.62	85.14	**84.18±1.13**
				3-way				
ISIC 2018	3 shot	55.67	54.83	55.50	58.50	58.66	66.94	**63.86±0.78**
	5 shot	59.67	59.33	65.41	61.25	61.50	69.78	**66.70±0.67**
	10 shot	65.92	69.75	69.75	71.00	66.50	73.81	**71.33±0.66**

Πίνακας 5.10: *ISIC Accuracy for all models*

resolutions. Πιο συγκεκριμένα περιέχει εικόνες στις οποίες έχει εφαρμοστεί μεγέθυνση (x40,x100,x200,x400). Τα αποτελέσματα που προέκυψαν φαίνονται παρακάτω και για τα 4 υποσύνολα.

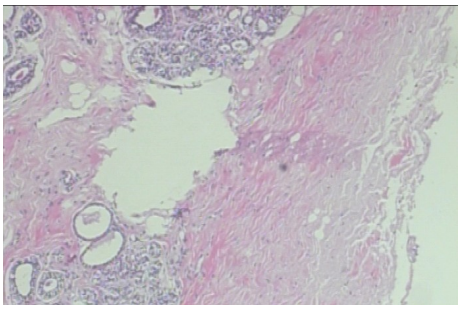
Train data results		
Dataset	Few-shot setting	Few TURE
	2-way	
BreakHis x 40	3-shot	**92.35±0.75**
	5-shot	**96.54±0.53**
	10-shot	**97.56±0.41**
BreakHis x 100	3-shot	**87.82±0.95**
	5-shot	**95.07±0.56**
	10-shot	**96.35±0.48**
BreakHis x 200	3-shot	**92.69±0.75**
	5-shot	**95.94±0.50**
	10-shot	**96.68±0.42**
BreakHis x 400	3-shot	**96.23±0.63**
	5-shot	**97.65±0.39**
	10-shot	**97.73±0.34**
	3-way	
BreakHis x 40	3-shot	**96.71±0.38**
	5-shot	**98.68±0.20**
	10-shot	**96.19±0.39**
BreakHis x 100	3-shot	**95.68±0.44**
	5-shot	**97.46±0.29**
	10-shot	**95.40±0.39**
BreakHis x 200	3-shot	**96.27±0.37**
	5-shot	**97.90±0.26**
	10-shot	**98.48±0.23**
BreakHis x 400	3-shot	**98.44±0.27**
	5-shot	**99.01±0.17**
	10-shot	**98.02±0.25**

Table 5.11: *BreakHis train results*

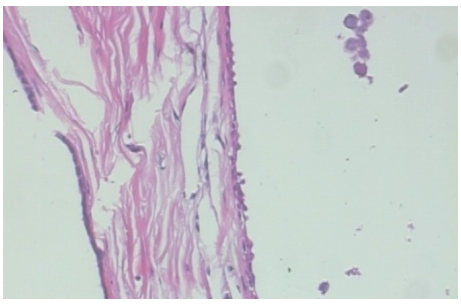
Όπως παρατηρείται απο τα 5.12 τα αποτελέσματα για τις μικρότερες μεγεθύνσεις των δεδομένων με την χρήση Transformers είναι κοντά σε αυτά του PFEMED και για τα 2-way αλλά και για τα 3-way προβλήματα. Ενώ όσο αυξάνεται η μεγέθυνση, η διαφορα στο accuracy αυτών των δύο μεθόδων ολο ενα γίνεται και μεγαλύτερη. Κάτι τέτοιο μπορεί να εξηγηθεί στην έλλειψη σημαντικών πληροφοριών στις εικόνες, καθώς όσο η μεγέθυνση γίνεται μεγαλύτερη τόσο πιο δύσκολη είναι η εξαγωγή διαφορετικών χαρακτηριστικών στα διάφορα μέρη της εικόνας. Κάτι τέτοιο φαίνεται και στις εικόνες (5.4,5.5,5.6,5.7), όπου για τα δείγματα με μεγάλη μεγέθυνση υπάρχουν σημαντικές περιοχές στην εικόνα που δεν παρέχουν χρήσιμες πληροφορίες για την ταξινόμηση. Ετσι ο χωρισμος των patches και η εφαρμογή self supervised learning με χρήση του Masked Image Modeling σε αυτές τις εικόνες μπορεί να μην δημιουργεί labels που εκφράζουν σημαντικές πληροφορίες με αποτέλεσμα το μοντέλο να αδυνατεί, να εκπαιδευτεί σε παραμέτρους οι οποίες μπορούν να προσαρμοστούν στο novel σύνολο δεδομένων.

Dataset	Few-shot setting	Transfer learning	Meta-Med				PFEMED	Few-TURE
			Normal Augmen-tation	Cutout	Mixup	CutMix		
				2-way				
BreakHis40X	3 shot	76.12	78.37	77.25	80.50	78.00	83.83	**80.02±1.30**
	5 shot	79.00	83.00	81.87	83.25	81.87	86.38	**80.71±0.86**
	10 shot	82.25	86.37	85.87	84.37	82.50	88.48	**87.13±0.73**
BreakHis100X	3 shot	78.50	78.75	77.12	78.25	79.62	82.16	**70.62±1.18**
	5 shot	78.62	81.38	79.75	82.88	84.12	85.28	**77.38±0.93**
	10 shot	82.00	83.88	83.37	84.75	80.88	86.90	**84.30±0.80**
BreakHis200X	3 shot	73.62	74.87	77.00	76.62	77.25	82.71	**72.29±1.29**
	5 shot	78.00	79.75	79.75	81.25	81.75	85.18	**74.05±1.05**
	10 shot	79.75	83.75	84.87	84.75	86.12	86.59	**82.05±0.97**
BreakHis400X	3 shot	72.88	74.75	68.25	74.25	75.25	79.09	**72.87±1.22**
	5 shot	74.62	79.62	76.75	81.37	77.12	82.03	**76.24±1.09**
	10 shot	77.50	83.00	81.37	82.75	81.00	86.63	**82.97±0.96**
				3-way				
BreakHis40X	3 shot	63.17	70.08	67.33	70.00	72.08	72.72	**68.06±0.90**
	5 shot	65.08	73.50	69.33	74.00	74.25	76.56	**72.33±0.78**
	10 shot	69.50	77.66	74.41	78.61	72.33	79.45	**78.22±0.65**
BreakHis100X	3 shot	62.42	63.08	61.66	63.33	63.92	69.21	**58.83±1.03**
	5 shot	64.33	66.42	63.41	68.83	66.83	75.04	**65.40±0.85**
	10 shot	66.50	74.08	68.08	72.42	75.33	78.93	**75.77±0.70**
BreakHis200X	3 shot	62.50	61.50	62.41	64.33	65.75	71.26	**56.60±0.86**
	5 shot	64.92	68.00	68.41	66.50	70.58	75.94	**62.32±0.79**
	10 shot	70.08	74.16	73.91	75.00	73.08	79.01	**73.55±0.74**
BreakHis400X	3 shot	56.75	64.66	61.33	64.41	65.25	65.42	**58.33±0.88**
	5 shot	61.75	67.75	66.50	68.25	67.66	69.14	**61.80±0.79**
	10 shot	63.50	74.25	74.44	73.91	68.25	74.53	**70.75±0.77**

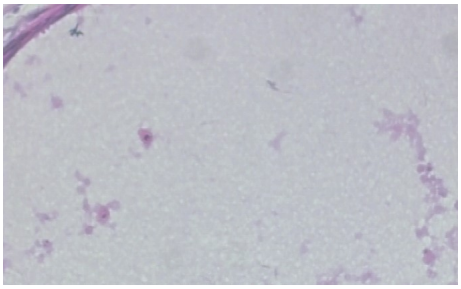
Πίνακας 5.12: *BreakHis Accuracy for all models*



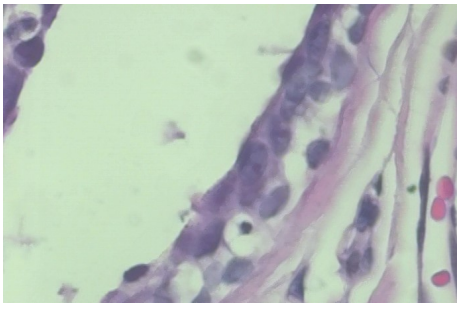
Σχήμα 5.4: *BreakHis x40 samples*



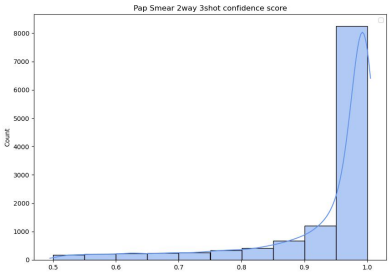
Σχήμα 5.5: *BreakHis x100 samples*



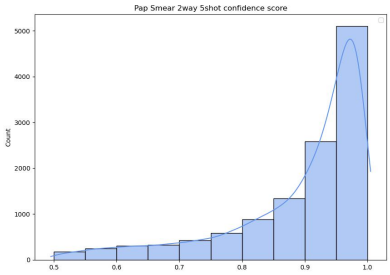
Σχήμα 5.6: *BreakHis x200 samples*



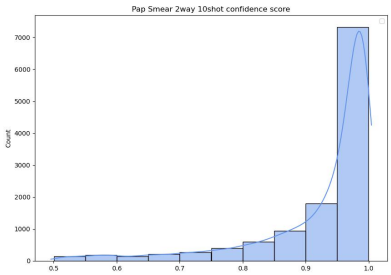
Σχήμα 5.7: *BreakHis x400 samples*



Σχήμα 5.8: *Pap Smear 2way 3shot*



Σχήμα 5.9: *Pap Smear 2way 5shot*



Σχήμα 5.10: *Pap Smear 2way 10shot*

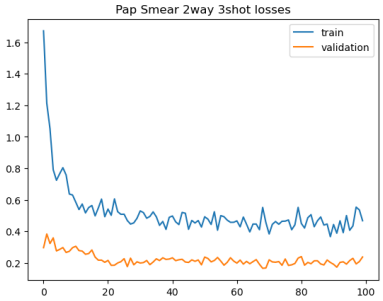
Σχήμα 5.11: *Confidence score of Pap Smear in 2way tasks*

Στο 5.11 απεικονίζονται οι πιθανότητες με τις οποίες κατηγοριοποίησε το μοντέλο του Vision Transformer τα δεδομένα του novel υποσυνόλου για τα 2way tasks. Απο την γραφική παράσταση μπορεί να παρατηρηθεί ότι το μοντέλο κατατάσει τις εικόνες με αρκετα μεγάλο confidence αφου το μεγαλύτερο πλήθος των τιμών κυμαίνεται απο 0.8-1.0. Κάτι τέτοιο μπορεί να μας οδηγήσει στο συμπέρασμα ότι το μοντέλο επιτυγχάνει αρκετά εύρωστα αποτελέσματα και μπορεί να γενικεύσει σε αρκετά καλό βαθμό σε ένα σύνολο από novel δεδομένα.

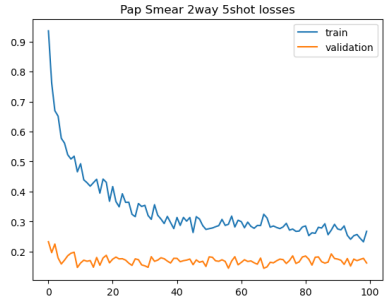
Στο 5.15 παρουσιάζονται γραφικές παραστάσεις στις οποίες απεικονίζεται το loss κατα την διάρκεια του fine-tuning για τα διάφορα 2way tasks απο τις οποίες μπορούμε να παρατηρήσουμε οτι το μοντέλο συγκλίνει αρκετά γρήγορα στο ελάχιστο loss, το οποίο φαίνεται να συμβαίνει γύρω στην εποχή 20.

Στο 5.19 παρουσιάζονται τα accuracies για τα διάφορα 2way tasks, όπου και σε αυτήν την περίπτωση όπως και στο loss φαίνεται οτι οι τιμές συγκλίνουν αρκετά γρήγορα, κοντά στην εποχή 20.

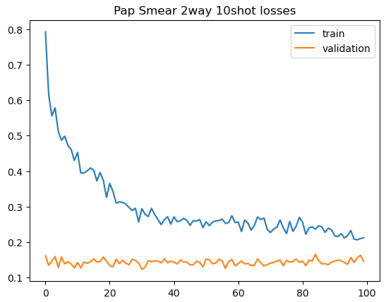
Στο 5.23 απεικονίζονται οι πιθανότητες με τις οποίες το μοντέλο του Vision Transformer κατηγοριοποίησε τα δεδομένα του novel υποσυνόλου των δεδομένων για τα 3way tasks. Απο το συγκεκριμένο ιστόγραμμα φαίνεται το μοντέλο να επιτυγχάνει αρκετα υψηλά confidence scores, πράγμα το οποία μας οδηγεί στα ίδια συμπεράσματα που αναλύθηκαν και για τα ιστογράμματα των 2way tasks.



Σχήμα 5.12: *Pap Smear 2way 3shot loss*

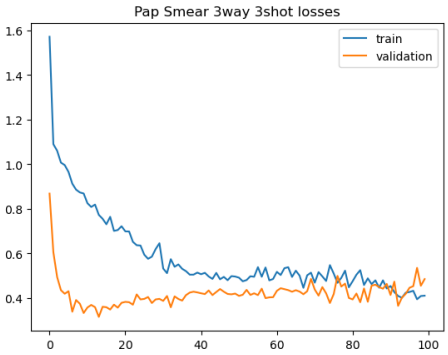


Σχήμα 5.13: *Pap Smear 2way 5shot loss*

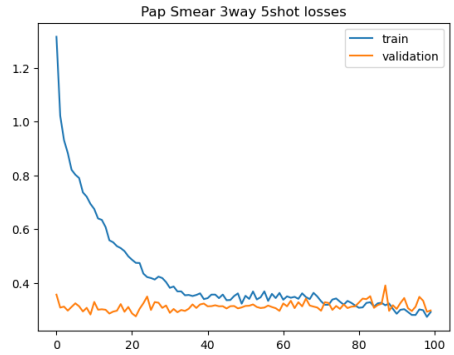


Σχήμα 5.14: *Pap Smear 2way 10shot loss*

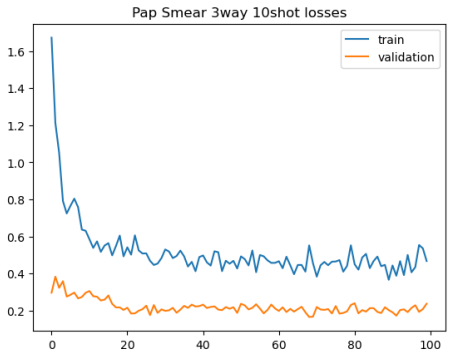
Σχήμα 5.15: *Losses of Pap Smear in 2way tasks*



Σχήμα 5.24: *Pap Smear 3way 3shot loss*

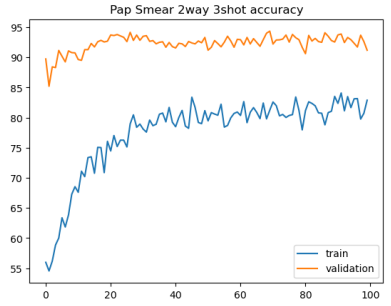


Σχήμα 5.25: *Pap Smear 3way 5shot loss*

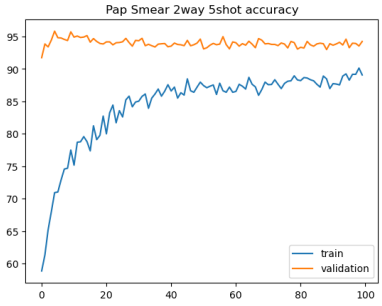


Σχήμα 5.26: *Pap Smear 3way 10shot loss*

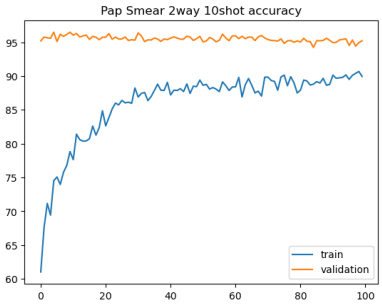
Σχήμα 5.27: *Losses of Pap Smear in 3way tasks*



Σχήμα 5.16: *Pap Smear 2way 3shot accuracy*

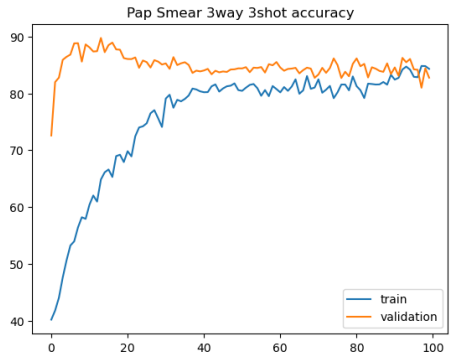


Σχήμα 5.17: *Pap Smear 2way 5shot accuracy*

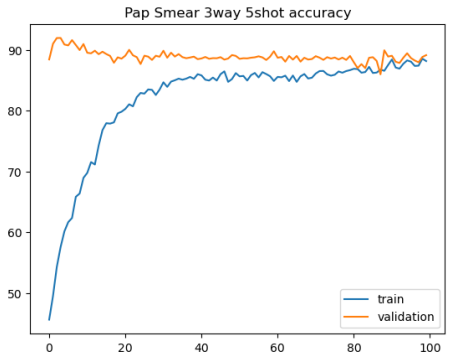


Σχήμα 5.18: *Pap Smear 2way 10shot accuracy*

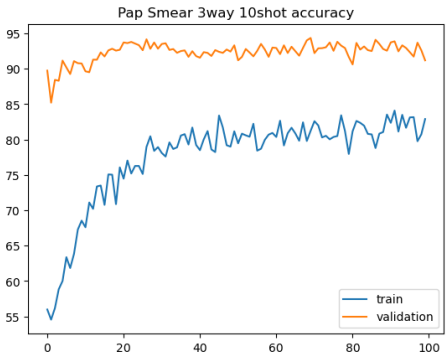
Σχήμα 5.19: *Accuracy of Pap Smear in 2way tasks*



Σχήμα 5.28: *Pap Smear 3way 3shot accuracy*

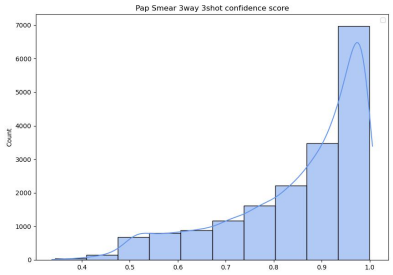


Σχήμα 5.29: *Pap Smear 3way 5shot accuracy*

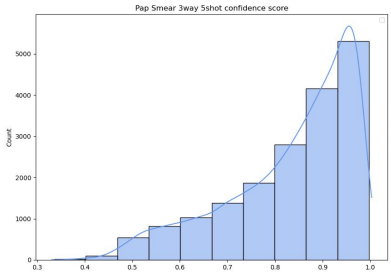


Σχήμα 5.30: *Pap Smear 3way 10shot accuracy*

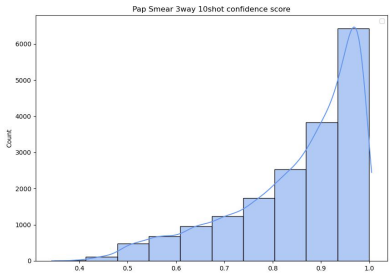
Σχήμα 5.31: *Accuracy of Pap Smear in 3way tasks*



Σχήμα 5.20: *Pap Smear 3way 3shot*



Σχήμα 5.21: *Pap Smear 3way 5shot*

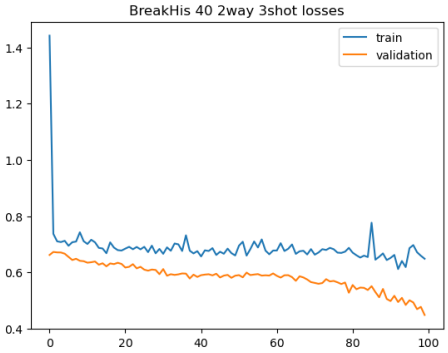


Σχήμα 5.22: *Pap Smear 3way 10shot*

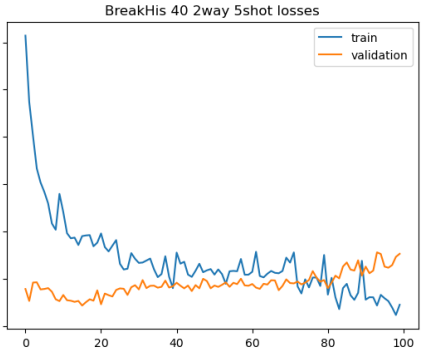
Σχήμα 5.23: *Confidence score of Pap Smear in 3way tasks*

Στα 5.27, 5.31 απεικονίζεται το loss και το accuracy για τα 3way tasks στο σύνολο δεδομένων pap smear, όπου μπορούμε να παρατηρήσουμε παρόμοια συμπεριφορά για την εκπαίδευση με αυτήν που παρατηρήθηκε για τα 2 way tasks, δηλαδή το μοντέλο συγκλίνει κοντα στις μέγιστες τιμές του μετα απο μικρό αριθμό εποχών.

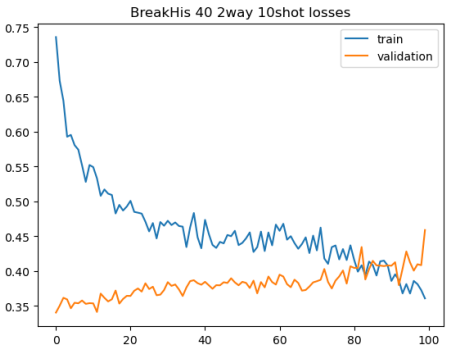
Απο τις 5.35, 5.43, 5.51, 5.59, 5.39, 5.47, 5.55, 5.63 γραφικές παραστάσεις μπορούμε να παρατηρήσουμε οτι για το σύνολο δεδομένων BreakHis 40 το μοντέλο κατα την διάρκεια της εκπαίδευσης συγκλίνει πιο γρήγορα στις βέλτιστες τιμές κυρίως για τα tasks τα οποία περιέχουν περισσότερα δείγματα, όπως τα 5shot, 10shot, ενω για τα tasks που περιέχουν λιγότερα δείγματα χρειάζονται περισσότερες εποχές εκπαίδευσης. Για τα Isic2018 δεδομένα παρατηρούνται μεγαλύτερες διακυμάνσεις στο loss, accuracy κατα την διαδικασία εκπαίδευσης. Επιπλεον, για τον μεγαλύτερο αριθμό εποχών κοντα στο 80-100 παρατηρείται οτι το μοντέλο οδηγείται σε overfitting καθώς το loss στα δεδομένα εκπαίδευσης συνεχίζει και αυξανεται αισθητά, ενω το loss στα δεδομένα επικύρωσης μειώνεται με σημαντικό ρυθμό.



Σχήμα 5.32: *BreakHis 40 2way 3shot loss*

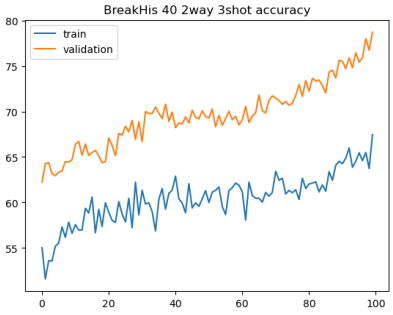


Σχήμα 5.33: *BreakHis 40 2way 5shot loss*

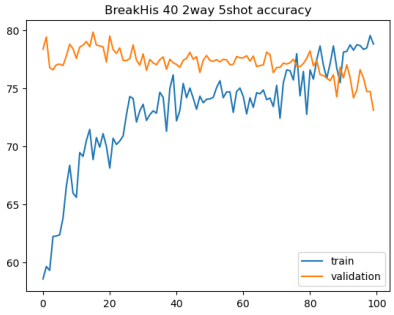


Σχήμα 5.34: *BreakHis 40 2way 10shot loss*

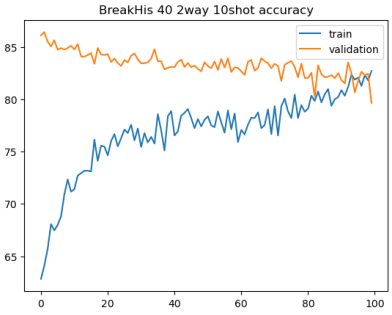
Σχήμα 5.35: *Losses of BreakHis 40 in 2way tasks*



Σχήμα 5.36: *BreakHis 40 2way 3shot accuracy*

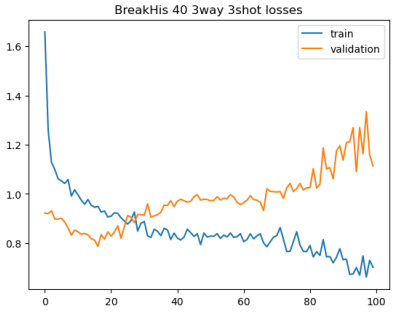


Σχήμα 5.37: *BreakHis 40 2way 5shot accuracy*

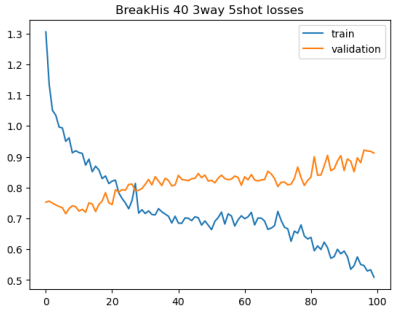


Σχήμα 5.38: *BreakHis 40 2way 10shot accuracy*

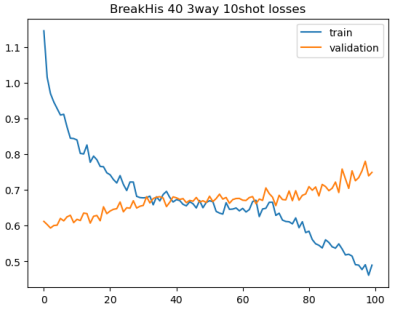
Σχήμα 5.39: *Accuracy of BreakHis 40 in 2way tasks*



Σχήμα 5.40: *BreakHis 40 3way 3shot loss*

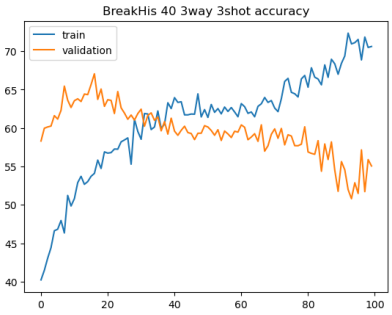


Σχήμα 5.41: *BreakHis 40 3way 5shot loss*

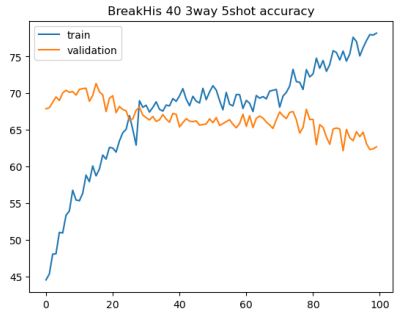


Σχήμα 5.42: *BreakHis 40 3way 10shot loss*

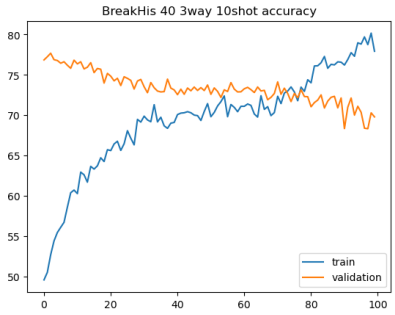
Σχήμα 5.43: *Losses of BreakHis 40 in 3way tasks*



Σχήμα 5.44: *BreakHis 40 3way 3shot accuracy*

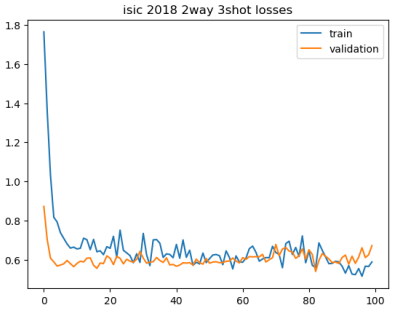


Σχήμα 5.45: *BreakHis 40 3way 5shot accuracy*

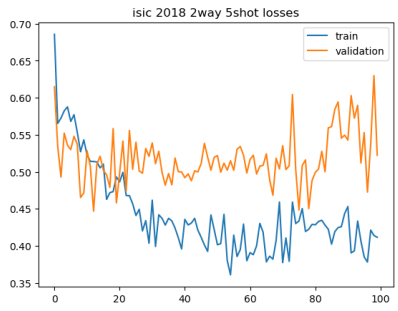


Σχήμα 5.46: *BreakHis 40 3way 10shot accuracy*

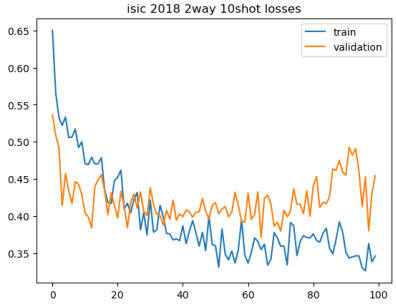
Σχήμα 5.47: *Accuracy of BreakHis 40 in 3way tasks*



Σχήμα 5.48: *ISIC 2018 2way 3shot loss*

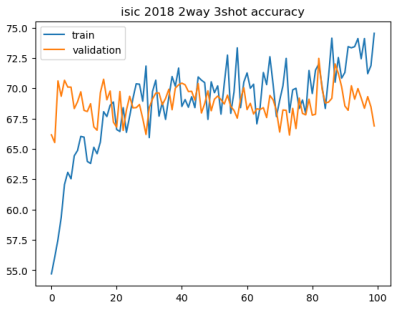


Σχήμα 5.49: *ISIC 2018 2way 5shot loss*

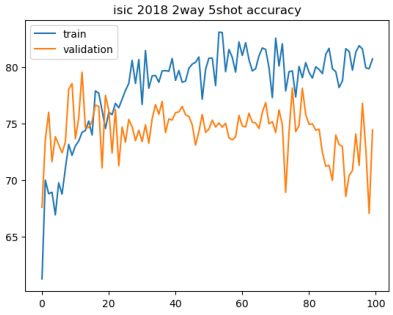


Σχήμα 5.50: *ISIC 2018 2way 10shot loss*

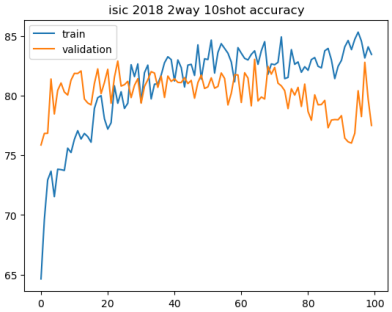
Σχήμα 5.51: *Losses of ISIC 2018 in 2way tasks*



Σχήμα 5.52: *ISIC 2018 3way 3shot accuracy*

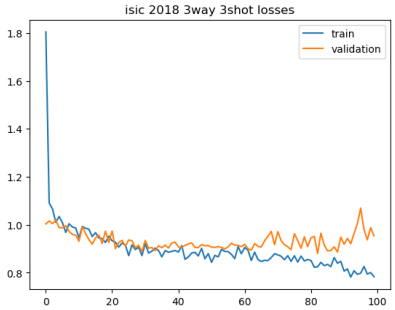


Σχήμα 5.53: *ISIC 2018 3way 5shot accuracy*

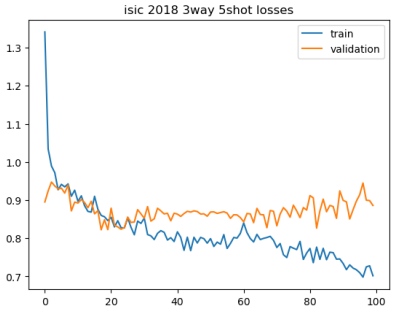


Σχήμα 5.54: *ISIC 2018 3way 10shot accuracy*

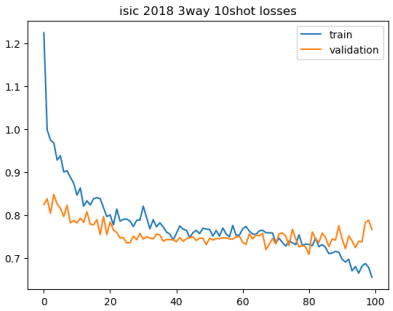
Σχήμα 5.55: *Accuracy of ISIC 2018 in 2way tasks*



Σχήμα 5.56: *ISIC 2018 3way 3shot loss*

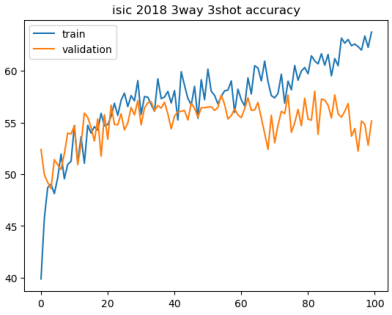


Σχήμα 5.57: *ISIC 2018 3way 5shot loss*

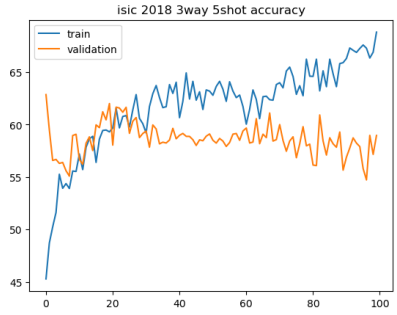


Σχήμα 5.58: *ISIC 2018 3way 3shot loss*

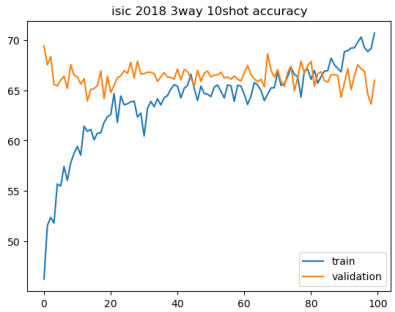
Σχήμα 5.59: *Losses of ISIC 2018 in 3way tasks*



Σχήμα 5.60: *ISIC 2018 3way 3shot accuracy*



Σχήμα 5.61: *ISIC 2018 3way 5shot accuracy*



Σχήμα 5.62: *ISIC 2018 3way 10shot accuracy*

Σχήμα 5.63: *Accuracy of ISIC 2018 in 3way tasks*

Μέρος IV

Επίλογος

Κεφάλαιο 6

Επίλογος

Στο συγκεκριμένο κεφάλαιο συνοψίζεται το περιεχόμενο το οποίο μελετήσαμε στην παρούσα διπλωματική εργασία και αναφέρονται τα γενικά συμπεράσματα τα οποία προέκυψαν απο την μελέτη μας. Επιπλεον προτείνονται μελλοντικές επεκτάσεις οι οποίες μπορούν να υιοθετηθούν για την ανάπτυξη των αποτελεσμάτων που προέκυψαν.

6.1 Συμπεράσματα

Στην συγκεκριμένη διπλωματική εργασια πραγματοποιήθηκε μελέτη διαφορετικών μοντέλων νευρωνικών δικτύων, με σκοπό την ορθή και αξιόπιστη ταξινόμηση βιοϊατρικών συνόλων δεδομένων τα οποία αποτελούνται από ανισοκατανεμημένο αριθμό δειγμάτων στις κλάσεις τους. Πιο συγκεκριμένα αυτά τα σύνολα δεδομένων χωρίστηκαν σε υποσύνολα D_{train}, D_{novel}, όπου το πρώτο περιήχε τις κλάσεις με τα περισσότερα δείγματα και το δεύτερο αυτές με τα λιγότερα δείγματα. Σκοπός ήταν η δημιουργία μοντέλων τα οποία με εκπαίδευση τους στο σύνολο D_{train} θα μπορούν να προσαρμοστούν και να επιτύχουν καλά αποτελέσματα για τα D_{novel}. Επειδή τα δεδομένα στα οποία πραγματοποιήθηκε η εκπαίδευση διέθεταν περιορισμένο αριθμό δειγμάτων, η διαδικασια εκπαίδευσης βασίστηκε στο few shot learning το οποίο αποτελεί μια μέθοδο που εκμεταλλεύεται μικρά σύνολα δειγμάτων, για να παράξει μοντέλα τα οποία πετυχαίνουν πιο γενικεύσιμα μοντέλα. Επίσης χρησιμοποιήθηκαν και διαφορετικοί μέθοδοι για την επίλυση αυτού του προβλήματος, μετά από μελέτη της βιβλιογραφίας παρατηρήθηκε οτι ο αλγόριθμός Reptile επιτυγχάνει πολυ καλα αποτελεσματα με την χρήση απλών δικτύων CNN. Επιπλέον, μελετήθηκε μια τεχνική συνδυασμού features απο δυο δίκτυα CNN τα οποία χρησιμοποιήθηκαν ως backbones για την παραγωγή πιο σύνθετων χαρακτηριστικών, τα οποία θα διαθέτουν περισσότερες πληροφορίες για την εξαγωγή χαρακτηριστικών.

Τέλος δοκιμάστηκαν δίκτυα Transformers τα οποία είχαν να αντιμετωπίσουν την πρόκληση των λίγων δεδομένων που διατίθονταν, το οποίο αποτελεί και το κύριο πρόβλημα της χρήσης δικτύων Transformers σε βιοιατρικά προβλήματα. Παρ' ολα αυτά με την χρήση δικτύων Vit small και με την εφαρμογή ενος τρόπου mapping των δεδομένων επιτεύχθηκαν αποτελέσματα ταξινόμησης τα οποία είναι κοντα στο state of the art κυρίως για το σύνολο δεδομένων Pap Smear. Επιπλέον, για το συγκεκριμένο σύνολο δεδομένων παρατηρήθηκε οτι οι προβλέψεις του μοντέλου είναι πολύ ισχυρές, καθώς πετυχαίνουν αρκετά υψηλά confidence scores.

Σαν τελικο συμπέρασμα της μελέτης αυτής της διπλωματικής εργασίας έγινε αντιλυπτο το γεγονος οτι παρολο που τα δίκτυα Transformers αντιμετωπίζουν πολλές προκλήσεις κατα την εκπαίδευση τους στα βιοϊατρικά δεδομένα κυρίως λόγω του περιορισμένου αριθμού δειγμάτων, μέσω εφαρμογής κατάλληλων τεχνικών εκπαίδευσης είναι δυνατόν το μοντέλο που βασίζεται σε Transformers να επιτύχει εξίσου καλά αποτελέσματα με τεχνικές που συνδυάζουν CNN μοντέλα και αποτελούν τις πιο σύνηθες μεθόδους επίλυσης προβλημάτων στον τομέα του medical imaging.

6.2 Μελλοντικές Επεκτάσεις

Τα αποτελέσματα που επιτύχαμε με την χρηση των Vision Transformers αν και είναι συγκρίσιμα με αυτά του state of the art, οπου χρησιμοποιούνται CNN δίκτυα δεν καταφέρνουν να επιτύχουν συγκρίσιμα αποτελέσματα σε ολα τα σύνολα δεδομένων και για όλα τα Few shot tasks που δοκιμάσαμε. Κάτι τέτοιο μας οδηγεί στην επιθυμια ενίσχυσης της τεχνικής των Vision Transformer που χρησιμοποιήθηκε. Πιο συγκεκριμένα μια ιδεα για την ενίσχυση των αποτελεσμάτων είναι η χρήση υβριδικών συστημάτων CNN-ViT Transformers, αυτή η ιδέα προέκυψε από την παρατήρηση της καλή απόδοσης των CNN δικτύων στα συγκεκριμένα αποτελέσματα αλλά και της ευρωστότητας των αποτελεσμάτων τα οποία εξάγουν[2], αυτό μας οδηγεί στο συμπέρασμα οτι τα συγκεκριμένα δίκτυα εξάγουν χάρτες χαρακτηριστικών, οι οποιοί περιέχουν πολλές χρήσιμες πληροφορίες σχετικά με τα δεδομένα. Έτσι σκοπός μας είναι η εισαγωγή αυτών των χαρτών χαρακτηριστικών στο σύστημα του ViT Transformer και η πραγματοποίηση της κατηγοριοποίησης με βάση τα embeddings που θα προκύψουν από αυτόν τον χάρτη χαρακτηριστικών.

Πιο συγκεκριμένα εμπνευσμένοι από το [3] επιθυμουμε να χρησιμοποιήσουμε το δίκτυο [4], για να εξάγουμε πιο σύνθετα χαρακτηριστικά τα οποία στην συνέχεια θα τα τροφοδοτήσουμε στο δίκτυο ViT small και απο εκεί θα εξάγουμε embeddings τα οποία θα τα χρησιμοποιήσουμε στην συνέχεια για την κατηγοριοποίηση των δεδομένων.

Βιβλιογραφία

[1] L. Pratt S. Thrun. *Learning to learn: introduction and overview, in: Learning to Learn pp. 3-17*. Springer, 1998.

[2] Rishav Singh, Vandana Bharti, Vishal Purohit, Abhinav Kumar, Amit Kumar Singh και Sanjay Kumar Singh. *MetaMed: Few-shot medical image classification using gradient-based meta-learning*. *Pattern Recognition*, 120:108111, 2021.

[3] Zhiyong Dai, Jianjun Yi, Lei Yan, Qingwen Xu, Liang Hu, Qi Zhang, Jiahui Li και Guoqiang Wang. *PFEMed: Few-shot medical image classification using prior guided feature enhancement*. *Pattern Recognition*, 134:109108, 2023.

[4] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li και Jiaya Jia. *Prior Guided Feature Enrichment Network for Few-Shot Segmentation*, 2020.

[5] Yuqing Hu, Vincent Gripon και Stéphane Pateux. *Leveraging the Feature Distribution in Transfer-based Few-Shot Learning*, 2021.

[6] Markus Hiller, Rongkai Ma, Mehrtash Harandi και Tom Drummond. *Rethinking Generalization in Few-Shot Classification*, 2022.

[7] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille και Tao Kong. *iBOT: Image BERT Pre-Training with Online Tokenizer*, 2022.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser και Illia Polosukhin. *Attention Is All You Need*, 2023.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit και Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2021.

[10] *Explaining Attention mechanisms*. https://towardsdatascience.com/all-you-need-to-know-about-attention-and-transformers-in-depth-understanding-part-1-552f0b41d021. Ημερομηνία πρόσβασης: 29-09-2023.

[11] Qiong Liu και Ying Wu. *Supervised Learning*. 2012.

[12] Yanbei Chen, Massimiliano Mancini, Xiatian Zhu και Zeynep Akata. *Semi-Supervised and Unsupervised Deep Visual Learning: A Survey*, 2022.

[13] Linus Ericsson, Henry Gouk, Chen Change Loy και Timothy M. Hospedales. *Self-Supervised Representation Learning: Introduction, advances, and challenges*. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

[14] Zewen Li, Wenjie Yang, Shouheng Peng και Fan Liu. *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*, 2020.

[15] Hossein Gholamalinezhad και Hossein Khosravi. *Pooling Methods in Deep Neural Networks, a Review*, 2020.

[16] Abien Fred Agarap. *Deep Learning using Rectified Linear Units (ReLU)*, 2019.

[17] Y. Lecun, L. Bottou, Y. Bengio και P. Haffner. *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] Alex Krizhevsky, Ilya Sutskever και Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks*. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[19] Karen Simonyan και Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition*. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke και Andrew Rabinovich. *Going deeper with convolutions*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep residual learning for image recognition*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Andrew Brock, Soham De, Samuel L. Smith και Karen Simonyan. *High-Performance Large-Scale Image Recognition Without Normalization*, 2021.

[23] Zhuang Liu, Hanzi Mao, Chao Yuan Wu, Christoph Feichtenhofer, Trevor Darrell και Saining Xie. *A ConvNet for the 2020s*, 2022.

[24] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yuwen Xiong και Song Han. *ResNeSt: Split-Attention Networks*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] Sergey Ioffe και Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, 2015.

[26] *Transformer overview*. https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/. Ημερομηνία πρόσβασης: 30-09-2023.

[27] Jacob Devlin, Ming Wei Chang, Kenton Lee και Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019.

[28] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever και Dario Amodei. *Language Models are Few-Shot Learners*, 2020.

[29] Dan Hendrycks και Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*, 2023.

[30] Chen Sun, Abhinav Shrivastava, Saurabh Singh και Abhinav Gupta. *Revisiting Unreasonable Effectiveness of Data in Deep Learning Era*, 2017.

[31] Jean Baptiste Cordonnier, Andreas Loukas και Martin Jaggi. *Multi-Head Attention: Collaborate Instead of Concatenate*, 2021.

[32] Alhassan Mumuni και Fuseini Mumuni. *Data augmentation: A comprehensive survey of modern approaches*. *Array*, 16:100258, 2022.

[33] Luis Perez και Jason Wang. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*, 2017.

[34] T. Dutta, A. Singh και S. Biswas. *Styleguide: Zero-Shot Sketch-Based Image Retrieval Using Style-Guided Image Generation*. *IEEE Transactions on Multimedia*, 2020.

[35] Archit Parnami και Minwoo Lee. *Learning from Few Examples: A Summary of Approaches to Few-Shot Learning*, 2022.

[36] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong και Qing He. *A Comprehensive Survey on Transfer Learning*, 2020.

[37] Timothy Hospedales, Antreas Antoniou, Paul Micaelli και Amos Storkey. *Meta-Learning in Neural Networks: A Survey*, 2020.

[38] M. Shorfuzzaman και M.S. Hossain. *Metacovid: A Siamese Neural Network Framework with Contrastive Loss for N-Shot Diagnosis of COVID-19 Patients*. *Pattern Recognition*, 113:107700, 2021.

[39] Rongkai Ma, Pengfei Fang, Gil Avraham, Yan Zuo, Tom Drummond και Mehrtash Harandi. *Learning Instance and Task-Aware Dynamic Kernels for Few-Shot Learning*. *arXiv preprint arXiv:2112.03494*, 2021.

[40] Rongkai Ma, Pengfei Fang, Tom Drummond και Mehrtash Harandi. *Adaptive Poincaré Point to Set Distance for Few-Shot Classification*. *arXiv preprint arXiv:2112.01719*, 2021.

[41] Christian Simon, Piotr Koniusz, Richard Nock και Mehrtash Harandi. *Adaptive Subspaces for Few-Shot Learning*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, σελίδες 4136–4145, 2020.

[42] Jake Snell, Kevin Swersky και Richard S. Zemel. *Prototypical Networks for Few-shot Learning*, 2017.

[43] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu και Daan Wierstra. *Matching Networks for One Shot Learning*, 2017.

[44] Han Jia Ye, Hexiang Hu, De Chuan Zhan και Fei Sha. *Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions*, 2021.

[45] Chi Zhang, Yujun Cai, Guosheng Lin και Chunhua Shen. *DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover's Distance and Structured Classifiers*. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] Chelsea Finn, Pieter Abbeel και Sergey Levine. *Model-agnostic meta-learning for fast adaptation of deep networks*. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, σελίδες 1126–1135. JMLR. org, 2017.

[47] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran και Stefano Soatto. *Meta-Learning with Differentiable Convex Optimization*, 2019.

[48] Alex Nichol, Joshua Achiam και John Schulman. *On First-Order Meta-Learning Algorithms*, 2018.

[49] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero και Raia Hadsell. *Meta-Learning with Latent Embedding Optimization*, 2019.

[50] Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann και Shimon Whiteson. *Fast Context Adaptation via Meta-Learning*, 2019.

[51] J. Jantzen, J. Norup, G. Dounias και B. Bjerregaard. *Pap-smear benchmark data for pattern classification*. *Nature Inspired Smart Information Systems (NiSIS)*, 2005.

[52] F. A. Spanhol, L. S. Oliveira, C. Petitjean και L. Heutte. *A dataset for breast cancer histopathological image classification*. *IEEE Trans. Biomed. Eng.*, 63(7):1455–1462, 2015.

[53] J. Zou, X. Ma, C. Zhong και Y. Zhang. *Dermoscopic Image Analysis for ISIC Challenge 2018*. *arXiv preprint arXiv:1807.08948*, 2018.

[54] Terrance DeVries και Graham W. Taylor. *Improved Regularization of Convolutional Neural Networks with Cutout*. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, σελίδες 1949–1958. JMLR. org, 2017.

[55] Sangdoo Yun, Dongyoon Han, Seong Joon Oh και Sanghyuk Chun. *CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features*. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[56] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin και David Lopez-Paz. *mixup: Beyond Empirical Risk Minimization*. *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[57] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song και Jacob Steinhardt. *PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures*, 2022.

[58] Sanghyuk Chun, Seong Joon Oh, Sangdoo Yun, Dongyoon Han, Junsuk Choe και Youngjoon Yoo. *An Empirical Evaluation on Robustness and Uncertainty of Regularization Methods*, 2020.

[59] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura και Yutaka Satoh. *Pre-training without Natural Images*, 2021.

[60] Kodai Nakashima, Hirokatsu Kataoka, Asato Matsumoto, Kenji Iwata και Nakamasa Inoue. *Can Vision Transformers Learn without Natural Images?*, 2021.

[61] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann και Wieland Brendel. *ImageNet-Trained CNNs Are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness*. *International Conference on Learning Representations (ICLR)*, 2019.

[62] Dan Hendrycks και Thomas Dietterich. *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations*, 2019.

[63] Eric Mintun, Alexander Kirillov και Saining Xie. *On Interaction Between Augmentations and Corruptions in Natural Corruption Robustness*, 2021.

[64] Chuan Guo, Geoff Pleiss, Yu Sun και Kilian Q. Weinberger. *On Calibration of Modern Neural Networks*, 2017.

[65] Khanh Nguyen και Brendan O'Connor. *Posterior calibration and exploratory analysis for natural language processing models*, 2015.

[66] Dan Hendrycks, Kimin Lee και Mantas Mazeika. *Using Pre-Training Can Improve Model Robustness and Uncertainty*, 2019.

[67] Dan Hendrycks και Kevin Gimpel. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*, 2018.

[68] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville και Yoshua Bengio. *Generative Adversarial Networks*, 2014.

[69] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer και Balaji Lakshminarayanan. *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*, 2020.

[70] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba και Koichi Kise. *Shakedrop Regularization for Deep Residual Learning*. *IEEE Access*, 7:186126–186136, 2019.

[71] Q. Zhang, H. Wang, H. Lu, D. Won και S.W. Yoon. *Medical Image Synthesis with Generative Adversarial Networks for Tissue Recognition*. *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, σελίδες 199–207. IEEE, 2018.

[72] Chen Chen, Zeju Li, Cheng Ouyang, Matt Sinclair, Wenjia Bai και Daniel Rueckert. *MaxStyle: Adversarial Style Composition for Robust Medical Image Segmentation*, 2022.

[73] Chen Chen, Kerstin Hammernik, Cheng Ouyang, Chen Qin, Wenjia Bai και Daniel Rueckert. *Cooperative Training and Latent Space Data Augmentation for Robust Medical Image Segmentation*, 2021.

[74] Léon Bottou, Frank E Curtis και Jorge Nocedal. *On the Convergence of Stochastic Gradient Descent for L2-Regularized Linear Regression*. *International Conference on Algorithmic Learning Theory*, σελίδες 3–19. Springer, 2018.

[75] Alex Nichol, Joshua Achiam και John Schulman. *Reptile: a Scalable Meta-Learning Algorithm*. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[76] S. Hochreiter, A. S. Younger και P. R. Conwell. *Learning to Learn Using Gradient Descent*. *International Conference on Artificial Neural Networks*, σελίδες 87–94. Springer, 2001.

[77] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li και L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, σελίδες 248–255. IEEE, 2009.

[78] N. Zhang, J. Donahue, R. Girshick και T. Darrell. *Part-Based R-CNNs for Fine-Grained Category Detection*. *European Conference on Computer Vision*, σελίδες 834–849. Springer, 2014.

[79] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa και Byron Boots. *One-Shot Learning for Semantic Segmentation*, 2017.

[80] Pinzhuo Tian, Zhangkai Wu, Lei Qi, Lei Wang, Yinghuan Shi και Yang Gao. *Differentiable Meta-Learning Model for Few-Shot Semantic Segmentation*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12087–12094, 2020.

[81] N. Dong και E. P. Xing. *Few-Shot Semantic Segmentation with Prototype Learning*. *British Machine Vision Conference (BMVC)*, 2018.

[82] K. Wang, J. Liew, Y. Zou, D. Zhou και J. Feng. *Panet: Few-Shot Image Semantic Segmentation with Prototype Alignment*. *International Conference on Computer Vision (ICCV)*, 2019.

[83] Y. Liu, X. Zhang, S. Zhang και X. He. *Part-Aware Prototype Network for Few-Shot Semantic Segmentation*. *European Conference on Computer Vision (ECCV)*, 2020.

[84] T. Hu, P. Yang, C. Zhang, G. Yu, Y. Mu και C. G. M. Snoek. *Attention-Based Multi-Context Guiding for Few-Shot Semantic Segmentation*. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.

[85] C. Zhang, G. Lin, F. Liu, R. Yao και C. Shen. *Canet: Class-Agnostic Segmentation Networks with Iterative Refinement and Attentive Few-Shot Learning*. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[86] Y. Yang, F. Meng, H. Li, Q. Wu, X. Xu και S. Chen. *A New Local Transformation Module for Few-Shot Segmentation*. *International Conference on Multimedia Modeling (MMM)*, 2020.

[87] S. Gairola, M. Hemani, A. Chopra και B. Krishnamurthy. *SimPropNet: Improved Similarity Propagation for Few-Shot Image Segmentation*. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[88] R. Azad, A. R. Fayjie, C. Kauffman, I. B. Ayed, M. Pedersoli και J. Dolz. *On the Texture Bias for Few-Shot CNN Segmentation*. *arXiv preprint arXiv:2007.00077*, 2020.

[89] Carl Doersch, Ankush Gupta και Andrew Zisserman. *CrossTransformers: spatially-aware few-shot transfer*, 2021.

[90] Hangbo Bao, Li Dong, Songhao Piao και Furu Wei. *BEiT: BERT Pre-Training of Image Transformers*, 2022.

[91] Sara Atito, Muhammad Awais και Josef Kittler. *SiT: Self-supervised vIsion Transformer*, 2022.

[92] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang και Jinqiao Wang. *MST: Masked Self-Supervised Transformer for Visual Representation*, 2021.

[93] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár και Ross Girshick. *Masked Autoencoders Are Scalable Vision Learners*, 2021.

[94] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell και Alexei A. Efros. *Context Encoders: Feature Learning by Inpainting*, 2016.

[95] Hao Tan, Jie Lei, Thomas Wolf και Mohit Bansal. *VIMPAC: Video Pre-Training via Masked Token Prediction and Contrastive Learning*, 2021.

[96] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski και Armand Joulin. *Emerging Properties in Self-Supervised Vision Transformers*, 2021.

[97] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg και Li Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge*, 2015.

[98] *BEiT Masking Generation*. https://github.com/microsoft/unilm/blob/master/beit/masking_generator.py. Ημερομηνία πρόσβασης: 25-09-2023.

Συντομογραφίες - Αρκτικόλεξα - Ακρωνύμια

βλπ	βλέπε
κ.λπ.	και λοιπά
κ.ο.κ	και ούτω καθεξής
CNN	Convolution Neural Networks
MIM	Masked Image Modeling

Απόδοση ξενόγλωσσων όρων

Απόδοση	Ξενόγλωσσος όρος
μετασχηματιστές	transformers
Μηχανισμοί προσοχής	Attention Mechanisms
Μετασχηματιστές όρασης	Vision Transformers
Σύνολο δεδομένων	Dataset
Οπισθοδιάδοση	Backpropagate
Εκπαιδευση	Train
Αυτο-Επιβλεπόμενη Μαθηση	Self-supervised learning