



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
“ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ”

**Χρήση Συνελικτικών Νευρωνικών Δικτύων για την Ανίχνευση COVID-19
σε Ακτινογραφικές Θώρακα**

Από Σπύρου Ευαγγελία-Μαρία
Υποβάλλεται
για την εκπλήρωση των προϋποθέσεων λήψης
Μεταπτυχιακού Διπλώματος
στην ειδίκευση «ΠΠΣ»
του ΠΜΣ “Πληροφορικά Συστήματα & Υπηρεσίες”
στο
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Φεβρουάριος 2024

Επιβλέπων: Μιχαήλ Φιλιππάκης
Ακαδημαϊκή Θέση: Καθηγητής

ΣΕΛΙΔΑ ΕΓΚΥΡΟΤΗΤΑΣ

Ονοματεπώνυμο Φοιτήτριας: Σπύρου Ευαγγελία-Μαρία

Τίτλος Μεταπτυχιακής Διπλωματικής Εργασίας: Χρήση Συνελικτικών Νευρωνικών Δικτύων για την Ανίχνευση COVID-19 σε Ακτινογραφικές Θώρακα

Η παρούσα Μεταπτυχιακή Διπλωματική Εργασία υποβάλλεται ως μερική εκπλήρωση των απαιτήσεων του Προγράμματος Μεταπτυχιακών Σπουδών “Πληροφοριακά Συστήματα & Υπηρεσίες” του Τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς και εγκρίθηκε στις [ημερομηνία έγκρισης] από τα μέλη της Εξεταστικής Επιτροπής.

Εξεταστική Επιτροπή

Επιβλέπων (Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Πειραιώς) Μιχαήλ Φιλιππάκης,
Καθηγητής

Μέλος Εξεταστικής Επιτροπής: Ανδρέας Μενύχτας , Επίκουρος Καθηγητής

Μέλος Εξεταστικής Επιτροπής: Δημοσθένης Κυριαζής, Καθηγητής

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΑΥΘΕΝΤΙΚΟΤΗΤΑΣ

Η Σπύρου Ευαγγελία-Μαρία, γνωρίζοντας τις συνέπειες της λογοκλοπής, δηλώνω υπεύθυνα ότι η παρούσα εργασία με τίτλο «Χρήση Συνελικτικών Νευρωνικών Δικτύων για την Ανίχνευση COVID-19 σε Ακτινογραφικές Θώρακα», αποτελεί προϊόν αυστηρά προσωπικής εργασίας και όλες οι πηγές που έχω χρησιμοποιήσει, έχουν δηλωθεί κατάλληλα στις βιβλιογραφικές παραπομπές και αναφορές. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Επιπλέον δηλώνω υπεύθυνα ότι η συγκεκριμένη Μεταπτυχιακή Διπλωματική Εργασία έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου. Σε κάθε περίπτωση, αναληθούς ή ανακριβούς δηλώσεως, υπόκειμαι στις συνέπειες που προβλέπονται τις διατάξεις που προβλέπει η Ελληνική και Κοινοτική Νομοθεσία περί πνευματικής ιδιοκτησίας.

Η ΔΗΛΟΥΣΑ

Ονοματεπώνυμο: Σπύρου Ευαγγελία-Μαρία

Αριθμός Μητρώου: ME2056

Υπογραφή

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου κύριο Μιχαήλ Φιλιππάκη για την καθοδήγηση που μου παρείχε για την υλοποίηση της παρούσας διπλωματικής εργασίας και για την άψογη συνεργασία μας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου για την στήριξη που μου παρείχαν κατά την διάρκεια των μεταπτυχιακών μου σπουδών.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	9
1.1 Ανάλυση του προβλήματος	9
1.2 Δομή της Διπλωματικής εργασίας	11
ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	12
2.1 Τεχνητή Νοημοσύνη.....	12
2.2 Μηχανική Μάθηση.....	13
2.3 Βαθιά Μάθηση.....	15
2.4 Ταξινόμηση εικόνας.....	16
2.5 Νευρωνικά Δίκτυα	16
2.5.1 Βιολογικά Νευρωνικά Δίκτυα	17
2.5.2 Τεχνητά Νευρωνικά Δίκτυα.....	18
2.5.3 Συνελικτικά Νευρωνικά Δίκτυα.....	19
2.6 Ανάλυση Αρχιτεκτονικής των ΣΝΔ	20
2.7 Residual Networks (ResNets)	24
2.8 Μεταφορά Μάθησης(Transfer Learning)	26
2.9 Συναρτήσεις Ενεργοποιήσεις	27
2.10 Συναρτήσεις Κόστους.....	28
2.11 Βελτιστοποιητής (Optimizer).....	30
2.12 Αξιολόγηση της Απόδοσης του Μοντέλου	32
2.13 Υπερπροσαρμογή & Υποπροσαρμογή	35
2.14 Μέθοδος Οπισθοδιάδοσης (Backpropagation).....	37
2.15 Εξαφάνιση και Εκτόξευση της κλίσης	38
ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ	40
3.1 Βιβλιοθήκες και Εργαλεία	40

3.1.1	<i>Python</i>	40
3.1.2	<i>TensorFlow</i>	40
3.1.3	<i>Pandas</i>	41
3.1.4	<i>Scikit-learn</i>	41
3.1.5	<i>Matplotlib</i>	41
3.1.6	<i>NumPy</i>	41
3.1.7	<i>Keras</i>	42
3.1.8	<i>Google Colab Pro</i>	42
3.2	Σύνολο δεδομένων	42
3.3	Αρχιτεκτονικός Σχεδιασμός Μοντέλων	46
3.4	Συνελικτικό Νευρωνικό δίκτυο από την αρχή (CNN from scratch).....	48
3.4.1	<i>Αξιολόγηση Convolutional Neural Network from Scratch</i>	50
3.5	ResNet-50 Model	57
3.5.1	<i>Αξιολόγηση μοντέλου ResNet-50</i>	59
ΚΕΦΑΛΑΙΟ 4		66
4.1	Συμπεράσματα	66
4.2	Μελλοντικές εργασίες και βελτιστοποιήσεις.....	67
ΒΙΒΛΙΟΓΡΑΦΙΑ		68
ΠΑΡΑΡΤΗΜΑ I		70
ΠΑΡΑΡΤΗΜΑ II		73
ΠΑΡΑΡΤΗΜΑ III		77

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1 : Η δομή ενός βιολογικού νευρώνα	17
Εικόνα 2 : Βασικό Νευρωνικό Δίκτυο	19
Εικόνα 3 : Τι βλέπει ο άνθρωπος vs τι βλέπει ο υπολογιστής	20
Εικόνα 4 : Διαγραμματική απεικόνιση ενός βασικού CNN	21
Εικόνα 5 : Max Pooling and Average Pooling	23
Εικόνα 6 : Αρχιτεκτονική ResNet-50	25
Εικόνα 7 : Πίνακας Σφαλμάτων	33
Εικόνα 8 : Οπτική Απεικόνιση Υπερπροσαρμογής & Υποπροσαρμογής	37
Εικόνα 9 : Κατανομή των κλάσεων στα δεδομένα εκπαίδευσης και δοκιμής	43
Εικόνα 10 : Ακτινογραφίες ασθενών που βρέθηκαν θετικοί στο Covid-19	44
Εικόνα 11 : Ακτινογραφίες ασθενών που βρέθηκαν αρνητικοί στο Covid-19	44
Εικόνα 12 : Κατανομή των κλάσεων στα δεδομένα εκπαίδευσης , δοκιμής και επικύρωσης	45
Εικόνα 13 : CNN Model from Scratch	48
Εικόνα 14 : Training VS Validation Accuracy (CNN from scratch)	51
Εικόνα 15 : Απώλεια Εκπαίδευσης και Επικύρωσης (CNN from scratch)	52
Εικόνα 16 : Ρυθμός Μάθησης ανά Εποχή (CNN from scratch)	53
Εικόνα 17 : Μετρήσεις επιδόσεων στο σύνολο δεδομένων επικύρωσης (CNN from scratch)	54
Εικόνα 18 : Πίνακας Σφαλμάτων (CNN from Scratch)	56
Εικόνα 19 : Αρχιτεκτονική ResNet-50	57
Εικόνα 20 : Training VS Validation Accuracy with frozen layers (ResNet-50)	59
Εικόνα 21 : Training VS Validation Loss with frozen layers (ResNet-50)	60
Εικόνα 22 : Training VS Validation Accuracy with unfrozen layers (ResNet-50)	61
Εικόνα 23 : Training VS Validation Loss with unfrozen layers (ResNet-50)	62
Εικόνα 24 : Μετρήσεις επιδόσεων στο σύνολο δεδομένων επικύρωσης (ResNet-50)	63
Εικόνα 25 : Πίνακας Σφαλμάτων (ResNet-50)	64

ΠΕΡΙΛΗΨΗ

Το κίνητρο πίσω από αυτή την εργασία πηγάζει από τις αξιοσημείωτες προόδους που σημειώθηκαν στον τομέα της επεξεργασίας και αναγνώρισης εικόνων με τη χρήση αλγορίθμων βαθιάς μάθησης τα τελευταία χρόνια, καθώς και από τις βαθείς επιπτώσεις της πανδημίας του Covid-19. Η ταχεία εξάπλωση του ιού δημιούργησε την επείγουσα ανάγκη ενίσχυσης του τομέα της υγειονομικής περίθαλψης με τεχνολογίες αιχμής ώστε να βελτιωθεί η ιατρική περίθαλψη των ασθενών.

Η εκθετική ανάπτυξη της έρευνας στον τομέα της επεξεργασίας και αναγνώρισης εικόνων έχει ανοίξει νέους δρόμους για την αξιοποίηση εξελιγμένων τεχνικών στην ιατρική διάγνωση.

Ειδικότερα, η ταξινόμηση εικόνας έχει αναδειχθεί σε πολύτιμο εργαλείο στην ακτινολογία για να βοηθήσει τους γιατρούς να εντοπίσουν ανωμαλίες σε ακτινογραφίες. Αξιοποιώντας τη δύναμη της βαθιάς μάθησης και των συνελκτικών νευρωνικών δικτύων, επιδιώκουμε να αξιοποιήσουμε αυτές τις εξελίξεις για την ανάπτυξη αποτελεσματικών μοντέλων για την ανίχνευση του Covid-19.

Η παρούσα διπλωματική έχει ως σκοπό τη διερεύνηση της αποτελεσματικότητας των τεχνικών βαθιάς μηχανικής μάθησης στην ανάλυση εικόνων ακτινογραφίας θώρακα και πιο συγκεκριμένα για να προσδιοριστεί εάν ένας ασθενής πάσχει ή έπασχε από Covid-19 ή όχι.

Για την εκπόνηση αυτής της εργασίας έγινε χρήση ενός μεγάλου συνόλου δεδομένων από ακτινογραφίες του θώρακα που συλλέχθηκαν από ασθενείς που διαγνώστηκαν είτε θετικά είτε αρνητικά στο Covid-19. Χρησιμοποιήθηκαν μοντέλα βαθιάς μάθησης, για την εξαγωγή ουσιαστικών χαρακτηριστικών και συμπερασμάτων από τις εικόνες και την ταξινόμησή τους σε κατηγορίες θετικών ή αρνητικών.

Τα αναμενόμενα αποτελέσματα της παρούσας εργασίας σκοπό έχουν να εξετάσουν ποσό κοντά μπορούμε να φτάσουμε στη σωστή διάγνωση με την βοήθεια της τεχνητής νοημοσύνης .Ωστόσο , σε καμία περίπτωση δεν μπορεί να παρακάμψει έναν επιστήμονα υγείας.

Λέξεις κλειδιά: Συνελκτικά Νευρωνικά Δίκτυα (ΣΝΔ) , Βαθιά Μάθηση, Μηχανική Μάθηση, Covid-19, Ακτινογραφία

Abstract

The significant advances in image processing and recognition using deep learning algorithms in recent years, as well as the profound impact of the Covid-19 pandemic, provide the motivation for this thesis. The virus's rapid spread has highlighted the urgent need to equip the healthcare sector with innovative technologies to improve patient care.

The growth of research in image processing and recognition has created new opportunities for advanced techniques in medical diagnosis.

Image classification is a valuable tool in radiology for identifying abnormalities in X-rays. Our aim is to develop effective models for Covid-19 detection by harnessing the power of deep learning and convolutional neural networks.

The objective of this study is to assess the efficacy of deep machine learning methods in analyzing chest X-rays, specifically in identifying the presence or absence of Covid-19 in patients.

This study used a large dataset of chest radiographs collected from patients diagnosed as either positive or negative for Covid-19. Deep learning models were employed to extract meaningful features and inferences from the images and classify them into positive or negative categories.

The expected results of this paper are intended to examine how close we can get to the correct diagnosis with the help of artificial intelligence. However, in no way can it bypass a health scientist.

Keywords: Convolutional Neural Networks (CNNs), Deep Learning, Machine Learning (ML), Covid-19, X-ray

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται αναφορά στην πανδημία του Covid-19 ,τονίζοντας τον κρίσιμο ρόλο που παίζουν οι ακτινογραφίες θώρακα στην διάγνωση του. Επιπλέον θέτονται οι στόχοι της έρευνας. Και τέλος παρουσιάζεται η δομή της παρούσας διπλωματικής εργασίας.

1.1 Ανάλυση του προβλήματος

Ο Covid-19 είναι μια ασθένεια που προκαλείται από τον ιό SARS-CoV-2, εντοπίστηκε για πρώτη φορά στο Wuhan της Κίνας στα τέλη του 2019 και από τότε έχει μετατραπεί σε παγκόσμια πανδημία. Η νόσος εκδηλώνεται με ποικίλα συμπτώματα, με πιο συνηθισμένα τον πυρετό, τον ξηρό βήχα και την κόπωση, ενώ σε σοβαρές περιπτώσεις παρατηρούνται πιο οξεία συμπτώματα, όπως δυσκολία στην αναπνοή και υψηλός πυρετός. Ο ιός μεταδίδεται κυρίως μέσω αναπνευστικών σταγονιδίων και αερολυμάτων, ιδίως σε εσωτερικούς χώρους. Η περίοδος επώασης κυμαίνεται έως και 14 ημέρες, με τα συμπτώματα να εμφανίζονται συνήθως εντός των 11,5 ημερών μετά τη μόλυνση. [1]

Οι διαγνωστικές μέθοδοι περιλαμβάνουν την PCR και τις ταχείες αντιγονικές εξετάσεις, ενώ οι εξετάσεις αντισωμάτων χρησιμοποιούνται για την ανίχνευση παλαιότερων λοιμώξεων. Τα προληπτικά μέτρα είναι κρίσιμα, με τον εμβολιασμό να αποτελεί την κύρια μέθοδο, η οποία συμπληρώνεται από την κοινωνική αποστασιοποίηση, τον κατάλληλο αερισμό, την τακτική υγιεινή των χεριών και την προαιρετική χρήση μάσκας σε δημόσιους χώρους. Η θεραπεία είναι βασισμένη στα συμπτώματα του εκάστοτε ασθενή και συμπεριλαμβάνει την υποστηρικτική φροντίδα και την απομόνωση. Με συνεχή έρευνα για θεραπευτικά φάρμακα, ένα από τα οποία εγκρίθηκε στο Ηνωμένο Βασίλειο το 2021. [1]

Η ακτινογραφία θώρακα αποτελεί θεμελιώδες μέρος της ακτινολογικής απεικόνισης εδώ και πολλά χρόνια. Είναι η πιο συχνά διενεργούμενη ακτινολογική εξέταση παγκοσμίως. Για παράδειγμα, στις ανεπτυγμένες χώρες, πραγματοποιούνται περίπου 238 ακτινογραφίες θώρακος ανά 1000 άτομα κάθε χρόνο, όπως ανέφεραν τα Ηνωμένα Έθνη το 2008. Συγκεκριμένα, στις Ηνωμένες Πολιτείες, μόνο το 2006 λήφθηκαν

περίπου 129 εκατομμύρια ακτινογραφίες, σύμφωνα με τον Mettler και τους συνεργάτες του. Η ευρεία χρήση και ζήτηση για ακτινογραφίες μπορεί να αποδοθεί σε μεγάλο βαθμό στην προσιτή τιμή τους και στη χαμηλή έκθεση σε ακτινοβολία, σε συνδυασμό με τη σχετικά υψηλή ευαισθησία τους στην ανίχνευση διάφορων ιατρικών καταστάσεων. Συνήθως, η ακτινογραφία είναι η αρχική απεικονιστική εξέταση που διενεργείται και διαδραματίζει καθοριστικό ρόλο στον έλεγχο, τη διάγνωση και τη θεραπεία ενός ευρέος φάσματος θεμάτων υγείας. [2]

Στο πλαίσιο της πανδημίας Covid-19, οι ακτινογραφίες θώρακος είχαν αποκτήσει ακόμη πιο κρίσιμο ρόλο. Χρησιμοποιήθηκαν συχνά για την αξιολόγηση και τη διαχείριση ασθενών για τους οποίους υπήρχε η υποψία ή ακόμη και για την κλινική εικόνα του ασθενή μετά τη νόσηση του από τον ιό για να καθοριστεί αν ο ιός του είχε προκαλέσει κάποια βλάβη. Και στις δυο περιπτώσεις ήταν ιδιαίτερα χρήσιμες, καθώς η απεικόνιση μπορεί να αποκαλύψει πνευμονικές αλλαγές και ανωμαλίες που σχετίζονται με τον ιό, βοηθώντας στην αξιολόγηση της σοβαρότητας και της εξέλιξης της νόσου και ο ασθενής να λάβει την απαραίτητη φροντίδα και περίθαλψη αλλά και στη δεύτερη περίπτωση ο ασθενής που νόσησε επιβεβαιωμένα από Covid-19 να γνωρίζουν οι θεράποντες γιατροί την κλινική εικόνα του μετά το πέρας της ασθένειας.

Οι ακτινογραφίες είναι ιδιαίτερα πολύτιμες λόγω της ταχείας εκτέλεσής τους και της άμεσης διαθεσιμότητας των αποτελεσμάτων, η οποία είναι απαραίτητη για τη διαχείριση μιας εξαιρετικά μολυσματικής νόσου όπως ο Covid-19. Αυτό το γρήγορο και αποτελεσματικό διαγνωστικό εργαλείο αποδείχθηκε καθοριστικό για την έγκαιρη λήψη αποφάσεων για την θεραπεία της νόσου κατά τη διάρκεια της πανδημίας.

Σε συνάρτηση με όσα αναφέρθηκαν και το αυξανόμενο ενδιαφέρον για τη χρήση προηγμένων τεχνολογιών όπως είναι τα νευρωνικά δίκτυα για την αντιμετώπιση και την ανάλυση μιας νόσου όπως είναι ο Covid-19 στην παρούσα διπλωματική θα προσεγγίσουμε έναν διαφορετικό τρόπο για την ανίχνευση θετικών και αρνητικών κρουσμάτων από τις μεθόδους που προαναφέραμε. Πιο συγκεκριμένα θα χρησιμοποιήσουμε δυο βαθιά συνελκτικά νευρωνικά δίκτυα ένα από την αρχή και ένα προ εκπαιδευμένο μοντέλο πάνω στο σύνολο δεδομένων ImageNet το ResNet-50. Τα δεδομένα που θα χρησιμοποιήσουμε για να εκπαιδεύσουμε να επικυρώσουμε και για να ελέγχουμε τα μοντέλα μας έχουν αντληθεί από το Kaggle.

Ο πρωταρχικός στόχος αυτής της έρευνας είναι να εκπαιδεύσει και να συγκρίνει αυτές τις δυο αρχιτεκτονικές πάνω στο ίδιο σύνολο δεδομένων και να εξετάσει ποια αρχιτεκτονική μπορεί να ταξινομήσει πιο ικανοποιητικά τους ασθενείς σε θετικούς και αρνητικούς . Αυτή η συγκριτική ανάλυση θα παράσχει πολύτιμες πληροφορίες σχετικά με την απόδοση και την αποτελεσματικότητα των διάφορων αρχιτεκτονικών συνελκτικών νευρωνικών δικτύων για την ανίχνευση του Covid-19 , βοηθώντας στην ανάπτυξη κατά προσέγγιση ακριβέστερων διαγνωστικών εργαλείων.

1.2 Δομή της Διπλωματικής εργασίας

Η παρούσα διπλωματική αποτελείται από τέσσερα κεφάλαια.

Στο πρώτο κεφάλαιο γίνεται μια σύντομη περιγραφή του προβλήματος και τους στόχους αυτής της εργασίας.

Στο δεύτερο κεφάλαιο γίνεται μια θεωρητική εισαγωγή στις έννοιες και στις τεχνικές που εφαρμόστηκαν ώστε να προσεγγίσουμε το πρόβλημα.

Στο τρίτο κεφάλαιο γίνεται μια σύντομη παρουσίαση των εργαλείων που χρησιμοποιήθηκαν και στη συνέχεια αναλύονται τα μοντέλα που προέκυψαν και οι επιδόσεις τους πάνω στο σύνολο δεδομένων που επιλέχθηκε.

Τέλος, στο τέταρτο κεφάλαιο καταγράφονται τα συμπεράσματα στα οποία καταλήξαμε μετά την ολοκλήρωση των πειραματικών διαδικασιών και τρόπους για τη βελτίωση των μοντέλων στο μέλλον.

ΚΕΦΑΛΑΙΟ 2: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτό το κεφάλαιο προσφέρεται μια συνολική επισκόπηση των βασικών εννοιών και μεθοδολογιών που υποστηρίζουν την τεχνητή νοημοσύνη και θέτονται οι βάσεις για την κατανόηση των μοντέλων και των τεχνικών που εφαρμοστήκαν.

2.1 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη (Artificial Intelligence) είναι ένας κλάδος της επιστήμης των υπολογιστών που στοχεύει στη δημιουργία μηχανών που μπορούν να μιμηθούν διάφορες λειτουργίες που μπορεί να κάνει ο άνθρωπος. Αντί να εκτελούν εργασίες κατά εντολή, τα συστήματα τεχνητής νοημοσύνης μπορούν να μαθαίνουν, να επιλύουν προβλήματα, να κατανοούν τη γλώσσα, να αναγνωρίζουν μοτίβα και να προσαρμόζονται σε νέες πληροφορίες ή ερεθίσματα. [3]

Η ραγδαία ανάπτυξη των συστημάτων τεχνητής νοημοσύνης οφείλεται σε μεγάλο βαθμό στη μηχανική μάθηση και ιδιαίτερα στην ανάπτυξη νευρωνικών δικτύων και της βαθιάς μηχανικής μάθησης. Για αυτό το λόγο όταν μιλάμε για τεχνητή νοημοσύνη στην ουσία μιλάμε για δύο θεμελιώδεις έννοιες τη μηχανική μάθηση και τη βαθιά μάθηση. [3]

Η μηχανική μάθηση περιλαμβάνει αλγορίθμους που επιτρέπουν στους υπολογιστές να μαθαίνουν από ένα μεγάλο όγκο δεδομένων. Η εκπαίδευση αυτού του συστήματος κατά αυτόν τον τρόπο του επιτρέπει να κάνει προβλέψεις ή να παίρνει αποφάσεις χωρίς να έχει προγραμματιστεί ρητά. [3]

Η βαθιά μάθηση από την άλλη είναι ένα υποσύνολο της μηχανικής μάθησης, χρησιμοποιεί τεχνητά νευρωνικά δίκτυα με πολλά επίπεδα. Αυτά τα επίπεδα επιτρέπουν στο σύστημα να μαθαίνει από μεγάλες ποσότητες δεδομένων και να εκτελεί σύνθετες εργασίες, όπως η αναγνώριση εικόνας και ομιλίας, η επεξεργασία φυσικής γλώσσας και η επεξεργασία κείμενων. [3]

Τα συστήματα τεχνητής νοημοσύνης έχουν υιοθετηθεί ευρέως σε διάφορους τομείς, από την υγειονομική περίθαλψη, την εκπαίδευση, τη χρηματοδότηση και το εμπόριο έως τα αυτόνομα οχήματα. Ωστόσο, ενώ η τεχνητή νοημοσύνη έχει εξαιρετικές

δυνατότητες, παρουσιάζει επίσης σημαντικές ηθικές, νομικές και κοινωνικές προκλήσεις που αν δεν αντιμετωπιστούν προσεκτικά και άμεσα θα έχουν τεράστιο αντίκτυπο στη ζωή των ανθρώπων. [3]

2.2 Μηχανική Μάθηση

Η μηχανική μάθηση (Machine Learning) είναι ένα υποσύνολο της τεχνητής νοημοσύνης όπως αναφέρθηκε και παραπάνω που επικεντρώνεται στην ανάπτυξη αλγορίθμων και στατιστικών μοντέλων, επιτρέποντάς στα συστήματα υπολογιστών να μαθαίνουν και να λαμβάνουν αποφάσεις ή ακόμη και να κάνουν προβλέψεις με βάση τα δεδομένα που έχουν εκπαιδευτεί. Ο πρωταρχικός στόχος της μηχανικής μάθησης είναι να επιτρέψει στις μηχανές να μαθαίνουν αυτόματα χωρίς ανθρώπινη παρέμβαση ή βοήθεια. [4]

Η ζήτηση για μηχανική μάθηση συνεχίζει να αυξάνεται λόγω της αυξανόμενης διαθεσιμότητας τεράστιων συνόλων δεδομένων και καθώς οι βιομηχανίες επιδιώκουν να εξάγουν σχετικές γνώσεις και να μειώσουν το κόστος και να αυτοματοποιήσουν κάποιες διεργασίες. Οι ερευνητές έχουν ακολουθήσει διάφορες προσεγγίσεις για την επίτευξη αυτόματης μάθησης χωρίς ρητό προγραμματισμό, αξιοποιώντας συχνά εκτεταμένα σύνολα δεδομένων. Η μηχανική μάθηση περιλαμβάνει μια σειρά αλγορίθμων για την αντιμετώπιση των προκλήσεων που σχετίζονται με δεδομένα, με την επιλογή αλγορίθμων να βασίζεται σε συγκεκριμένες απαιτήσεις του προβλήματος, τις εκτιμήσεις μεταβλητών και τις κατάλληλες τεχνικές μοντελοποίησης. [4]

Η μηχανική μάθηση αποτελείται από τέσσερις πολύ βασικές κατηγορίες την Επιβλεπόμενη, τη Μη Επιβλεπόμενη, την Ημι-Επιβλεπόμενη και την Ενισχυτική Μάθηση.

Επιβλεπόμενη Μάθηση (Supervised Learning): Η επιβλεπόμενη μάθηση περιλαμβάνει την εκπαίδευση αλγορίθμων σε ένα σύνολο δεδομένων με ετικέτες, όπου παρέχονται τόσο στην είσοδο όσο και στην έξοδο, επιτρέποντάς του να μάθει τα υποκείμενα μοτίβα. Οι προβλέψεις του αλγορίθμου προσαρμόζονται με βάση τα σφάλματα που έγιναν κατά την εκπαίδευση, με στόχο την ελαχιστοποίηση τους. Ο στόχος είναι να

εκπαιδεύει ο αλγόριθμος ώστε να αναγνωρίζει και να προβλέπει τις ετικέτες νέων αγνώστων σε αυτόν δεδομένων. [4]

Μη Επιβλεπόμενη (Unsupervised Learning): Σε αντίθεση με την επιβλεπόμενη μάθηση, η μη επιβλεπόμενη μάθηση εκπαιδεύεται με δεδομένα χωρίς ετικέτες. Οι αλγόριθμοι βασίζονται στις δικές τους ικανότητες για να εντοπίσουν και να παρουσιάσουν τις ενδιαφέρουσες δομές ή τα μοτίβα μέσα στα δεδομένα που τους δίνονται. Ο στόχος είναι να καταφέρει να ομαδοποιήσει τα παρόμοια δείγματα δεδομένων με βάση τα κοινά χαρακτηριστικά τους και στη συνέχεια να τα εφαρμόσει σε νέα άγνωστα σε αυτόν δεδομένα. [4]

Ημι – Επιβλεπόμενη Μάθηση (Semi - Supervised Learning): Η ημι – επιβλεπόμενη μάθηση είναι ένας συνδυασμός των δύο προηγούμενων. Χρησιμοποιεί μια ποσότητα δεδομένων με ετικέτα σε συνδυασμό με μια άλλη ποσότητα δεδομένων χωρίς ετικέτα. Είναι ιδιαίτερα χρήσιμη και γεφυρώνει το χάσμα μεταξύ της επιβλεπόμενης και μη επιβλεπόμενης μάθησης προσφέροντας μια εναλλακτική λύση σε τομείς όπου τα δεδομένα με ετικέτα είναι δύσκολο να συλλεχθούν ή είναι περιορισμένα, ενώ από την άλλη μπορεί να είναι διαθέσιμα αρκετά δεδομένα χωρίς ετικέτα. Ο αλγόριθμος μαθαίνει από τα δεδομένα με ετικέτα και στη συνέχεια ενσωματώνει τα δεδομένα χωρίς ετικέτα ώστε να μπορέσει να ανακαλύψει τα υποκείμενα μοτίβα και τη δομή τους. [4]

Ενισχυτική Μάθηση (Reinforcement Learning): Η ενισχυτική μάθηση επικεντρώνεται στον τρόπο με τον οποίο οι πράκτορες λογισμικού μαθαίνουν πώς να συμπεριφέρονται σε ένα περιβάλλον και να λαμβάνουν βέλτιστες αποφάσεις με σκοπό τη μεγαλύτερη ανταμοιβή τους. Στόχος είναι η ανάπτυξη στρατηγικών που επιτρέπουν στους πράκτορες να λαμβάνουν τεκμηριωμένες αποφάσεις για τη μεγιστοποίηση της συνολικής ανταμοιβής που λαμβάνουν. Η ενισχυτική μάθηση διαφέρει από την επιβλεπόμενη και τη μη επιβλεπόμενη μάθηση στο ότι δίνει έμφαση στη μάθηση από τις αλληλεπιδράσεις και την ανατροφοδότηση αντί να βασίζεται σε δεδομένα με ετικέτα ή στον εντοπισμό μοτίβων. [4]

2.3 Βαθιά Μάθηση

Η βαθιά μάθηση (Deep Learning) όσο και η μηχανική μάθηση εμπίπτουν στην ομπρέλα της τεχνητής νοημοσύνης, επικεντρώνεται στην ανάπτυξη και εφαρμογή τεχνητών νευρωνικών δικτύων, ιδίως των βαθιών νευρωνικών δικτύων. Περιλαμβάνει την εκπαίδευση αυτών των δικτύων ώστε να μαθαίνουν και να εξάγουν αυτόματα ιεραρχικές αναπαραστάσεις δεδομένων μέσω πολλαπλών επιπέδων διασυνδεδεμένων κόμβων ή νευρώνων.

Σε αντίθεση με τους παραδοσιακούς αλγορίθμους μηχανικής μάθησης που απαιτούν χειροκίνητη σχεδίαση χαρακτηριστικών, τα μοντέλα βαθιάς μάθησης έχουν τη δυνατότητα να μαθαίνουν και να εξάγουν αυτόματα σημαντικά χαρακτηριστικά απευθείας από τα ακατέργαστα δεδομένα. Αυτό επιτυγχάνεται με την αξιοποίηση της ισχύος των νευρωνικών δικτύων με πολλαπλά κρυφά επίπεδα (layers), τα οποία επιτρέπουν στα μοντέλα να καταγράφουν περίπλοκα μοτίβα και σχέσεις στα δεδομένα. [5]

Οι αλγόριθμοι βαθιάς μάθησης υπερέχουν στην επεξεργασία σύνθετων και μεγάλης κλίμακας συνόλων δεδομένων, όπως οι εικόνες, τα κείμενα, η ομιλία και τα δεδομένα χρονοσειρών. Έχουν επιδείξει εξαιρετικές επιδόσεις σε διάφορους τομείς, όπως η όραση υπολογιστών, η επεξεργασία φυσικής γλώσσας, η αναγνώριση ομιλίας και τα συστήματα συστάσεων. [5]

Η διαδικασία εκπαίδευσης στη βαθιά μάθηση περιλαμβάνει την τροφοδοσία του νευρωνικού δικτύου με δεδομένα με ετικέτα και την επαναληπτική προσαρμογή των βαρών σύνδεσης μεταξύ των νευρώνων ώστε να ελαχιστοποιηθεί η διαφορά μεταξύ των προβλεπόμενων και των πραγματικών εξόδων. Αυτή η διαδικασία, γνωστή ως backpropagation, επιτρέπει στο δίκτυο να βελτιστοποιήσει την απόδοσή του και να βελτιώσει την ικανότητά του να κάνει ακριβείς προβλέψεις ή ταξινομήσεις. [5]

2.4 Ταξινόμηση εικόνας

Η ταξινόμηση εικόνας είναι η διαδικασία λήψης μιας εικόνας ή μιας φωτογραφίας και η προσπάθεια ενός υπολογιστή να την ταξινομήσει αυτόματα ή να προσπαθήσει να βρει την πιθανή κατηγορία στην οποία ανήκει. Στην καθημερινότητά μας έχει διάφορες εφαρμογές όπως στα smartphones όπου μπορούμε να δούμε τις φωτογραφίες μας ταξινομημένες κατά φίλους, οικογένεια, ταξίδια, κτλ. Όπως επίσης έχει γίνει ιδιαίτερα χρήσιμη σε μια ποικιλία εφαρμογών, όπως η αναγνώριση προσώπου, η ανίχνευση αντικειμένων σε αυτόνομα οχήματα, η ανάλυση ιατρικών εικόνων και πολλά άλλα.

Στην πράξη κατά τη διαδικασία ταξινόμησης εικόνων ένας αλγόριθμος εκπαιδεύεται για να αναγνωρίζει διάφορες πτυχές ή κατηγορίες μέσα σε μια εικόνα και να τις ταξινομήσει. Δεδομένου ότι ο υπολογιστής κατανοεί τις τιμές της έντασης μιας ψηφιακής εικόνας και όχι απευθείας την εικόνα, χρησιμοποιεί αυτές τιμές ώστε να μπορέσει να κάνει την ταξινόμηση.

Ο τύπος νευρωνικών δικτύων που εφαρμόζεται συνήθως για την αναγνώριση μοτίβων σε μια εικόνα είναι τα συνελκτικά νευρωνικά δίκτυα, ένας τύπος αλγορίθμου βαθιάς μάθησης ειδικά σχεδιασμένου για την επεξεργασία δεδομένων που μοιάζουν με πλέγμα, όπως οι εικόνες.

Η διαδικασία εκπαίδευσης περιλαμβάνει την τροφοδοσία του αλγορίθμου με μεγάλο αριθμό ψηφιακών εικόνων με ετικέτα. Ο αλγόριθμος γνωρίζοντας από πριν την κατηγορία ή την κλάση που ανήκει η κάθε εικόνα που εκπαιδεύεται μπορεί να μάθει τα ξεχωριστά χαρακτηριστικά που σχετίζονται με την κάθε κατηγορία.

Αφού εκπαιδευτεί ο αλγόριθμος και με βάση αυτά που έχει μάθει μπορεί να κάνει πρόβλεψη για μια νέα άγνωστη σε αυτόν ψηφιακή εικόνα και να μπορέσει να την τοποθετήσει στην κατηγορία στην οποία ανήκει.

2.5 Νευρωνικά Δίκτυα

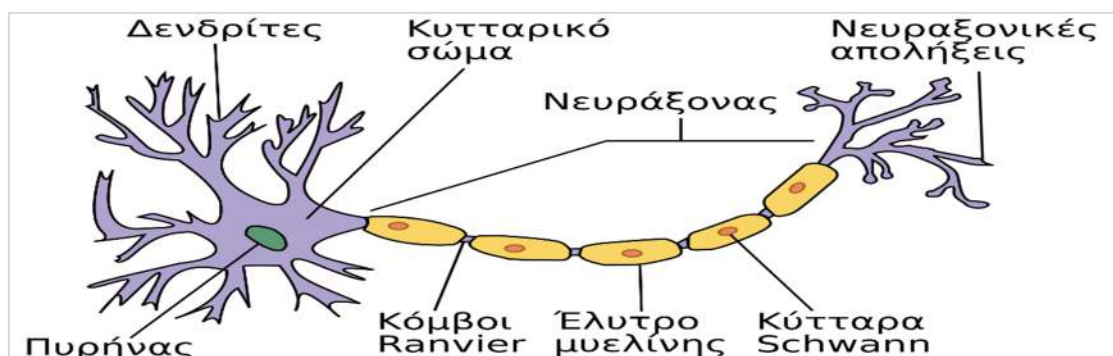
Τα νευρωνικά δίκτυα αποτελούν τον πυρήνα της σύγχρονης τεχνητής νοημοσύνης και της μηχανικής μάθησης. Το εντυπωσιακό με τα νευρωνικά δίκτυα είναι ότι πρόκειται

για αλγορίθμους που είναι εμπνευσμένοι από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου. Τα νευρωνικά δίκτυα είναι ιδιαίτερα ισχυρά στην αναγνώριση προτύπων, την ταξινόμηση δεδομένων και την πρόβλεψη. Χρησιμοποιούνται σε ποικίλες εφαρμογές, όπως την αναγνώριση εικόνας και ομιλίας, την ανάλυση συναισθημάτων, τα αυτόνομα οχήματα και πολλές άλλες.

2.5.1 Βιολογικά Νευρωνικά Δίκτυα

Για να μπορέσουμε να καταλάβουμε καλύτερα πως λειτουργούν τα τεχνητά νευρωνικά δίκτυα θα πρέπει να έχουμε μια εικόνα για το πως λειτουργεί ο ανθρώπινος εγκέφαλος. Αφού η ιδέα για το πως θα μπορούσε να λειτουργήσει ένας τεχνητός νευρώνας από τον καθηγητή νευρολογίας Frank Rosenblatt [6] εμπνεύστηκε σε αντιστοιχία με τα βιολογικό νευρώνα. Οι νευρώνες αποτελούν το βασικό δομικό κομμάτι του ανθρώπινου εγκεφάλου. Ο ανθρώπινος εγκέφαλος περιέχει δισεκατομμύρια νευρώνες χωρισμένο σε ομάδες και κάθε μια από αυτές αποτελεί ένα φυσικό νευρωνικό δίκτυο. [7]

Οι νευρώνες λαμβάνουν ηλεκτροχημικές εισροές μέσω των δενδριτών τους από άλλους νευρώνες. Εάν το άθροισμα αυτών των εισόδων υπερβεί ένα ορισμένο όριο, ο νευρώνας ενεργοποιείται και μεταδίδει ένα ηλεκτροχημικό σήμα κατά μήκος του κυττάρου -νευροάξονα του, μεταφέροντας το σήμα σε άλλους συνδεδεμένους νευρώνες στις απολήξεις του άξονα. Όπως φαίνεται και από την παρακάτω εικόνα 1. [7] [8]



Εικόνα 1 : Η δομή ενός βιολογικού νευρώνα

Πηγή :[<https://el.wikipedia.org/wiki/Νευράξονας>]

Αυτοί οι διασυνδεδεμένοι νευρώνες μπορούν στη συνέχεια επίσης να ενεργοποιηθούν. Είναι σημαντικό ότι οι νευρώνες ενεργοποιούνται μόνο όταν το συνολικό λαμβανόμενο σήμα υπερβαίνει ένα συγκεκριμένο επίπεδο. Δεν υπάρχουν διαφορετικοί βαθμοί πυροδότησης πρόκειται για μια δυαδική διαδικασία.

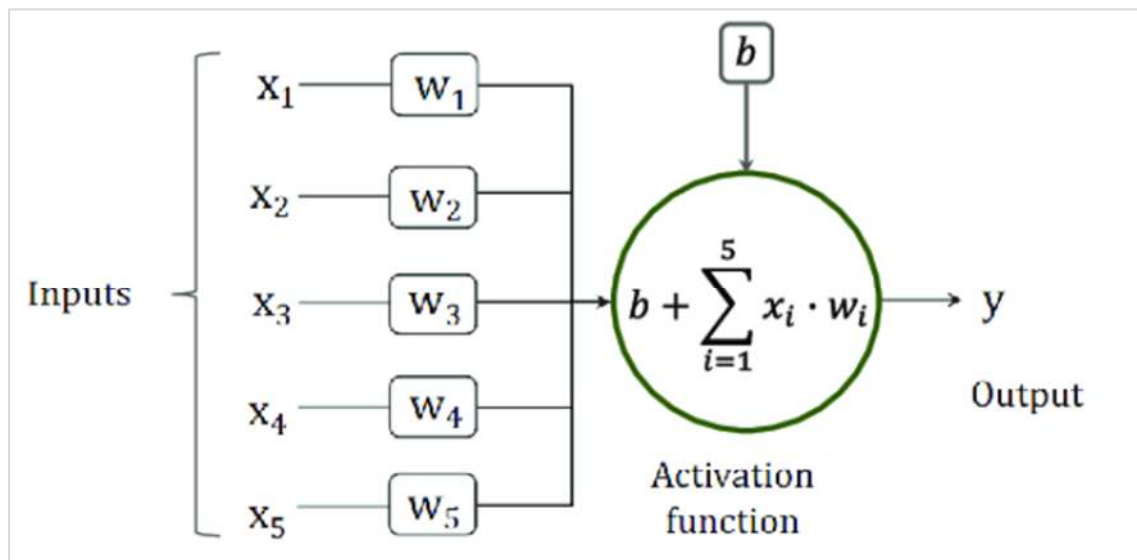
Ολόκληρος ο εγκέφαλος αποτελείται από διασυνδεδεμένους νευρώνες που μεταδίδουν ηλεκτροχημικά σήματα, λειτουργώντας συλλογικά ως απλές μονάδες επεξεργασίας. Κάθε μονάδα υπολογίζει το σταθμισμένο άθροισμα των εισόδων της και πυροδοτεί ένα δυαδικό σήμα εάν η συνολική είσοδος ξεπεράσει ένα προκαθορισμένο όριο. [7] [8]

2.5.2 Τεχνητά Νευρωνικά Δίκτυα

Όπως προ είπαμε η έμπνευση για τα τεχνητά νευρωνικά δίκτυα που έφερε την επανάσταση στην τεχνητή νοημοσύνη προήλθε από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου. Ένας τεχνητός νευρώνας αποτελείται από έναν αριθμό εισόδων $x_1, x_2, x_3 \dots x_n$ και μια μόνο έξοδο y .

$$f(x) = b + \sum_{i=1}^n x_i w_i$$

Σε αυτή την εξίσωση το x_i αντιπροσωπεύει τις εισόδους και το w_i είναι τα αντίστοιχα βάρη και b είναι η πόλωση. Τα βάρη κατά τη διάρκεια της εκπαίδευσης μεταβάλλονται συνεχώς και έτσι καθορίζεται σε ποιον βαθμό θα επηρεάσει την έξοδο του νευρώνα η κάθε είσοδος. Υπολογίζεται το σταθμισμένο άθροισμα των εισόδων πολλαπλασιασμένο με τα βάρη συν η πόλωση. Η συνάρτηση ενεργοποίησης f , εφαρμόζεται στο σταθμισμένο άθροισμα συν την πόλωση για να εισάγει μη γραμμικότητα και να καθορίσει την έξοδο του νευρώνα. Όπως φαίνεται και στην παρακάτω εικόνα 2.



Εικόνα 2 : Βασικό Νευρωνικό Δίκτυο

Πηγή : [https://www.researchgate.net/figure/General-scheme-of-an-artificial-neural-network-ANN_fig12_352338692]

Η επιλογή της συνάρτησης ενεργοποίησης εξαρτάται από το συγκεκριμένο πρόβλημα και την επιθυμητή συμπεριφορά του δικτύου.

2.5.3 Συνελικτικά Νευρωνικά Δίκτυα

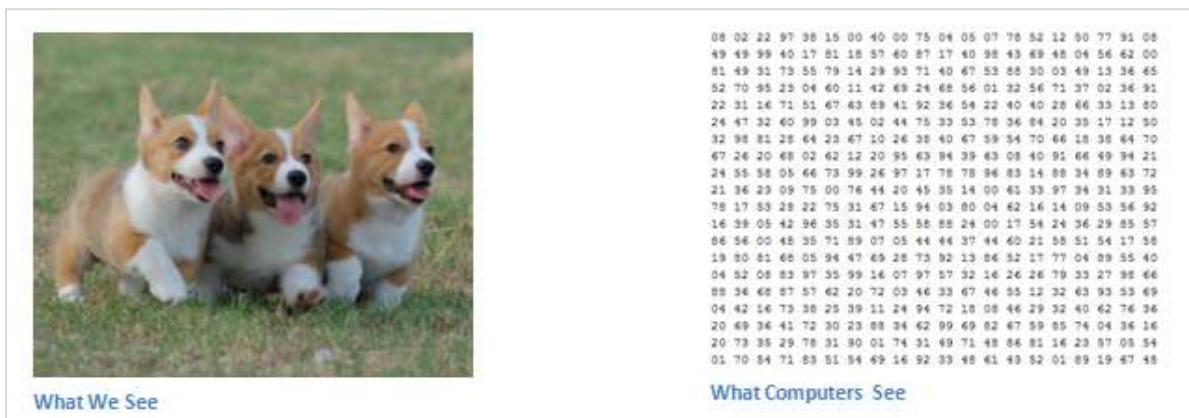
Τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks) ανήκουν στην κατηγορία των βαθιών νευρωνικών δικτύων και αποτελούν αλγορίθμους βαθιάς μάθησης. Είναι ένας ειδικός τύπος νευρωνικών δικτύων πολλών επιπέδων που παρότι έχουν σχεδιαστεί ειδικά για την ανάλυση των οπτικών εισόδων, όπως είναι οι εικόνες πλέον η χρήση τους έχει επεκταθεί και σε άλλους τομείς όπως είναι η επεξεργασία φυσικής γλώσσας, η αναγνώριση αντικειμένων και η αναγνώριση ομιλίας κ.α.

Η ιδέα πίσω από αυτήν την αρχιτεκτονική είναι να προσεγγίσουν οι μηχανές τον τρόπο με τον οποίο ο άνθρωπος αντιλαμβάνεται το κόσμο καθώς η αρχιτεκτονική είναι παρόμοια με αυτή της συνδεσιμότητας των νευρώνων στον ανθρώπινο εγκέφαλο και πιο συγκεκριμένα είναι εμπνευσμένη από την οργάνωση του οπτικού φλοιού. [9]

Αυτά τα δίκτυα μαθαίνουν πρώτα να αναγνωρίζουν απλά σχήματα όπως και εμείς μαθαίνουμε να αναγνωρίζουμε βασικά αντικείμενα. Στη συνέχεια όπως και ο άνθρωπος που μπορεί να επικεντρώνεται στις λεπτομέρειες έτσι και τα δίκτυα αυτά με

τη χρήση των πολλαπλών επιπέδων καθένα από το οποίο βασίζεται στο προηγούμενο προσπαθούν να αντιληφθούν χαρακτηριστικά και λεπτομέρειες υψηλότερου επιπέδου σε μια εικόνα.

Ωστόσο, σε αντίθεση με τον τρόπο που αντιλαμβανόμαστε εμείς μια εικόνα, ο υπολογιστής όταν λαμβάνει μια εικόνα ως είσοδο, το μόνο που βλέπει είναι ένας πίνακας από εικονοστοιχεία (pixels) όπως φαίνεται και από την εικόνα 3 που ακολουθεί.



Εικόνα 3 : Τι βλέπει ο άνθρωπος vs τι βλέπει ο υπολογιστής

Πηγή :[<https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>]

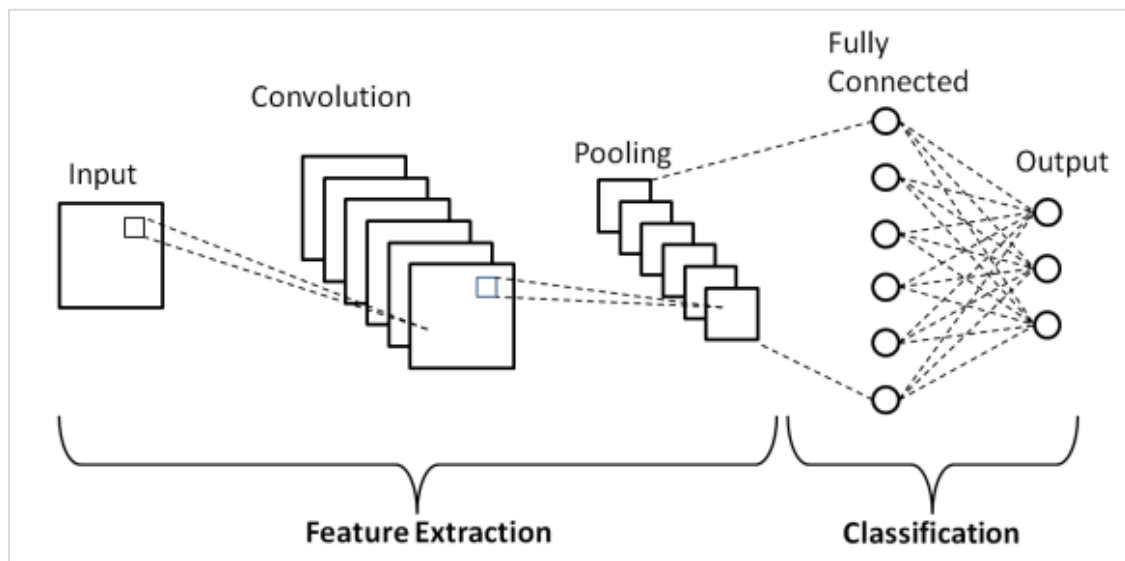
Ανάλογα με την ανάλυση και το μέγεθος της εικόνας, αυτός ο πίνακας μπορεί να είναι 32x32x3 όπου το 3 να συμβολίζει το RGB (κόκκινο, πράσινο και μπλε).Κάθε αριθμός σε αυτόν τον πίνακα αντιστοιχεί και σε μια τιμή από το 0 έως το 255.

Για εμάς ο παραπάνω πίνακας στην εικόνα 3 μπορεί να είναι απλά αριθμοί αλλά για τον υπολογιστή είναι ο μόνος τρόπος να κατανοήσει την εικόνα.

2.6 Ανάλυση Αρχιτεκτονικής των ΣΝΔ

Όπως αναφέραμε και στην παραπάνω ενότητα τα συνελκτικά νευρωνικά δίκτυα είναι ειδικά σχεδιασμένα για την επεξεργασία και την ανάλυση δεδομένων εικόνας και η δομή τους είναι ειδικά σχεδιασμένη ώστε να μπορεί να εξάγει χρήσιμα χαρακτηριστικά

από αυτού του τύπου τα δεδομένα. Τα επίπεδα τους θα μπορούσαν να διακριθούν σε δυο βασικές κατηγορίες τα επίπεδα που είναι αρμόδια για τον εντοπισμό και την εξαγωγή διάφορων χαρακτηριστικών από τις εικόνες καθώς και στη μείωση των διαστάσεων της εικόνας γνωστά ως επίπεδα συνέλιξης (Convolutional Layers) και στα επίπεδα που είναι υπεύθυνα για την τελική διαδικασία ταξινόμησης των εικόνων σε διάφορες κλάσεις γνωστά ως επίπεδα ταξινόμησης (Classification Layer).



Εικόνα 4 : Διαγραμματική απεικόνιση ενός βασικού CNN

Πηγή :[https://www.researchgate.net/figure/Schematic-diagram-of-a-basic-convolutional-neural-network-CNN-architecture-26_fig1_336805909]

Όπως μπορούμε να διακρίνουμε και από τη διαγραμματική απεικόνιση στην εικόνα 4 μια τυπική αρχιτεκτονική ενός CNN θα μπορούσε να ήταν η εξής:

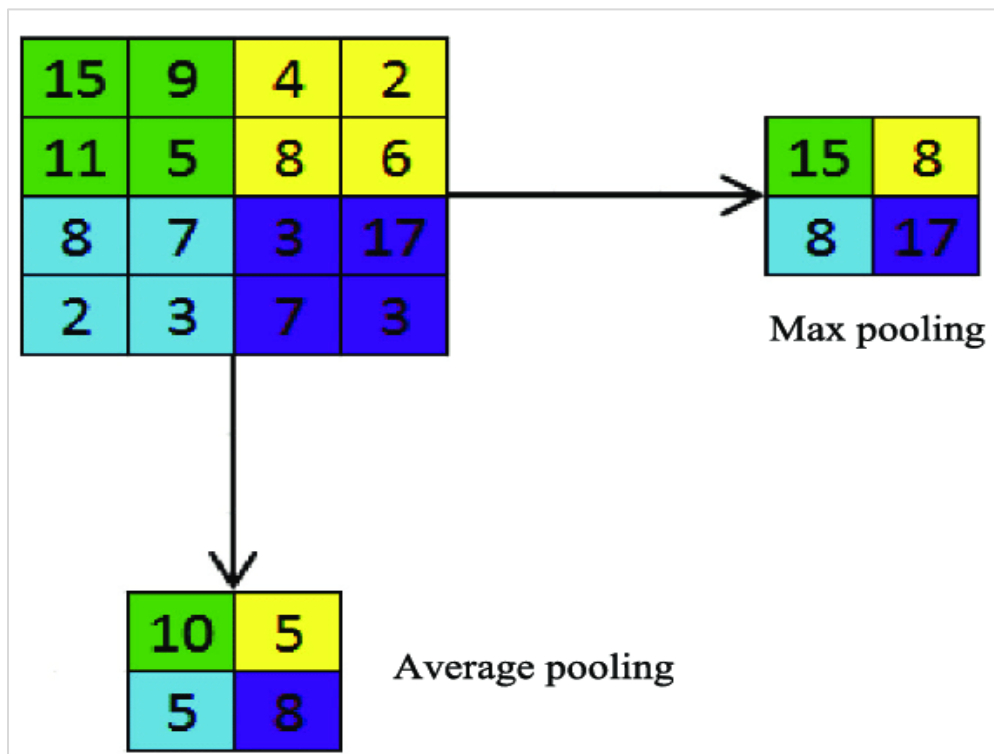
Επίπεδο εισόδου (Input Layer): Είναι το σημείο εκκίνησης του μοντέλου. Λαμβάνει σαν είσοδο μια εικόνα με τη μορφή πίνακα τιμών.

Συνελικτικά επίπεδα (Convolutional Layers): Τα συνελικτικά επίπεδα τα οποία είναι από τα πιο βασικά δομικά στοιχεία ενός ΣΝΔ είναι αυτά που σαρώνουν όλο το πλάτος και το ύψος της εικόνας και με την εφαρμογή φίλτρων εστιάζουν σε συγκεκριμένα χαρακτηριστικά όπως είναι οι ακμές, τα χρώματα και οι υφές προκειμένου να δημιουργήσουν το χάρτη χαρακτηριστικών (feature map). Ο χάρτης χαρακτηριστικών δεν είναι τίποτα άλλο παρά από ένα πίνακα που προκύπτει από το άθροισμα των βαρών του φίλτρου επί της αντίστοιχης φωτεινότητας της εικόνας. Το μέγεθος που θα

έχει ένας χάρτης χαρακτηριστικών δεν είναι συγκεκριμένο κάθε φορά και πρέπει να ορίζεται κατά τη σχεδίαση ενός ΣΝΔ. Και πιο συγκεκριμένα εξαρτάται από τις παραμέτρους padding, depth και strike.

- *Padding*: Το padding είναι ένα πρόσθετο πλαίσιο μηδενικών που μπορεί να προστεθεί γύρω από μια εικόνα ώστε να διατηρηθούν οι διαστάσεις της μετά τη συνέλιξη, επιτρέποντας στο δίκτυο να επεξεργάζεται πληροφορίες από τις άκρες της εικόνας οι οποίες διαφορετικά θα είχαν χαθεί.
- *Depth*: Το depth χαρακτηρίζει τον αριθμό των φίλτρων που θα εφαρμοστούν στη συνέλιξη. Κάθε φίλτρο είναι σχεδιασμένο να ανιχνεύει ένα συγκεκριμένο χαρακτηριστικό όπως γραμμές ή σχήματα. Δεν υπάρχει κάποιος κανόνας για τον αριθμό των φίλτρων καθώς κάθε φορά εξαρτάται από το είδος και την πολυπλοκότητα της εικόνας. Σε απλές εικόνες πολλά φίλτρα θα μπορούσαν να ανιχνεύουν παρόμοια χαρακτηριστικά ενώ λίγα φίλτρα σε εικόνες πολύπλοκες θα μπορούσαν να χάσουν σημαντικές λεπτομέρειες.
- *Stride*: Το stride αναφέρεται στο κατά πόσα εικονοστοιχεία (pixels) μετακινείται το φίλτρο σε κάθε βήμα της συνέλιξης. Ένα μικρότερο stride σημαίνει ότι το φίλτρο μετακινείται σε μικρότερα βήματα, πράγμα που μπορεί να αυξήσει τη λεπτομέρεια στον χάρτη χαρακτηριστικών αλλά ταυτόχρονα αυξάνει και τον όγκο των υπολογισμών.

Επίπεδο Ομαδοποίησης (Pooling Layers): Μετά από τα συνελικτικά επίπεδα ακολουθούν τα επίπεδα ομαδοποίησης τα οποία χρησιμοποιούνται για τη μείωση των διαστάσεων του χάρτη χαρακτηριστικών που προκύπτει από τα συνελικτικά επίπεδα. Παρά τη μείωση των διαστάσεων τα επίπεδα ομαδοποίησης διατηρούν τις πιο σημαντικές πληροφορίες και χαρακτηριστικά. Αυτή η λειτουργία βοηθάει το μοντέλο ώστε μη μαθαίνει την ακριβή θέση των χαρακτηριστικών στην εικόνα και να μπορεί να εντοπίσει ένα αντικείμενο σε μια εικόνα ακόμη και αν αυτό έχει μετατοπιστεί. Υπάρχουν δυο είδη ομαδοποίησης το μέγιστο (Max) και ο μέσος όρος (Average).



Εικόνα 5 : Max Pooling and Average Pooling

Πηγή : [https://www.researchgate.net/figure/Average-versus-max-pooling_fig1_317496930]

Το μέγιστο επίπεδο ομαδοποίησης παίρνει τη μέγιστη τιμή από την περιοχή του χάρτη χαρακτηριστικών που εξετάζει και συγκεκριμένα το φωτεινότερο εικονοστοιχείο (pixel). Το μέσο επίπεδο ομαδοποίησης από την άλλη υπολογίζει και κρατάει το μέσο όρο από την περιοχή που εξετάζει.

Πλήρως συνδεδεμένα επίπεδα (Fully Connected Layer): Στο τέλος του δικτύου υπάρχουν τα πλήρως συνδεδεμένα επίπεδα τα οποία, έχουν και την ευθύνη να ταξινομήσουν τα δεδομένα, χρησιμοποιούν τα χαρακτηριστικά και τις πληροφορίες που έμαθαν τα προηγούμενα επίπεδα για να ταξινομήσουν την εικόνα εισόδου σε διαφορές κατηγορίες.

Εκτός από αυτά τα επίπεδα είναι καλό να κάνουμε μια αναφορά και σε δυο ακόμη δομικά στοιχεία των ΣΝΔ τα οποία η παρουσία τους είναι αρκετά σημαντική.

Επίπεδο κανονικοποίησης παρτίδας (Batch Normalization): Είναι ένα επίπεδο που χρησιμοποιείται στα νευρωνικά δίκτυα για να κάνει την εκπαίδευση τους ταχύτερη και πιο αποδοτική. Συνήθως τοποθετείτε μετά τα συνελκτικά ή τα πλήρως συνδεδεμένα επίπεδα. Το επίπεδο κανονικοποίησης παρτίδας λειτουργεί σαν σταθεροποιητής για

τα νευρωνικά δίκτυα εξασφαλίζοντας ότι ανεξάρτητα από το πως έχουν αλλάξει τα δεδομένα σε κάθε επίπεδο του δικτύου τα κανονικοποιεί ώστε να έχουν μέσο όρο 0 και διακύμανση 1. [10]

Επίπεδο αποκοπής νευρώνων (Dropout): Είναι ένα επίπεδο που εφαρμόζει μια απλή αλλά αποτελεσματική τεχνική που χρησιμοποιείται στην εκπαίδευση των νευρωνικών δικτύων για να αποτρέψει την υπερβολική προσαρμογή τους. Η υπερπροσαρμογή συμβαίνει όταν ένα δίκτυο μαθαίνει τα δεδομένα εκπαίδευσης πολύ καλά συν του θορύβου και των λεπτομερειών που δεν γενικεύονται σε νέα δεδομένα. Το dropout απενεργοποιεί τυχαία ορισμένους νευρώνες του δικτύου κατά τη διάρκεια της εκπαίδευσης. Αυτό σημαίνει ότι αυτοί οι νευρώνες δεν συμβάλλουν στην επεξεργασία των δεδομένων, ούτε συμμετέχουν στη μάθηση από τα δεδομένα κατά τη διάρκεια αυτών των βημάτων εκπαίδευσης. [11]

2.7 Residual Networks (ResNets)

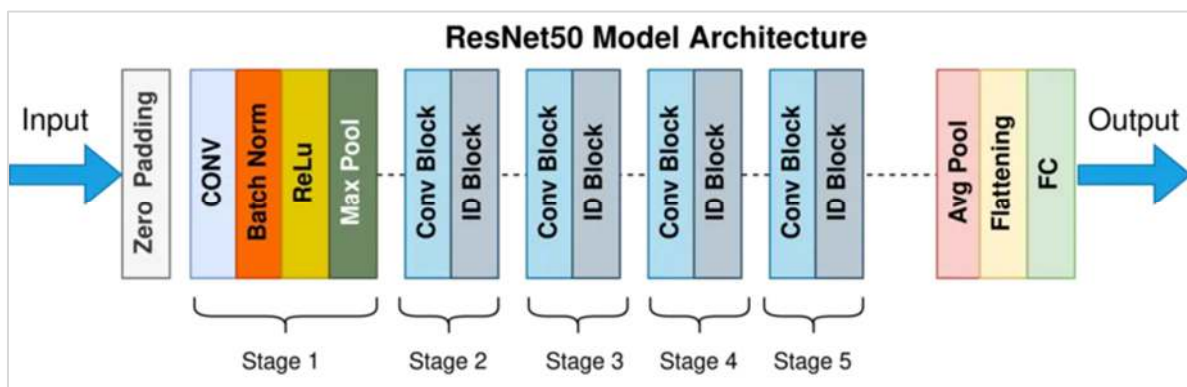
Τα Residual Networks (ResNets) αποτελούν μία κατηγορία βαθιών νευρωνικών δικτύων που έχουν σχεδιαστεί για να αντιμετωπίζουν το πρόβλημα της εξασθένησης της εκμάθησης σε πολύ βαθιά δίκτυα. Σύμφωνα με την πρόσφατη έρευνα των Xu, Fu, και Zhu (2023), τα Residual Networks (ResNet) έχουν δείξει σημαντική πρόοδο και επιτυχία στον τομέα της κλινικής βοηθητικής διάγνωσης σοβαρών ασθενειών όπως οι όγκοι των πνευμόνων, ο καρκίνος του μαστού, οι δερματικές παθήσεις και οι καρδιαγγειακές και εγκεφαλοαγγειακές παθήσεις. [12]

Ένα εξαιρετικό παράδειγμα αυτής της κατηγορίας είναι το μοντέλο ResNet-50, το οποίο είναι γνωστό για τη βαθιά αρχιτεκτονική του η οποία αποτελείται από 50 επίπεδα. Αυτό το βάθος παίζει σημαντικό ρόλο στην εκμάθηση συνθέτων μοτίβων στα δεδομένα, ιδιαίτερα σε εικόνες όπως οι ακτινογραφίες που αποτελούνται από περίπλοκες λεπτομέρειες που έχουν μεγάλη αξία για την ανάλυση και την κατανόηση τους.

Το μοντέλο ResNet-50 χρησιμοποιείται συχνά σε προ εκπαιδευμένη μορφή και πιο συγκεκριμένα η εκπαίδευση του έχει πραγματοποιηθεί σε ένα μεγάλο σύνολο

δεδομένων όπως είναι το ImageNet το οποίο περιέχει εκατομμύρια εικόνες με χιλιάδες ετικέτες. Αυτή η προ εκπαίδευση συμβάλει ιδιαίτερα σε προβλήματα όπου τα διαθέσιμα δεδομένα είναι περιορισμένα ή δεν είναι αρκετά ποικίλα ώστε να εκπαιδευτεί ένα βαθύ νευρωνικό δίκτυο από την αρχή και απαιτεί πολλούς πόρους, καθώς και ότι επιτρέπει στο δίκτυο την αξιοποίηση των χαρακτηριστικών που έχει ήδη μάθει από την προ εκπαίδευση του και την εφαρμογή τους σε διαφορετικούς τύπους εικόνων.

Η βασική δομή ενός ResNet-50 μοντέλου αποτελείται από τέσσερα βασικά τμήματα τα συνελικτικά επίπεδα (Convolutional Layers), τα Identify blocks, τα συνελικτικά blocks και τα πλήρως συνδεδεμένα επίπεδα.



Εικόνα 6 : Αρχιτεκτονική ResNet-50

Πηγή: <https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f>

Τα συνελικτικά επίπεδα επεξεργάζονται και εφαρμόζουν διάφορα φίλτρα στα δεδομένα εισόδου προκειμένου να εξάγουν κάποια χαρακτηριστικά και λεπτομέρειες όπως είναι οι ακμές, οι υφές και τα σχήματα. Αυτό κατορθώνεται μέσω μιας σειράς από συνελικτικά επίπεδα το καθένα από το οποίο ακολουθείται από ένα επίπεδο κανονικοποίησης παρτίδας (Batch Normalization) και μια συνάρτηση ενεργοποίησης ReLU. Μετά από αυτά τα επίπεδα ακολουθούν τα επίπεδα ομαδοποίησης όπως τα Max Pooling layers τα οποία χρησιμεύουν στη μείωση των διαστάσεων διασφαλίζοντας ότι θα διατηρηθούν μόνο τα πιο σημαντικά χαρακτηριστικά.

Τα Identify blocks και τα συνελικτικά blocks είναι από τα πιο βασικά δομικά στοιχεία του μοντέλου και αποτελούν μια καινοτομία. Τα Identify blocks λειτουργούν σαν μια

παράκαμψη για τα δεδομένα όταν ρέουν μέσα από το μοντέλο. Τα δεδομένα έχουν τη δυνατότητα να παρακάμψουν ορισμένα επίπεδα και να φτάσουν κατευθείαν στο τέλος του block. Αυτή η παράκαμψη είναι ιδιαίτερα χρήσιμη καθώς τα δεδομένα περνούν κανονικά από τα επίπεδα μαθαίνοντας και αλλάζοντας αλλά ένα αντίγραφο των αρχικών δεδομένων κινείται πιο γρήγορα και προστίθεται στο τέλος του block. Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη καθώς βοηθάει το δίκτυο να θυμάται τις παλιές πληροφορίες ενώ συνεχίζει να μαθαίνει νέα πράγματα.

Τα συνελικτικά blocks έχουν παρόμοια λειτουργικότητα με τα identify blocks με τη διάφορα ότι τα δεδομένα πριν παρακάμψουν τα επίπεδα και φτάσουν στο τέλος του block διέρχονται μέσα από ένα ειδικό επίπεδο (1x1 convolutional Layer) το οποίο προετοιμάζει μια πιο απλή εκδοχή των δεδομένων πριν τα προσθέσει ξανά στην κύρια διαδρομή τους. Και τα δυο αυτά δομικά στοιχεία είναι πολύ σημαντικά καθώς επιτρέπουν στο δίκτυο να διατηρήσει σημαντικές πληροφορίες από τα αρχικά δεδομένα ενώ παράλληλα αποκομίζει νέες γνώσεις από τις αλλαγές από τη ροή των δεδομένων από τα επίπεδα. Αυτή η μέθοδος αποτρέπει το δίκτυο από το να υπερφορτωθεί και να ξεχάσει αυτά που έμαθε νωρίτερα κάτι που αποτελεί κοινό πρόβλημα στα πολύ βαθιά δίκτυα.

Τα πλήρως συνδεδεμένα επίπεδα είναι αυτά που θα πάρουν και την τελική απόφαση και θα ταξινομήσουν τα δεδομένα. Η έξοδος από αυτά τα επίπεδα περνούν από μια συνάρτηση ενεργοποίησης, στην περίπτωση μας Sigmoid η οποία όπως προαναφέραμε και παραπάνω παίρνει τιμές μεταξύ 0 και 1. Η τιμή που παίρνει δεν είναι πάρα μια πιθανότητα που υποδηλώνει σε ποια κλάση ταξινομεί τα δεδομένα. Αν η πιθανότητα είναι κάτω από 0.5 τα δεδομένα στην περίπτωση μας η εικόνα ανήκει στην κλάση 0 (αρνητικό) και αν είναι πάνω από 0.5 στην κλάση 1 (θετικό). [13]

2.8 Μεταφορά Μάθησης(Transfer Learning)

Η μεταφορά μάθησης είναι μια μέθοδος της μηχανικής μάθησης όπου μας επιτρέπει να χρησιμοποιήσουμε προ εκπαιδευμένα μοντέλα σε νέα προβλήματα όπου ο όγκος των δεδομένων είναι μικρότερος. Αυτή η τεχνική είναι ιδανική όταν υπάρχει έλλειψη δεδομένων με ετικέτες και υπολογιστική ισχύ. Με αυτή τη μέθοδο μπορούμε να

εκμεταλλευτούμε την προηγούμενη μάθηση για να ενισχύσουμε τη γενίκευση σε νέα δεδομένα. Δεν είναι μόνο μια μέθοδος αλλά και μια πάρα πολύ καλή στρατηγική για τη σχεδίαση μοντέλων που μπορούν να επιτύχουν βελτιωμένες επιδόσεις με πιο αποτελεσματικό τρόπο στην επιστήμη δεδομένων. Ο τρόπος που λειτουργεί η μεταφορά μάθησης είναι ότι διατηρούμε τα αρχικά επίπεδα ενός νευρωνικού δικτύου που έχουν μάθει να αναγνωρίζουν γενικά χαρακτηριστικά όπως ακμές και σχήματα ενώ τα τελευταία επίπεδα ξανά εκπαιδεύονται για να προσαρμόζονται στα ειδικότερα χαρακτηριστικά του νέου μοντέλου. [14]

2.9 Συναρτήσεις Ενεργοποιήσεις

Η επιλογή της συνάρτησης ενεργοποίησης παίζει πάρα πολύ σημαντικό λόγο γιατί καθορίζει την έξοδο του νευρώνα. Οι πιο συνηθισμένες συναρτήσεις ενεργοποίησης περιλαμβάνουν τις Sigmoid, ReLU, Tanh και Softmax.

Sigmoid / Logistic Activation Function: Είναι μια μη γραμμική συνάρτηση ενεργοποίησης που χρησιμοποιείται ευρέως στα νευρωνικά δίκτυα, το χαρακτηριστικό της είναι ότι δέχεται οποιοδήποτε πραγματικό αριθμό και τον επιστρέφει σε μια τιμή μεταξύ 0 και 1 για αυτό την καθιστά κατάλληλη για μοντέλα που απαιτούν δυαδική ταξινόμηση ή εκτίμηση πιθανότητας. Ορίζεται από τον παρακάτω τύπο. [15]

$$f(x) = \frac{1}{1 + e^{-x}}$$

ReLU / Rectified Linear Unit: Είναι μια απλή και υπολογιστικά αποδοτική συνάρτηση ενεργοποίησης που εισάγει μη γραμμικότητα στο δίκτυο. Ο τρόπος που λειτουργεί είναι ότι για όλες τις αρνητικές τιμές εισόδου, η τιμή εξόδου είναι ίση με το 0 και για όλες τις θετικές τιμές εισόδου η τιμή εξόδου είναι ίση με την τιμή εισόδου. Με αυτόν τον τρόπο επιτρέπει στο νευρώνα να ενεργοποιείται όταν η τιμή είναι θετική και να απενεργοποιείται όταν η τιμή είναι αρνητική. Λόγω αυτής της λειτουργίας της η εφαρμογή της σε νευρωνικά δίκτυα με πολλά επίπεδα είναι ιδιαίτερα χρήσιμη γιατί κάθε αυτόν τον τρόπο μπορεί να κάνει το δίκτυο ελαφρύτερο και οι υπολογισμοί να γίνονται πιο γρήγορα. Ορίζεται από τον παρακάτω τύπο. [15]

$$f(x) = \max(0, x)$$

Tanh /Hyperbolic Tangent: Η συνάρτηση Tanh είναι μια μη γραμμική συνάρτηση και όπως και η Sigmoid δέχεται έναν οποιοδήποτε πραγματικό αριθμό και τον επιστρέφει σε μια τιμή μεταξύ -1 και 1 και για αυτό χρησιμοποιείται για δεδομένα που απαιτείται ταξινόμηση. Ο λόγος που μερικές φορές προτιμάται έναντι της Sigmoid είναι γιατί μπορεί να καταλάβει καλύτερα τις κλήσεις κατά τη διάρκεια του backpropagation οδηγώντας σε πιο γρήγορη μάθηση σε κάποιες περιπτώσεις. Ορίζεται από τον παρακάτω τύπο . [15]

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Softmax: Η συγκεκριμένη συνάρτηση ενεργοποίησης συνηθίζεται να χρησιμοποιείται στα νευρωνικά δίκτυα για προβλήματα ταξινόμησης πολλαπλών κλάσεων. Ο τρόπος που λειτουργεί είναι ότι λαμβάνει ένα διάνυσμα πραγματικών αριθμών ως είσοδο και το κανονικοποιεί σε μια κατανομή πιθανότητας σε πολλαπλές κλάσεις. Επίσης, εξασφαλίζει ότι οι πιθανότητες εξόδου αθροίζουν στο 1. Ορίζεται από τον παρακάτω τύπο. [15]

$$\sigma(Z_j) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}, \text{ για } j = 1, \dots, K$$

2.10 Συναρτήσεις Κόστους

Οι συναρτήσεις κόστους ή οι συναρτήσεις απώλειας όπως είναι γνωστές παίζουν ένα πολύ σημαντικό ρόλο στη μηχανική και βαθιά μάθηση καθώς χρησιμοποιούνται για να εκτιμηθεί πόσο καλή επίδοση έχει ένα νευρωνικό δίκτυο. Στην πραγματικότητα είναι ένα μέτρο που μετράει την απόσταση μεταξύ των προβλεπόμενων τιμών του μοντέλου και των πραγματικών τιμών στα δεδομένα εκπαίδευσης.

Υπάρχουν διαφορετικών ειδών συναρτήσεις κόστους και ανάλογα με τον τύπο του προβλήματος χρησιμοποιείται η πιο κατάλληλη. Ωστόσο, θα μπορούσαμε να τις χωρίσουμε σε δυο μεγάλες κατηγορίες. Συναρτήσεις απώλειας που εφαρμόζονται σε προβλήματα παλινδρόμησης , όπου το μοντέλο προσπαθεί να προβλέψει μια συνεχή

τιμή (όπως είναι οι χρονοσειρές) και η δεύτερη κατηγορία είναι για τα προβλήματα κατηγοριοποίησης όπου το μοντέλο προσπαθεί να προβλέψει πόσο πιθανόν είναι ένα δεδομένο να ανήκει σε μια κατηγορία.

Mean Squared Error (MSE): Το μέσο τετράγωνο σφάλμα χρησιμοποιείται για προβλήματα που προσπαθούμε να προβλέψουμε μια συνεχή τιμή. Υπολογίζεται από τη μέση τετραγωνική διαφορά μεταξύ των προβλεπόμενων τιμών και των πραγματικών τιμών. Ορίζεται από τον παρακάτω τύπο. [16]

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Mean Absolute Error (MAE): Το μέσο απόλυτο σφάλμα όπως και το μέσο τετράγωνο σφάλμα χρησιμοποιείται για προβλήματα συνεχών τιμών. Υπολογίζεται από το άθροισμα των απόλυτων σφαλμάτων δια το μέγεθος του δείγματος. Ορίζεται από τον παρακάτω τύπο. [16]

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

Binary Cross-Entropy Loss: Χρησιμοποιείται σε προβλήματα ταξινόμησης για δεδομένα που ανήκουν σε δυο κλάσεις. Όπου t_i είναι η πραγματική κλάση και s_1 είναι η προβλεπόμενη πιθανότητα ότι ανήκει στην κλάση. Η δυαδική συνάρτηση απώλειας τιμωρεί περισσότερο το μοντέλο όταν η προβλεπόμενη πιθανότητα έχει μεγάλη διαφορά από την αληθινή κλάση. [17]

$$BinaryCrossEntropyLoss = - \sum_{i=1}^{c=2} t_i \log(s_i) = -(t_1 \log(s_1) + (1 - t_1) \log(1 - s_1))$$

Categorical Cross-Entropy Loss: Χρησιμοποιείται σε προβλήματα ταξινόμησης για δεδομένα που ανήκουν από δυο κλάσεις και πάνω. Πιο συγκεκριμένα οι ετικέτες αναπαρίστανται σε ένα διάνυσμα μεγέθους C όπου τα στοιχεία του διανύσματος παίρνουν την τιμή 0, εκτός από το στοιχείο του διανύσματος που δείχνει την κλάση στην οποία ανήκει το δεδομένο. [17]

$$Categorical Cross - Entropy Loss = - \log \frac{e^{s_p}}{\sum_j^C e^{s_j}}$$

2.11 Βελτιστοποιητής (Optimizer)

Με τον όρο βελτιστοποιητή (optimizer) στη μηχανική μάθηση και ιδιαίτερα στα νευρωνικά δίκτυα εννοούμε διάφορους αλγορίθμους ή μεθόδους οι οποίοι χρησιμοποιούνται για την ελαχιστοποίηση της συνάρτησης κόστους. Αυτό επιτυγχάνεται μέσω της ρύθμισης των βαρών και του ρυθμού μάθησης (learning rate).

Ο ρυθμός μάθησης είναι η παράμετρος η οποία καθορίζει τον ρυθμό με τον οποίο θα βελτιώνονται τα βάρη σε κάθε επανάληψη μέχρι να βρεθεί το τοπικό ελάχιστο. Μια μεγάλη τιμή στον ρυθμό μάθησης μπορεί να προσπεράσει το τοπικό ελάχιστο σε κάθε επανάληψη και μια πολύ μικρή τιμή στον ρυθμό μάθησης μπορεί να αργήσει να το εντοπίζει. Μια καλή πρακτική είναι να ξεκινάμε με μια αρχική τιμή στον ρυθμό μάθησης και αν η συνάρτηση απώλειας δεν βελτιώνεται για κάποιες εποχές να το μειώνουμε κατά ένα ποσοστό.

Ο τρόπος που λειτουργεί ένας optimizer μπορεί να χωριστεί σε δυο βήματα. Πρώτα υπολογίζει την κλίση της συνάρτησης κόστους και βρίσκει προς ποια κατεύθυνση πρέπει να προσαρμοστούν τα βάρη για να ελαχιστοποιηθεί το κόστος και στη συνέχεια ενημερώνει τα βάρη του δικτύου προς την αντίθετη κατεύθυνση από την κλίση. Υπάρχουν διάφοροι optimizers τους οποίους μπορούμε να χρησιμοποιήσουμε και η επιλογή καθορίζει την απόδοση ενός μοντέλου σημαντικά γιατί ο κάθε optimizer ακολουθεί και διαφορετικό τρόπο αντιμετώπισης του προβλήματος. Μερικοί από τους πιο σύνηθες που μπορεί να συναντήσουμε είναι οι παρακάτω :

Gradient Descent: Είναι η απλούστερη μορφή αλγορίθμου και από τους πιο δημοφιλείς optimizer και πολλοί από τους optimizers βασίζονται πάνω σε αυτόν. Δίνεται από τον τύπο :

$$\theta_{t+1} = \theta_t - \eta_t \nabla F(\theta_t) ,$$

όπου η_t είναι ο ρυθμός μάθησης και $\nabla F(\theta_t)$ είναι η κλίση της συνάρτησης κόστους για την t επανάληψη. [18]

Ο Gradient Descent επειδή υπολογίζει την απώλεια και την κλίση για όλο το σύνολο δεδομένων σε κάθε επανάληψη αυτό συνεπάγεται ότι για να μπορέσει να συγκλίνει στο τοπικό ελάχιστο μπορεί να χρειάζεται πάρα πολύς χρόνος.

Stochastic Gradient Descent (SGD): Αποτελεί μια παραλλαγή του Gradient Descent και διαφοροποιείται ως προς τον τρόπο που υπολογίζει την κλίση. Ο SGD υπολογίζει

την κλίση χρησιμοποιώντας ένα παράδειγμα εκπαίδευσης κάθε φορά και έτσι ενημερώνει τις παραμέτρους πιο συχνά με αποτέλεσμα να συγκλίνει πιο γρήγορα. Ωστόσο, επειδή ενημερώνει συχνά τις παραμέτρους, οι παράμετροι έχουν υψηλές διακυμάνσεις και αυτό μπορεί να οδηγήσει σε υψηλές διακυμάνσεις στη συνάρτησή κόστους και να οδηγήσει σε αργή σύγκλιση. [19]

$$\theta = \theta - \eta \nabla F(\theta; x(i); y(i)),$$

όπου $\{(x(i), y(i))\}$, είναι τα παραδείγματα εκπαίδευσης

Mini Batch Gradient Descent: Αποτελεί και αυτό μια παραλλαγή του Gradient Descent και χρησιμοποιεί ένα υποσύνολο των δεδομένων (Mini Batch) για να υπολογίζει τη συνάρτηση απώλειας και έτσι οι παράμετροι ενημερώνονται μετά από κάθε παρτίδα δεδομένων. Επιπλέον, παρότι ενημερώνει συχνά τις παραμέτρους, η συνάρτηση απώλειας έχει χαμηλή διακύμανση και οδηγείται πιο σταθερά σε σύγκλιση. [19]

$$\theta = \theta - \eta \nabla F(\theta; B(i))$$

όπου $\{B(i)\}$ είναι οι παρτίδες των παραδειγμάτων εκπαίδευσης

Momentum: Λαμβάνει υπόψιν τις προηγούμενες κλίσεις και έτσι μειώνει την υψηλή διακύμανση και επιταχύνει τη σύγκλιση της συνάρτησης απώλειας. [19]

$$V(t) = \gamma V(t-1) + \eta \nabla F(\theta),$$

τα βάρη ανανεώνονται με $\theta = \theta - V(t)$

AdaGrad: Είναι αρκετά πιο διαφορετικός από τους προηγούμενους αλγόριθμους βελτιστοποίησης που αναφέραμε. Οι προηγούμενοι αλγόριθμοι έχουν σταθερό ρυθμό εκμάθησης ενώ ο AdaGrad χρησιμοποιεί διαφορετική τιμή στον ρυθμό μάθησης και αυτό εξαρτάται από το πόσο αλλάζουν οι παράμετροι κατά τη διάρκεια της εκπαίδευσης, όσο λιγότερο αλλάζουν οι παράμετροι τόσο λιγότερο αλλάζει και ο ρυθμός μάθησης. Στα θετικά του μπορούμε να συνυπολογίσουμε ότι δεν χρειάζεται να θέσουμε εμείς τον ρυθμό μάθησης. Στα αρνητικά του είναι ότι ο ρυθμός μάθησης μειώνεται τόσο πολύ που το δίκτυο σταματάει να μαθαίνει. [19]

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i},$$

όπου $g_{t,i}$ μια παράγωγος της συνάρτησης απώλειας για συγκριμένες παραμέτρους

$$i \text{ στην επανάληψη } t, \quad g_{t,i} = \nabla_{\theta} f(\theta_{t,i})$$

AdaDelta: Είναι μια καλύτερη εκδοχή του AdaGrad καθώς επιδιώκει να μειώσει την ανεξέλεγκτη φθίνουσα τιμή που μπορεί να πάρει ο ρυθμός μάθησης. Αυτό επιτυγχάνεται με τον περιορισμό των συσσωρεύσεων των προηγούμενων κλίσεων σε μια σταθερή τιμή w . [20]

$$\theta_{t+1} = \theta_t - \Delta \theta_t$$

RMSProp: Στόχος του είναι να προσαρμόζει τον ρυθμό μάθησης κατά την εκπαίδευση και αυτό το καταφέρνει διατηρώντας το μέσο όρο των τετράγωνων των κλίσεων. Ο ρυθμός μάθησης επηρεάζεται από αυτό το μέσο όρο και ο ρυθμός μάθησης είναι προσαρμοσμένος για κάθε βάρος σύμφωνα με τις προηγούμενες κλίσεις τους. [20]

$$\theta_{t+1} = \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Adam: Είναι ο αλγόριθμος που χρησιμοποιείται πιο συχνά καθώς συνδυάζει τις θετικές λειτουργίες των αλγορίθμων Momentum και RMSProp. Λαμβάνει υπόψιν του τις προηγούμενες κλίσεις και τα τετράγωνα των κλίσεων και υπολογίζει προσαρμοστικούς ρυθμούς μάθησης για κάθε παράμετρο. [19]

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{U}_t + \epsilon}} \hat{m}_t$$

2.12 Αξιολόγηση της Απόδοσης του Μοντέλου

Κατά τη διάρκεια της ανάπτυξης ενός μοντέλου μηχανικής μάθησης χρησιμοποιούμε διάφορες μετρικές για να μπορέσουμε να αξιολογήσουμε το μοντέλο μας. Οι μετρικές στις οποίες θα κάνουμε αναφορά είναι :

- Ακρίβεια ή ορθότητα (Accuracy)
- Ακρίβεια (Precision)
- Ανάκληση ή ευαισθησία (Recall)
- ο αρμονικός μέσος όρος της ακρίβειας (Precision) και της ανάκλησης (f1- Score)

Για να μπορέσουμε να κατανοήσουμε καλύτερα πως υπολογίζονται αυτές οι μετρικές, μπορούμε να δούμε τον παρακάτω πίνακα σύγκρισης ή σφαλμάτων.

		Predicted Label	
		Positive	Negative
Actual Label	Positive	TRUE POSITIVE TP	FALSE NEGATIVE FN
	Negative	FALSE POSITIVE FP	TRUE NEGATIVE TN

Εικόνα 7 : Πίνακας Σφαλμάτων

Πηγή: [https://www.researchgate.net/figure/Example-of-confusion-matrix-True-Positive-TP-The-number-of-instances-that-a-model_fig1_329526806]

Accuracy είναι ο λόγος των σωστά προβλεπόμενων παρατηρήσεων προς το σύνολο των παρατηρήσεων.

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

Precision είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς το σύνολο των προβλεπόμενων θετικών παρατηρήσεων.

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

Recall είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς όλες τις παρατηρήσεις της πραγματικής κλάσης.

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

Και τέλος F1 -Score είναι ο σταθμισμένος μέσος όρος του Precision και Recall , λαμβάνοντας υπόψιν τόσο τα ψευδώς θετικά όσο και τα ψευδώς αρνητικά.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Οι μετρικές που αναλύσαμε, χρησιμοποιούνται στα διάφορα στάδια της ανάπτυξης του μοντέλου όπως είναι το στάδιο της εκπαίδευσης, το στάδιο της επικύρωσης και το στάδιο της δοκιμής.

Στο στάδιο της εκπαίδευσης παρακολουθούμε αυτές τις μετρικές για να κατανοήσουμε πόσο καλά μαθαίνει το μοντέλο. Ωστόσο, δεν αποτελούν το βασικό μας μέλημα κατά τη διάρκεια της εκπαίδευσης καθώς μπορεί να οδηγηθούμε σε λάθος συμπεράσματα. Αν κατά τη διάρκεια της εκπαίδευσης παίρνουμε ικανοποιητικές τιμές πολύ πιθανόν να σημαίνει και υπερπροσαρμογή του μοντέλου μας δηλαδή ότι το μοντέλο έχει μάθει πολύ καλά τα δεδομένα εκπαίδευσης αλλά δεν μπορεί να αποδώσει καλά σε δεδομένα που δεν γνωρίζει. Για αυτό και ο κύριος στόχος μας είναι η ελαχιστοποίηση της συνάρτησης απώλειας.

Στο στάδιο της επικύρωσης σε αντίθεση με το στάδιο της εκπαίδευσης οι μετρικές αυτές παίζουν πιο σημαντικό ρόλο. Καθώς το μοντέλο αξιολογείται σε δεδομένα που δεν γνωρίζει και μας δίνεται η δυνατότητα να εντοπίσουμε αν υπάρχουν προβλήματα υπερπροσαρμογής και υποπροσαρμογής. Έτσι μπορούμε να οδηγηθούμε σε αλλαγές των υπερπαραμέτρων του μοντέλου και τη λήψη αποφάσεων για την αρχιτεκτονική του μοντέλου.

Στο στάδιο του ελέγχου είναι ιδιαίτερα ζωτικής σημασίας να δώσουμε έμφαση σε αυτές τις μετρικές καθώς το μοντέλο έχει περάσει από τα δυο παραπάνω στάδια που προαναφέραμε και είναι η στιγμή που το μοντέλο θα δούμε πως θα αντιδράσει σε δεδομένα που του είναι άγνωστα.

Στο πλαίσιο της συγκεκριμένης μελέτης είναι αρκετά σημαντικό να δώσουμε μια βαρύτητα παραπάνω στις μετρικές που θα χρησιμοποιήσουμε καθώς όλες οι μετρικές δεν είναι κατάλληλες για την αξιολόγηση της απόδοσης ενός μοντέλου και είναι πάρα πολύ παράμετροι που πρέπει να λάβουμε υπόψιν μας.

Η ακρίβεια (Accuracy) σε ένα σύνολο δεδομένων που δεν είναι ισορροπημένο δηλαδή να υπάρχουν περισσότερα δεδομένα που ανήκουν στη μια κλάση και όχι στην άλλη μπορεί να μην είναι αξιόπιστη, καθώς το μοντέλο μπορεί να φαίνεται ακριβές αλλά στην πραγματικότητα να προβλέπει την κλάση που ανήκουν τα περισσότερα δεδομένα.

Η ακρίβεια (Precision) ωστόσο είναι ιδιαίτερα σημαντική σε δεδομένα που έχουν σχέση με την ιατρική απεικόνιση καθώς μετράει το ποσοστό των περιπτώσεων που το μοντέλο πρόβλεψε ως θετικά και είναι πραγματικά θετικά προς όλες τις περιπτώσεις που το μοντέλο τις χαρακτήρισε ως θετικές. Μια υψηλή τιμή θα μπορούσε στην περίπτωση μας να σημαίνει ότι το μοντέλο έκανε ακριβής διάγνωση χωρίς πολλές λάθος θετικές διαγνώσεις.

Η ανάκληση (Recall) είναι επίσης σημαντική όταν μελετάμε δεδομένα που έχουν να κάνουν με ιατρική διάγνωση γιατί το κόστος των πραγματικά θετικών που πρόβλεψε ως αρνητικά είναι ιδιαίτερα σημαντικό. Ένα υψηλό ποσοστό στην ανάκληση μας εξασφαλίζει ότι ένα μεγάλο σύνολο από τις διαγνώσεις που έκανε τις αναγνώρισε σωστά ως πραγματικά θετικές.

Το f1-Score επειδή όπως προαναφέραμε είναι ο σταθμισμένος μέσος της ακρίβειας (Precision) και της ανάκλησης και επειδή όταν τα δεδομένα που μελετάμε έχουν σχέση με τη διάγνωση και μας αφορά εάν το μοντέλο προβλέπει τόσο τα ψευδώς θετικά όσο και τα ψευδώς αρνητικά είναι μια από τις βασικές μετρικές που θα λάβουμε υπόψιν μας για να αξιολογήσουμε τα μοντέλα μας. Μια υψηλή τιμή του f1-Score μας δείχνει ότι το μοντέλο δεν είναι μόνο ακριβές αλλά και ισορροπημένο ως προς την ικανότητα του να προβλέπει σωστά τις θετικές διαγνώσεις και να ελαχιστοποιεί τις ψευδώς θετικά προβλέψεις. Επιπλέον, σε περιπτώσεις που προαναφέραμε όπου τα δεδομένα δεν είναι ισορροπημένα η ακρίβεια (Precision) και η ανάκληση είναι πιο σημαντικές μετρικές από την ακρίβεια (Accuracy). Το f1-score ως σταθμισμένος μέσος ορός αυτών των δυο είναι συνήθως μια πάρα πολύ καλή επιλογή για την αξιολόγηση της απόδοσης ενός μοντέλου.

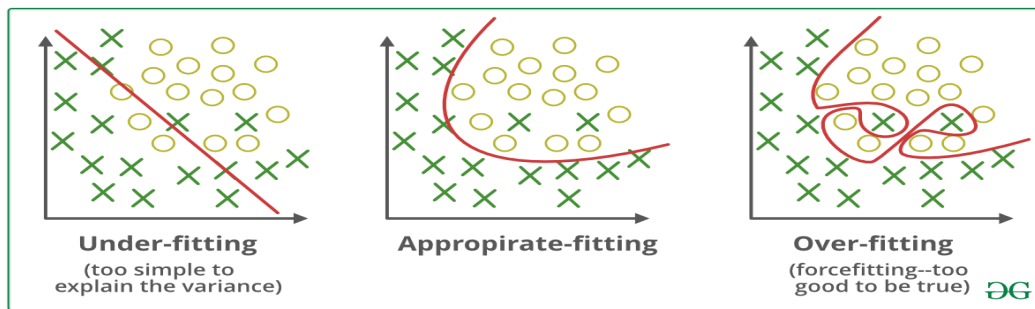
2.13 Υπερπροσαρμογή & Υποπροσαρμογή

Η υπερπροσαρμογή (Overfitting) και η υποπροσαρμογή (underfitting) είναι δυο από τα βασικά προβλήματα που έχουμε να αντιμετωπίσουμε στον τομέα της μηχανικής μάθησης και ιδιαίτερα στα συνελκτικά νευρωνικά δίκτυα.

Με τον όρο υπερπροσαρμογή αναφερόμαστε στην κατάσταση στην οποία το μοντέλο έχει μάθει πολύ καλά τα δεδομένα εκπαίδευσης και αποδίδει εξαιρετικά αλλά σε νέα άγνωστα για αυτό δεδομένα η απόδοση του δεν είναι η αναμενόμενη. Αυτό είναι εύκολο να το διαπιστώσουμε από τη μεγάλη διαφορά που μπορεί να έχει η ακρίβεια (accuracy) στην εκπαίδευση και στην επικύρωση. Όπως και από τη συνάρτηση απώλειας της εκπαίδευσης, η οποία θα πρέπει να μειώνεται ταυτόχρονα με τη συνάρτηση απώλειας της επικύρωσης. Αν εντοπίσουμε ότι η συνάρτηση απώλειας στην επικύρωση δεν αλλάζει αυτό είναι το σημείο που καταλαβαίνουμε ότι ξεκινάει το μοντέλο να μαθαίνει ασήμαντες λεπτομέρειες και πρέπει να τερματίσουμε την εκπαίδευση του.

Συνήθως είναι πιο πιθανόν να το συναντήσουμε σε μοντέλα όπου διενεργούνται μη γραμμικοί μετασχηματισμοί, όπως και τα μοντέλα που θα μελετήσουμε εξαιτίας της αυξημένης ευελιξίας που έχουν κατά το στάδιο της εκπαίδευσης. Οι αίτιες που μπορεί να συμβαίνει αυτό και θα πρέπει να εξετάσουμε είναι αν τα δεδομένα μας είναι επαρκή για να γίνει η εκπαίδευσή σωστά αλλιώς θα πρέπει να εφαρμοστούν τεχνικές επαύξησης των δεδομένων ή ακόμη και να ενισχύσουμε το σύνολο δεδομένων μας με καινούργια δεδομένα στο στάδιο της εκπαίδευσης. Επιπλέον, θα πρέπει να εξετάσουμε την πολυπλοκότητα του μοντέλου μας. Η μείωση των επιπέδων καθώς και των παραμέτρων θα μας δώσει ένα πιο απλό μοντέλο και μπορεί να συμβάλλει στην αποφυγή της υπερπροσαρμογής. Άλλες τεχνικές που μπορούμε να εφαρμόσουμε είναι η κανονικοποίηση, η προσθήκη ενός επιπέδου dropout ή ενός επιπέδου Batch Normalization τα οποία μπορούν να βοηθήσουν στην αποφυγή της υπερπροσαρμογής. [21] [22]

Σε αντίθεση με την υπερπροσαρμογή η υποπροσαρμογή εμφανίζεται όταν παρατηρούμε ότι το μοντέλο μας έχει πολύ κακές επιδόσεις και στα δεδομένα εκπαίδευσης και στα δεδομένα επικύρωσης και αυτό είναι ιδιαίτερα χρήσιμο γιατί μπορούμε να την αντιληφθούμε εύκολα. Η υποπροσαρμογή μπορεί να οφείλεται και σε τεχνικές που ακολουθήσαμε ώστε να αποφύγουμε την υπερπροσαρμογή όπως είναι η κανονικοποίηση αλλά και σε ένα πολύ απλό μοντέλο για τα δεδομένα που εξετάζουμε το οποίο δεν είναι σε θέση να μάθει σημαντικές πληροφορίες για τα δεδομένα. Αρκετά συχνό πρόβλημα το οποίο μπορεί να οδηγήσει στην υποπροσαρμογή είναι και το σύνολο δεδομένων το οποίο είτε μπορεί να μην είναι αρκετό, είτε μπορεί να μην είναι κατάλληλο. Τρόποι για να αντιμετωπίσουμε την υποπροσαρμογή είναι να αυξήσουμε την πολυπλοκότητα του μοντέλου μας προσθέτοντας περισσότερα επίπεδα ή νευρώνες, η αύξηση των δεδομένων εκπαίδευσης ώστε να έχει περισσότερα παραδείγματα για να μπορέσει να κατανοήσει καλύτερα τα μοτίβα, η αύξηση της διάρκειας εκπαίδευσης ώστε να του δοθεί χρόνος να μάθει, καθώς και μια πιο ισορροπημένη προσέγγιση διαχείρισης και της υπερπροσαρμογής αλλά και της υποπροσαρμογής γιατί όπως αναφέραμε στην περίπτωση που προσπαθούμε να αντιμετωπίσουμε το ένα πρόβλημα μπορεί να προκαλέσουμε το άλλο. [21]



Εικόνα 8 : Οπτική Απεικόνιση Υπερπροσαρμογής & Υποπροσαρμογής

Πηγή : [\[https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/\]](https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/)

Στην Εικόνα 8 παρουσιάζονται τρία διαγράμματα που απεικονίζουν τα φαινόμενα της υπερπροσαρμογής, της υποπροσαρμογής αλλά και της επιθυμητής προσαρμογής στην εκπαίδευση. Όπως φαίνεται στο πρώτο διάγραμμα η γραμμή δεν ακολουθεί επαρκώς τη διακύμανση των δεδομένων γεγονός που δείχνει ότι υπάρχει υπερπροσαρμογή και στο τελευταίο διάγραμμα βλέπουμε ότι η γραμμή ακολουθεί πολύ στενά τα δεδομένα γεγονός που φανερώνει ότι το μοντέλο έχει απομνημονεύσει τα δεδομένα. Στο μέσο διάγραμμα φαίνεται ότι η γραμμή ταιριάζει καλά με τα δεδομένα υποδεικνύοντας ότι έχουμε ένα καλά προσαρμοσμένο μοντέλο.

2.14 Μέθοδος Οπισθοδιάδοσης (Backpropagation)

Η οπισθοδιάδοση από την άνοδο της δημοτικότητας μετά το 1986 είναι μια πάρα πολύ δημοφιλής τεχνική στην εκπαίδευση των νευρωνικών δικτύων. Η απλότητα και η αποτελεσματικότητά της στην εκπαίδευση μη γραμμικών, πολύπλοκων δικτύων την καθιστούν ιδιαίτερα χρήσιμη. Η σύγχρονη μορφή της, που έχει τις ρίζες της σε εργασίες της δεκαετίας του 1960, αναπτύχθηκε ανεξάρτητα από τους Werbos, Parker και Rumelhart γύρω στις αρχές της δεκαετίας του 1980. Περιλαμβάνει την προσαρμογή των τιμών των βαρών σε ένα δίκτυο με στόχο την ελαχιστοποίηση του σφάλματος και την καλύτερη απόδοση του. [23]

Η οπισθοδιάδοση αποτελείται από δυο στάδια, το στάδιο της εμπρόσθιας διάδοσης όπου τα δεδομένα περνούν μέσα από τα επίπεδα του δικτύου και φτάνουν στο επίπεδο όπου γίνεται η ταξινόμηση, εκεί υπολογίζεται η απώλεια δηλαδή το ποσό λάθος ήταν

η ταξινόμηση του δικτύου. Και στο στάδιο της αντίστροφης διάδοσης όπου υπολογίζεται η κλίση της συνάρτησης απώλειας σε σχέση με κάθε βάρος. Αρχίζει με το επίπεδο εξόδου και προχωράει προς τα πίσω ενημερώνοντας τα βάρη ώστε να βελτιώσει σταδιακά την απόδοση του δικτύου.

Μια πάρα πολύ σημαντική απόφαση είναι η επιλογή του κατάλληλου ρυθμού μάθησης, καθώς μπορεί ένας πολύ υψηλός ρυθμός μάθησης να προκαλέσει την πρόωρη σύγκλιση του μοντέλου και να το οδηγήσει σε λανθασμένες λύσεις ενώ ένας πολύ χαμηλός ρυθμός μάθησης μπορεί να ελαττώσει τη διαδικασία μάθησης.

Η οπισθοδιάδοση είναι επίσης μια κατάλληλη τεχνική για αρχιτεκτονικές βαθιάς μάθησης και ιδανική για περιπτώσεις που υπάρχει μεγάλος όγκος δεδομένων. Έχει την ικανότητα να χειρίζεται με αποτελεσματικότητα πολύπλοκες μη γραμμικές σχέσεις στα δεδομένα και για αυτό το λόγο τη συναντάμε σε εργασίες που έχουν να κάνουν με επεξεργασία δεδομένων εικόνας καθώς και άλλους πολλούς τομείς της τεχνητής νοημοσύνης. [23]

2.15 Εξαφάνιση και Εκτόξευση της κλίσης

Η εξαφάνιση (Vanishing) και η εκτόξευση (Exploding) της κλίσης είναι δυο πολύ συχνά προβλήματα που συναντάμε κατά την εκπαίδευση βαθιών νευρωνικών δικτύων και ειδικά στις περιπτώσεις που χρησιμοποιείται η μέθοδος της οπισθοδιάδοσης για τον υπολογισμό των κλίσεων.

Η εξαφάνιση της κλίσης συμβαίνει όταν οι κλίσεις των βαρών του δικτύου γίνονται τόσο μικρές ώστε να προσεγγίζουν το μηδέν. Αυτό προκαλείται όταν γίνεται η διάδοση της κλίσης του σφάλματος σε προηγούμενα επίπεδα και ο επαναλαμβανόμενος πολλαπλασιασμός έχει ως αποτέλεσμα η κλίση να γίνεται τόσο μικρή που να μην έχει κανένα νόημα. Αυτό έχει ως αποτέλεσμα τα βάρη στα προηγούμενα επίπεδα του δικτύου να αλλάζουν ελάχιστα και τα επίπεδα να μαθαίνουν πολύ αργά ή και καθόλου. Ειδικά όταν δουλεύουμε σε βαθιά δίκτυα με συναρτήσεις ενεργοποίησης όπως είναι η Sigmoid και η Tanh που έχουν ένα πολύ μικρό εύρος τιμών που μπορούν να πάρουν οι τιμές εισόδου. [24]

Η Εκτόξευση της κλίσης είναι ακριβώς το αντίθετο πρόβλημα που έχουμε να αντιμετωπίσουμε. Οι κλίσεις γίνονται υπερβολικά μεγάλες καθώς αυξάνονται εκθετικά κατά τη διάρκεια της οπισθοδιάδοσης εξαιτίας του επαναλαμβανόμενου πολλαπλασιασμού. Αυτό έχει ως αποτέλεσμα η διαδικασία της μάθησης να γίνεται ασταθής και το μοντέλο να αποκλίνει ή να μη συγκλίνει. [24]

Τρόποι για να μπορέσουμε να αντιμετωπίσουμε αυτά τα προβλήματα μπορεί να είναι η αλλαγή της συνάρτησης ενεργοποίησης και της χρήση της συνάρτησης ενεργοποίησης ReLU (Rectified Linear Unit), η αποκοπή της κλίσης όταν υπερβούν κάποιο όριο για να μπορέσουμε να αποφύγουμε την εκτόξευση ή την εξαφάνιση της κλίσης. Επίσης, η εφαρμογή της τεχνικής Batch Normalization μπορεί να βοηθήσει ώστε να σταθεροποιηθεί η διαδικασία μάθησης καθώς και οι συνδέσεις παράκαμψης που υπάρχουν σε αρχιτεκτονικές όπως το ResNet βοηθούν στη μείωση του προβλήματος της εξαφανιζόμενης κλίσης διευκολύνοντας τη ροή της πληροφορίας μέσα στο δίκτυο. [24]

ΚΕΦΑΛΑΙΟ 3: ΜΕΘΟΔΟΛΟΓΙΑ

Σε αυτή την ενότητα θα αναφερθούμε εκτενώς στη μεθοδολογία και τα βήματα που έγιναν για την υλοποίηση αυτής της διπλωματικής εργασίας. Για αρχή θα γίνει μια πρώτη παρουσία των εργαλείων και βιβλιοθηκών που επιλέχθηκαν. Στη συνέχεια θα γίνει μια παρουσίαση των δεδομένων και των τεχνικών επεξεργασίας των εικόνων που πραγματοποιήθηκαν καθώς και θα αναλύσουμε την αρχιτεκτονική των μοντέλων όπως διαμορφώθηκαν πριν την εκπαίδευση τους. Τέλος θα παρουσιάσουμε και θα αναλύσουμε τα αποτελέσματα που προέκυψαν μετά την εκπαίδευση των μοντέλων.

3.1 Βιβλιοθήκες και Εργαλεία

3.1.1 Python

Η γλώσσα προγραμματισμού που επιλέχθηκε για την υλοποίηση αυτής της εργασίας είναι η Python 3. Η επιλογή της γλώσσας δεν ήταν κάτι που αποτέλεσε μεγάλο δίλημμα γιατί είναι μια γλώσσα προγραμματισμού που φημίζεται για την απλότητά της και την καθιστά προσιτή τόσο σε αρχάριους όσο και σε πιο ειδικούς. Έκτος από αυτή την ευκολία η Python διαθέτει πληθώρα βιβλιοθηκών τόσο για τη διαχείριση δεδομένων όπως είναι η NumPy και η Pandas όσο και βιβλιοθήκες που είναι ειδικά σχεδιασμένες για εφαρμογές νευρωνικών δικτύων και απλοποιούν σημαντικά τη διαδικασία.

3.1.2 TensorFlow

Το TensorFlow είναι μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα (open source) που αναπτύχθηκε από την ομάδα της Google Brain και χρησιμοποιείται σε διαφορετικά πεδία της τεχνητής νοημοσύνης όπως είναι η βαθιά μάθηση και κατ' επέκταση τα νευρωνικά δίκτυα. Είναι ιδιαίτερα χρήσιμη βιβλιοθήκη καθώς δίνει τη δυνατότητα να αναπτυχθούν σύνθετα μοντέλα με την ευέλικτη αρχιτεκτονική της. Το TensorFlow προσφέρει ένα API υψηλού επιπέδου όπως το keras, το οποίο είναι ιδιαίτερα εύχρηστο λόγω της απλότητας του για την κατασκευή και την εκπαίδευση μοντέλων νευρωνικών δικτύων. Ο πυρήνας του είναι γραμμένος C++ και αυτό το κάνει να είναι πιο αποτελεσματικό και γρήγορο. Σαν βασική ιδέα είναι ότι αναλαμβάνει το

βάρος των υπολογισμών δίνοντας την ελευθερία στον προγραμματιστή να ασχοληθεί με τη δομή και τον σχεδιασμό του μοντέλου ανάλογά με το θέμα.

3.1.3 Pandas

Το Pandas αποτελεί και αυτό μια πολύ βασική και διαδεδομένη βιβλιοθήκη της Python και ιδιαίτερα για την επεξεργασία και την ανάλυση δεδομένων. Έχει σχεδιαστεί ειδικά για την τον χειρισμό ,τον μετασχηματισμό και την ανάλυση δεδομένων. Αυτό το στοιχείο που κάνει το Pandas ιδιαίτερα δημοφιλή είναι το DataFrame που είναι μια δισδιάστατη δομή σε μορφή πίνακα με γραμμές και στήλες και μπορεί να αποτελείται και από ετερογενή δεδομένα. Μοιάζει περισσότερο με έναν πίνακα μιας βάσης δεδομένων η ένα αρχείο csv. Το Pandas είναι ιδιαίτερα χρήσιμο για την προετοιμασία των δεδομένων καθώς βοηθάει ιδιαίτερα στον χειρισμό ελλιπών τιμών, στην επιλογή συγκεκριμένων στηλών, στην αντικατάστασης καθώς και στην ομαδοποίηση των δεδομένων.

3.1.4 Scikit-learn

Το Scikit-learn είναι και αυτό μια πάρα πολύ γνωστή βιβλιοθήκη της Python που χρησιμοποιείται στο τομέα της μηχανικής μάθησης και της επιστήμης δεδομένων. Παρέχει μια ολοκληρωμένη συλλογή από αλγορίθμους τόσο επιβλεπόμενης όσο και μη επιβλεπόμενης μάθησης.

3.1.5 Matplotlib

Το Matplotlib είναι μια βιβλιοθήκη της Python που χρησιμοποιείται για τη δημιουργία στατικών, κινουμένων και διαδραστικών διαγραμμάτων .Προσφέρει πολλούς τύπους γραφικών παραστάσεων και αυτό το καθιστά ένα πάρα πολύ ισχυρό εργαλείο για την οπτικοποίηση των δεδομένων.

3.1.6 NumPy

Το NumPy είναι μια βασική βιβλιοθήκη της Python που χρησιμοποιείται ευρέως για την εκτέλεση αριθμητικών υπολογισμών και ιδιαίτερα σε πράξεις μεταξύ πολυδιάστατων

πινάκων. Συνοδεύεται από μια μεγάλη συλλογή μαθηματικών συναρτήσεων. Είναι γραμμένη C και C++ και αυτό την καθιστά πολύ πιο γρήγορη σε υπολογισμούς προηγμένων μαθηματικών και άλλου τύπου πράξεων σε μεγάλα σύνολα δεδομένων από ένα τυπικό κώδικα σε Python.

3.1.7 Keras

Το Keras είναι μια βιβλιοθήκη ανοιχτού κώδικα (open source) και χρησιμοποιείται στο πεδίο της βαθιάς μάθησης. Αναπτύχθηκε αρχικά ως μέρος της ερευνητικής προσπάθειας του έργου ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Έχει ξεχωρίσει για την ευκολία στη χρήση του και αποτελεί ένα προσιτό τρόπο για την εισαγωγή στην προηγμένη μηχανική μάθηση και τα νευρωνικά δίκτυα. Το API που παρέχει το keras είναι αρκετά απλό για τον χρήστη καθώς τα μοντέλα Keras συναρμολογούνται σαν δομικά στοιχεία όπου τα διάφορα επίπεδα ενός νευρωνικού δικτύου όπως είναι τα νευρωνικά επίπεδα, οι συναρτήσεις κόστους, οι βελτιστοποιητές (optimizers), οι συναρτήσεις ενεργοποίησης κ.α. είναι όλα αυτόνομα και μπορούν να συνδυαστούν με τον τρόπο που θα επιλέξει ο χρήστης. Το keras είναι τόσο ισχυρό που χρησιμοποιείται από οργανισμούς όπως είναι η Nasa. [25]

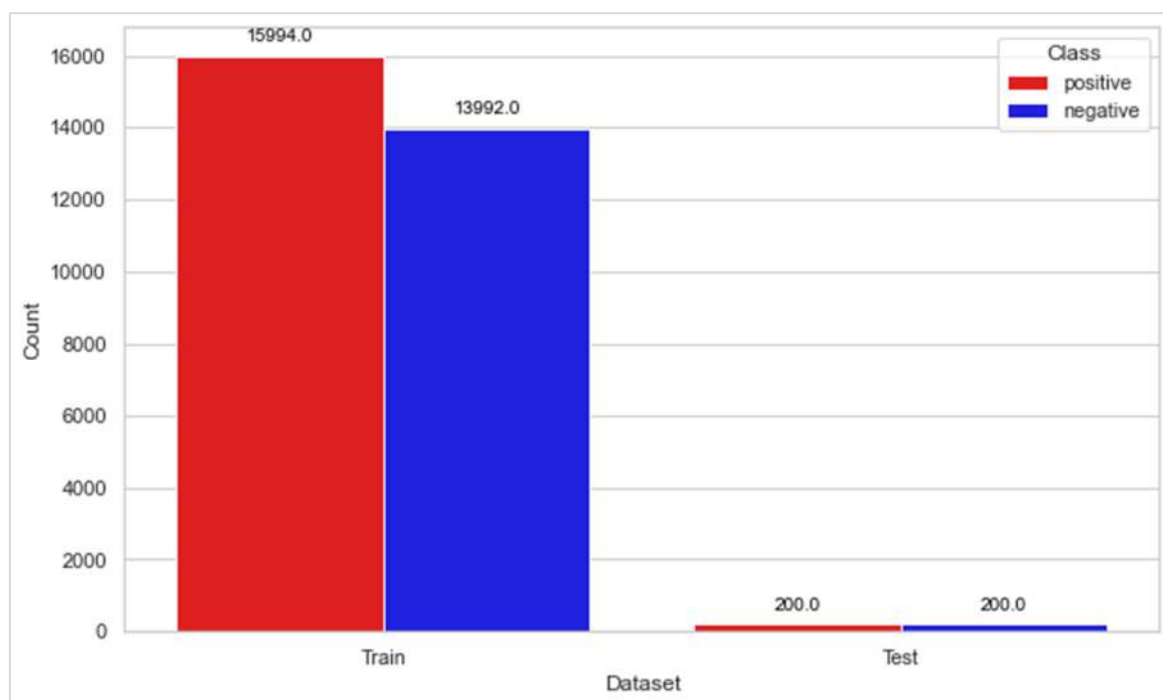
3.1.8 Google Colab Pro

Επειδή η εκτέλεση των μοντέλων απαιτούσε σημαντικούς υπολογιστικούς πόρους δηλαδή μια ισχυρή CPU και ιδανικά μια GPU για ταχύτερη επεξεργασία το Google Colab Pro αποτέλεσε το ιδανικό περιβάλλον για την υλοποίηση της παρούσας διπλωματικής. Τόσο γιατί παρέχει ισχυρούς υπολογιστικούς πόρους όπως GPU και TPU (Tensor Processing Units) οι οποίοι μπορούν να συμβάλουν στην ταχύτερη εκπαίδευση των μοντέλων και στην εξαγωγή συμπερασμάτων και ιδιαίτερα σε πολύπλοκα μοντέλα που χρειάζονται εκτεταμένη υπολογιστική ισχύ.

3.2 Σύνολο δεδομένων

Το σύνολο δεδομένων covidx-cxr2 που αντλήθηκε από το Kaggle ενημερώθηκε έως τις 06/02/2022 με δεδομένα, αποτελεί πολύτιμο πόρο για το έργο της ανίχνευσης

Covid-19 από ακτινογραφίες θώρακα. Αυτό που καθιστά αυτό το σύνολο δεδομένων ιδιαίτερα αξιοσημείωτο είναι η προσεκτική επιμέλειά του για την επίτευξη μιας ισορροπημένης εκπροσώπησης τόσο των θετικών όσο και των αρνητικών περιπτώσεων Covid-19. Η ισορροπία αυτή εξασφαλίζει ότι τα μοντέλα που εκπαιδεύονται σε αυτό το σύνολο δεδομένων δεν μεροληπτούν προς καμία από τις δύο κατηγορίες, επιτρέποντας ακριβέστερες και αμερόληπτες αξιολογήσεις. Το σύνολο δεδομένων όπως θα δούμε και στη συνέχεια αποτελείται από ένα σύνολο δεδομένων για την εκπαίδευση (training) και ένα σύνολο δεδομένων ελέγχου (testing).

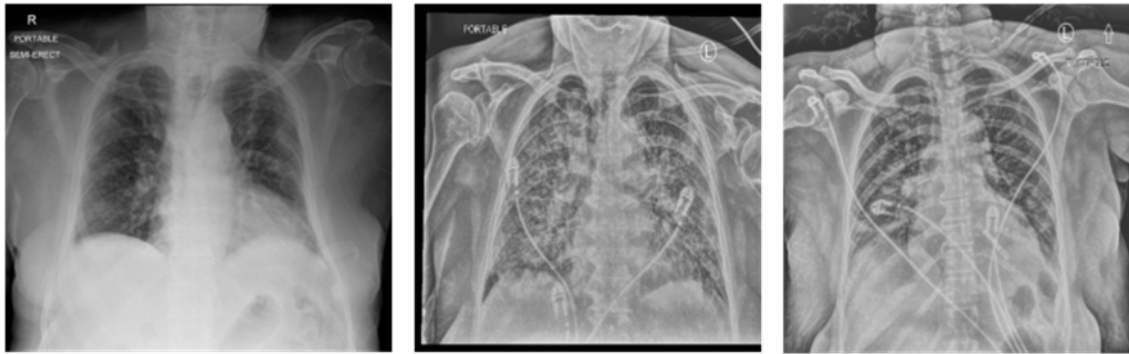


Εικόνα 9 : Κατανομή των κλάσεων στα δεδομένα εκπαίδευσης και δοκιμής

Όπως φαίνεται και από το παραπάνω διάγραμμα τα δεδομένα αποτελούνται:

- Από το σύνολο δεδομένων εκπαίδευσης (train dataset), στο οποίο υπάρχουν 13992 ακτινογραφίες ανθρώπων που δεν πάσχουν από Covid-19 και 15994 ακτινογραφίες ανθρώπων που έχουν νοσήσει από Covid-19.
- Αντίστοιχα βλέπουμε και το σύνολο δεδομένων ελέγχου (test dataset) το οποίο αποτελείται από 200 ακτινογραφίες υγιών ανθρώπων και 200 ακτινογραφίες ανθρώπων που έχουν νοσήσει.

Οι παρακάτω εικόνες αποτελούν ένα δείγμα ακτινογραφιών που ανήκουν στην κάθε κλάση



Εικόνα 10 : Ακτινογραφίες ασθενών που βρέθηκαν θετικοί στο Covid-19

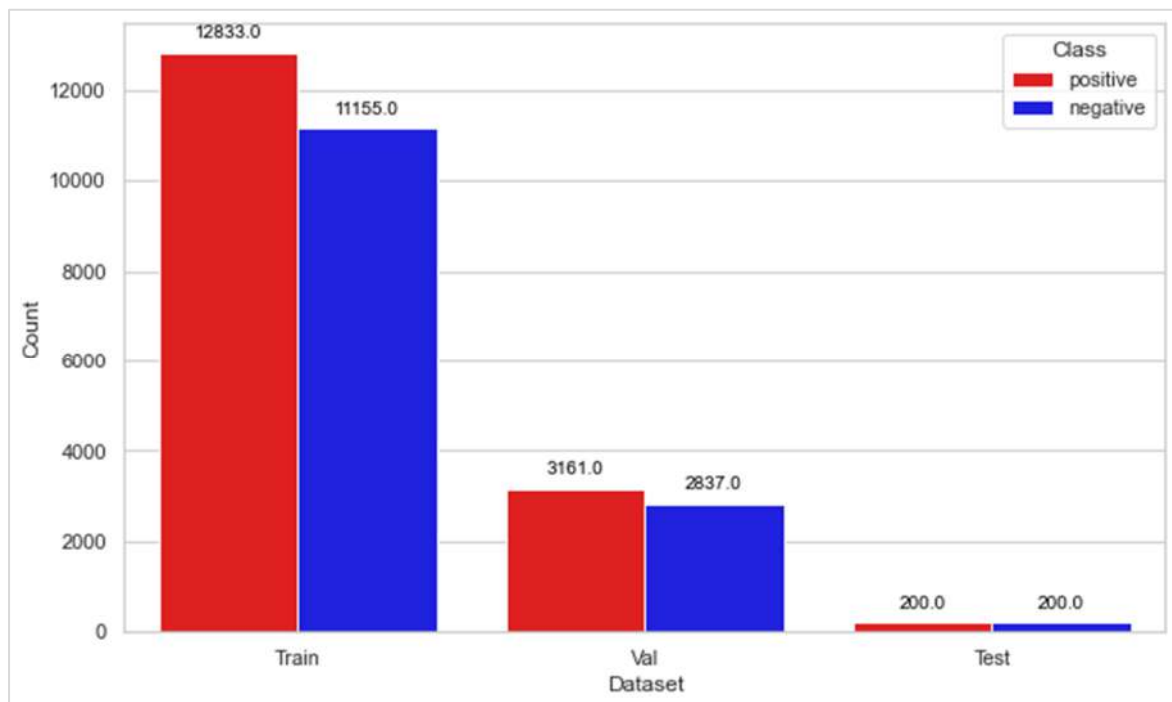


Εικόνα 11 : Ακτινογραφίες ασθενών που βρέθηκαν αρνητικοί στο Covid-19

Επειδή, το σύνολο δεδομένων των εικόνων αποτελούνταν από διαφορετικά μεγέθη, έγινε μετασχηματισμός των διαστάσεων των εικόνων σε 224 x 224. Το μέγεθος αυτό ήταν ιδανικό διότι δεν θέλουμε να χάσουμε μεγάλο μέρος από την πληροφορία της εικόνας και επίσης για το μοντέλο ResNet-50 είναι ένα μέγεθος που προτείνεται σαν καλή πρακτική.

Το σύνολο δεδομένων αφού μετασχηματίστηκε στις διαστάσεις που αναφέραμε έγινε διάσπαση των δεδομένων που ανήκουν στα δεδομένα εκπαίδευσης σε δεδομένα εκπαίδευσης και σε δεδομένα επικύρωσης με τη χρήση της `train_test_split` από τη βιβλιοθήκη `sklearn` της `Python`, ώστε να μπορούμε να αξιολογούμε το κάθε μοντέλο κατά τη διάρκεια της εκπαίδευσης και να αποτρέψουμε την υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης καθώς και για την καλύτερη αξιολόγηση και συντονισμό των υπερπαραμέτρων. Χρησιμοποιήθηκε η ρύθμιση του `random_state` με έναν σταθερό αριθμό το 42 ώστε να εξασφαλίσουμε ότι ο διαχωρισμός είναι

αναπαραγωγικός επιτρέποντας σε οποιονδήποτε τρέχει το κώδικα να μπορεί να λάβει τα ίδια αποτελέσματα. Ο αριθμός δεν έχει καμία ιδιαίτερη ιδιότητα απλά είναι ένας ευρέως χρησιμοποιούμενος αυθαίρετος αριθμός, θα μπορούσε στη θέση του να είναι οποιοσδήποτε ακέραιος αριθμός που θα μας έδινε διαφορετικές εκδοχές διαχωρισμού των δεδομένων αλλά και πάλι η αναπαραγωγή του κώδικα που είναι και το ζητούμενο θα υφίστανται.



Εικόνα 12 : Κατανομή των κλάσεων στα δεδομένα εκπαίδευσης , δοκιμής και επικύρωσης

Όπως βλέπουμε και από το διάγραμμα το σύνολο των δεδομένων επικύρωσης αποτελείται από το 20% του αρχικού συνόλου δεδομένων του training.

Στη συνέχεια εφαρμόστηκαν τεχνικές κανονικοποίησης των εικονοστοιχείων (pixels) με τη βοήθεια του ImageDataGenerator της βιβλιοθήκης Keras .Η κανονικοποίηση των εικονοστοιχείων (pixels) σε ένα σταθερό εύρος όπως το $[0,1]$ είναι μια κοινή πρακτική που βοηθάει τα νευρωνικά δίκτυα να οδηγούνται σε μια πιο σταθερή και αποτελεσματική μάθηση καθώς εξασφαλίζει ότι το μοντέλο θα είναι λιγότερο ευαίσθητο σε αλλαγές στην κλίμακα εισόδου και το καθιστά πιο ανθεκτικό στις διάφορες διακυμάνσεις που μπορεί να έχουν τα δεδομένα. Επιπλέον, συμβάλλει στη διατήρηση της αριθμητικής σταθερότητας και βοηθάει στην αποφυγή προβλημάτων όπως είναι το

exploding ή το vanishing gradients που μπορεί να προκύψουν ιδιαίτερα σε περιπτώσεις που εργαζόμαστε με έναν μεγάλο όγκο δεδομένων. Η κανονικοποίηση εφαρμόστηκε στο σύνολο δεδομένων της εκπαίδευσης, της επικύρωσης αλλά και στο σύνολο δεδομένων ελέγχου.

Στο σύνολο δεδομένων της εκπαίδευσης εφαρμόστηκαν και άλλοι μετασχηματισμοί όπως η επαύξηση δεδομένων δηλαδή τη δημιουργία τροποποιημένων εκδόσεων των εικόνων του συνόλου, παρέχοντας στο μοντέλο περισσότερα παραδείγματα για να μάθει. Πιο συγκεκριμένα, εφαρμόστηκε η τεχνική της τυχαίας μεγέθυνσης (zoom_range=0.1) με σκοπό να βοηθήσει το μοντέλο να καταφέρει να αναγνωρίσει αντικείμενα στην εικόνα ανεξάρτητα το μέγεθος τους. Επίσης, εφαρμόστηκαν οι τεχνικές της τυχαίας μετατόπισης και στον οριζόντιο (width_shift_range=0.1) αλλά και στο κάθετο άξονα (height_shift_range=0.1), της τυχαίας περιστροφής (rotation_range=10), της τυχαίας οριζόντιας αναστροφής (horizontal_flip=True) και τυχαίας μετατροπής (shear_range=0.1)

Οι τιμές που δοθήκαν σε αυτές τις παραμέτρους είναι πολύ μικρές γιατί οι ακτινογραφίες περιέχουν χρήσιμες πληροφορίες και έπρεπε να διασφαλιστεί ότι δεν θα παραμορφωθούν.

3.3 Αρχιτεκτονικός Σχεδιασμός Μοντέλων

Κατά τη διαδικασία της σχεδίασης της αρχιτεκτονικής των μοντέλων που εφαρμόστηκαν έπρεπε να ληφθούν υπόψιν κάποιοι παράγοντες όπως είναι ο όγκος και η πολυπλοκότητα των δεδομένων, οι ελάχιστοι υπολογιστικοί πόροι που είχαμε στη διάθεση μας, καθώς φυσικά όσα αναφέρθηκαν να μην επηρεάσουν σε μεγάλο βαθμό την απόδοση και την αξιοπιστία των μοντέλων.

Κρίσιμο ρόλο έπαιξαν οι υπερπαραμέτροι, μερικές από τις οποίες διατηρήθηκαν σταθερές, όπως θα δούμε και παρακάτω πιο αναλυτικά ενώ άλλες προσαρμόστηκαν για την ιδανική προσαρμογή και απόδοση των μοντέλων.

Για την ανάπτυξη κάθε νευρωνικού δικτύου και την αξιολόγηση τους ακολουθήθηκαν τα παρακάτω βήματα :

Define the Network :Καθορισμός των επιπέδων του νευρωνικού δικτύου και των συνδέσεων τους. Όπως είναι ο τύπος των επιπέδων, ο αριθμός των νευρώνων σε κάθε επίπεδο και οι συναρτήσεις ενεργοποίησης.

Compile the Network: Καθορίστηκε η συνάρτηση απώλειας, ο βελτιστοποιητής (optimizer) που καθαρίζει τον τρόπο με τον οποίο το δίκτυο θα ενημερώνει τα βάρη του κατά τη διάρκεια της εκπαίδευσης. Καθώς και οι μετρικές που θα αξιολογήσουν το δίκτυο.

Fit the Network: Σε αυτό βήμα το δίκτυο ξεκινάει να εκπαιδεύεται και να μαθαίνει από τα δεδομένα.

Evaluate the Network: Αφού ολοκληρωθεί η εκπαίδευση το δίκτυο αξιολογείται πάνω σε δεδομένα που του είναι άγνωστα δηλαδή στο σύνολο δεδομένων δοκιμής (test dataset).

Predictions: Έγιναν οι πραγματικές προβλέψεις πάνω στο σύνολο δεδομένων δοκιμής για να ελεγχθεί ο αριθμός των δεδομένων που κατάφερε να ταξινομήσει σωστά.

Δυο από τις πιο δύσκολες αποφάσεις που έπρεπε να παρθούν ήταν η επιλογή της αρχιτεκτονικής των μοντέλων καθώς υπάρχουν πολλές αρχιτεκτονικές μοντέλων που θα ήταν κατάλληλες για τον σκοπό αυτής της διπλωματικής. Ωστόσο, λήφθηκε η απόφαση να χρησιμοποιηθούν αρχιτεκτονικές συνελικτικών νευρωνικών δικτύων όπως είναι το ResNet-50 το οποίο έχει αποδειχθεί ότι είναι αποτελεσματικό σε εργασίες ταξινόμησης εικόνας όπως είναι οι ακτινογραφίες όπου υπάρχει η ανάγκη να καταγράφουν λεπτομέρειες και πολύπλοκα μοτίβα. Καθώς και ένα συνελικτικό νευρωνικό δίκτυο το οποίο θα υλοποιηθεί από την αρχή.

Επιπλέον, μια ακόμη δύσκολη απόφαση ήταν ο ορισμός των δικτύων για το οποίο δεν υπάρχει κατά βάση κάποιος κανόνας και οι τελικές δομές τους προέκυψαν μετά από πειραματισμούς και ελέγχους.

Στόχος είναι να συγκρίνουμε τα αποτελέσματα από αυτές τις δυο αρχιτεκτονικές σε σχέση και με τους πειραματισμούς που κάναμε για τις υπερπαραμέτρους και για τον αριθμό και τον τύπο των επιπέδων που χρησιμοποιήθηκαν. Από τη μια μεριά θέλουμε να μελετήσουμε τα αποτελέσματα που μπορούμε να πάρουμε χρησιμοποιώντας ένα προ εκπαιδευμένο μοντέλο όπως είναι το ResNet-50 που δίνει τη δυνατότητα να επαναχρησιμοποιήσουμε τη γνώση που έχει λάβει και τα χαρακτηριστικά που έχει μάθει πάνω στο δικό μας πρόβλημα και από την άλλη θέλουμε να δούμε πως ένα

μοντέλο που εκπαιδεύεται από την αρχή πάνω στο σύνολο δεδομένων που θα χρησιμοποιήσουμε μπορεί να δώσει έστω και κάποια ικανοποιητικά αποτελέσματα και να μας προβληματίσει για περαιτέρω έρευνα.

3.4 Συνελικτικό Νευρωνικό δίκτυο από την αρχή (CNN from scratch)

Η αρχιτεκτονική του μοντέλου κατά τη διάρκεια των πειραμάτων άλλαξε αρκετές φορές μορφή αλλά η αρχιτεκτονική που κατάφερε να δώσει τα πιο ικανοποιητικά αποτελέσματα είναι αυτή που ακολουθεί στην εικόνα 13.

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 222, 222, 32)	896
batch_normalization_4 (Batch Normalization)	(None, 222, 222, 32)	128
max_pooling2d_3 (MaxPooling2D)	(None, 111, 111, 32)	0
dropout_4 (Dropout)	(None, 111, 111, 32)	0
conv2d_4 (Conv2D)	(None, 109, 109, 64)	18496
batch_normalization_5 (Batch Normalization)	(None, 109, 109, 64)	256
max_pooling2d_4 (MaxPooling2D)	(None, 54, 54, 64)	0
dropout_5 (Dropout)	(None, 54, 54, 64)	0
conv2d_5 (Conv2D)	(None, 52, 52, 128)	73856
batch_normalization_6 (Batch Normalization)	(None, 52, 52, 128)	512
max_pooling2d_5 (MaxPooling2D)	(None, 26, 26, 128)	0
dropout_6 (Dropout)	(None, 26, 26, 128)	0
flatten_1 (Flatten)	(None, 86528)	0
dense_2 (Dense)	(None, 256)	22151424
batch_normalization_7 (Batch Normalization)	(None, 256)	1024
dropout_7 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257

Εικόνα 13 : CNN Model from Scratch

Πιο συγκεκριμένα η δομή του διαμορφώθηκε από τρία σύνολα επιπέδων. Το κάθε σύνολό περιέχει ένα συνελκτικό επίπεδο, ένα επίπεδο κανονικοποίησης παρτίδας (Batch Normalization Layer), ένα επίπεδο ομαδοποίησης μεγίστου (MaxPooling Layer) και ένα επίπεδο αποκοπής νευρώνων(Dropout Layer). Τα συνελκτικά επίπεδα αυξάνουν τον αριθμό των φίλτρων τους ανά επίπεδο ώστε να μπορούν να εντοπίσουν τα πιο περίπλοκα μοτίβα στα πιο βαθιά επίπεδα. Επίσης, κατά τη διάρκεια της εκπαίδευσης στα πρώτα μοντέλα που έγιναν τα πρώτα πειράματα παρατηρήθηκε το φαινόμενο της υπερπροσαρμογής και για αυτό το λόγο προστέθηκαν τα Dropout επίπεδα με πιθανότητα απόρριψης 0.25 μετά από κάθε επίπεδο MaxPooling. Τα επίπεδα MaxPooling μείωναν τις διαστάσεις των feature maps αυτό είχε σαν αποτέλεσμα τα επόμενα συνελκτικά επίπεδα να εστιάζουν σε συγκεκριμένα μοτίβα.

Μετά από αυτό το σύνολο επιπέδων που περιγράψαμε ακολουθεί ένα επίπεδο εξομάλυνσης(Flatten layer) το οποίο μετατρέπει τους τρισδιάστατους χάρτες χαρακτηριστικών (feature maps) σε μονοδιάστατους ώστε να μπορέσει να γίνει η σύνδεση μεταξύ των συνελκτικών επιπέδων με το πυκνό επίπεδο (dense layer).

Το πυκνό επίπεδο (dense layer) το οποίο αποτελείται από 256 νευρώνες όπου πραγματοποιείται η ταξινόμηση βασισμένη στα χαρακτηριστικά που έχουν εξαχθεί. Δυο ακόμη επίπεδα κανονικοποίησης παρτίδας (BatchNormalization) και αποκοπής νευρώνων(Dropout) με πιθανότητα απόρριψης 0.5 ώστε να μειώσουμε περαιτέρω τον κίνδυνο υπερπροσαρμογής.

Και τέλος ένα επίπεδο εξόδου δηλαδή ένα πυκνό (Dence) επίπεδο με ένα μόνο νευρώνα καθώς η δυαδική ταξινόμηση απαιτεί μια μόνο έξοδο για την αναπαράσταση των δύο κλάσεων .Η συνάρτηση ενεργοποίησης επιλέχθηκε είναι η Sigmoid γιατί είναι κατάλληλη για δυαδική ταξινόμηση δεδομένου ότι η έξοδός της είναι μια πιθανότητα που κυμαίνεται από 0 έως 1 και αντιπροσωπεύει πόσο πιθανό είναι η είσοδος να ανήκει στη θετική κλάση.

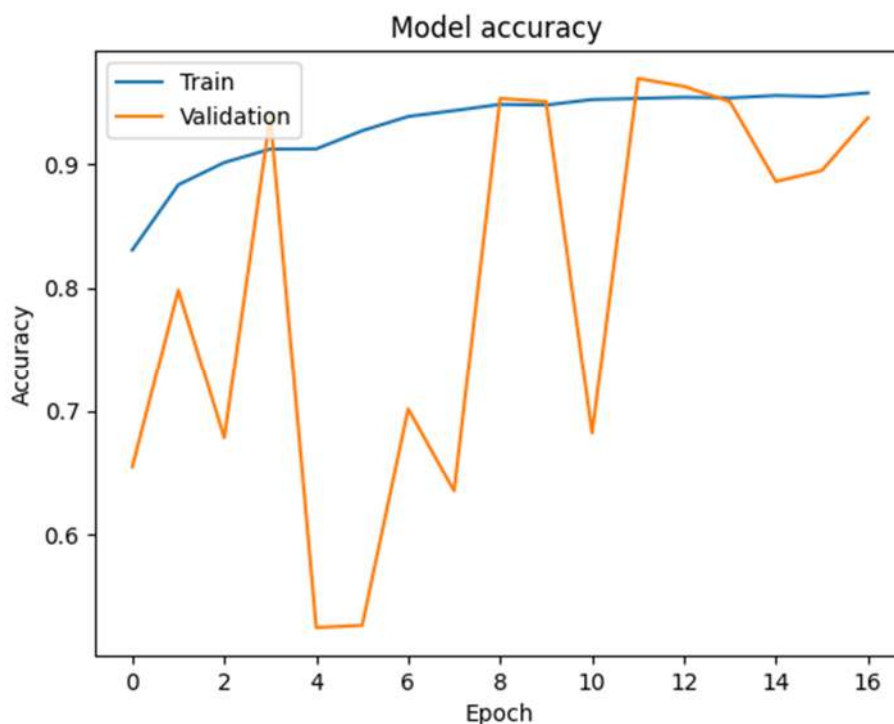
Ως βελτιστοποιητής (optimizer) επιλέχθηκε ο Adam με ρυθμό εκμάθησης 0.001 και σαν συνάρτηση κόστους έχει οριστεί η Binary-Crossentropy η οποία όπως έχουμε ήδη αναφέρει χρησιμοποιείται σε εργασίες όπου τα δεδομένα ανήκουν σε δυο κλάσεις . Κατά τη διάρκεια της εκπαίδευσης χρησιμοποιήθηκαν διάφορα callbacks και πιο συγκεκριμένα το EarlyStopping το οποίο διακόπτει την εκπαίδευση όταν δεν παρατηρείτε κάποια βελτίωση στην τιμή της συνάρτησης κόστους για τα δεδομένα

επικύρωσης για πέντε συνεχόμενες εποχές (epochs) και με την παράμετρο `restore_best_weights` επαναφέρει τις παραμέτρους του μοντέλου στις τιμές που είχαν όταν είχαν την καλύτερη απόδοση στην τιμή της συνάρτησης κόστους για τα δεδομένα επικύρωσης. Το `ModelCheckpoint` το οποίο αποθηκεύει το καλύτερο μοντέλο που θα προκύψει σε αρχείο της μορφής `h5`. Καθώς και το `ReduceLROnPlateau` το οποίο μειώνει τον ρυθμό μάθησης όταν δεν παρατηρείται κάποια βελτίωση στην τιμή της συνάρτησης κόστους για τα δεδομένα επικύρωσης για δυο συνεχόμενες εποχές (epochs). Η τιμή του αριθμού των εποχών (epochs) ορίστηκε στις 20 εποχές ωστόσο κατά τη διάρκεια της εκπαίδευσης λόγω του `EarlyStopping` που είχαμε ορίσει η εκπαίδευση ολοκληρώθηκε στη δέκατη έκτη εποχή. Επίσης, σαν batch size επιλέχθηκε το 36. Πραγματοποιήθηκαν πειραματισμοί κυρίως με μεγαλύτερες τιμές στο batch size λόγω υπολογιστικών πόρων αλλά τα αποτελέσματα δεν ήταν τα αναμενόμενα.

3.4.1 Αξιολόγηση Convolutional Neural Network from Scratch

Για την αξιολόγηση του νευρωνικού δικτύου θα χρησιμοποιήσουμε διάφορες μετρικές που έχουν αναφερθεί στην παρούσα εργασία όπως είναι το accuracy, το precision, το recall και το f1-score τόσο κατά τη διάρκεια της εκπαίδευσης όσο και κατά τη διάρκεια της πρόβλεψης σε νέα άγνωστα για αυτό δεδομένα.

Το πρώτο βήμα είναι να αξιολογήσουμε την ακρίβεια (Accuracy) του μοντέλου στις διαδοχικές εποχές (epochs) κατά τη διάρκεια της εκπαίδευσης και να κατανοήσουμε καλύτερα πως εξελίχθηκε η μάθηση του μοντέλου.



Εικόνα 14 : Training VS Validation Accuracy (CNN from scratch)

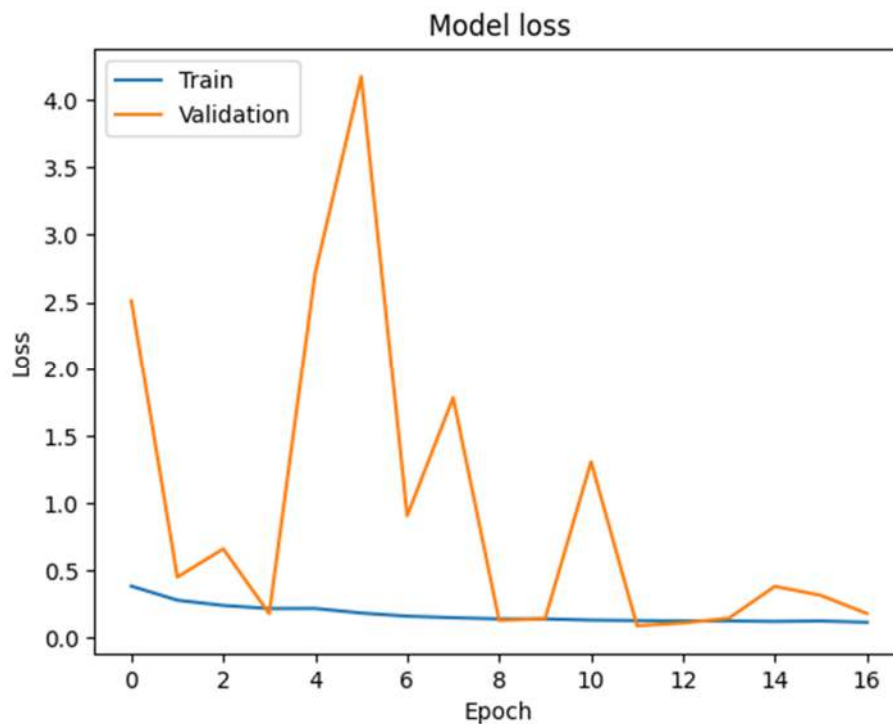
Από την εικόνα 14 παρατηρούμε ότι η ακρίβεια (Accuracy) εκπαίδευσης δείχνει μια γενική τάση βελτίωσης από εποχή σε εποχή . Αρχικά, η ακρίβεια του μοντέλου είναι περίπου 0.8 και σταδιακά βελτιώνεται φτάνοντας πάνω από 0.9 υποδεικνύοντας υψηλό επίπεδο ακρίβειας στο σύνολο εκπαίδευσης. Αυτό υποδηλώνει ότι το μοντέλο είναι ικανό να μαθαίνει χαρακτηριστικά και μοτίβα από τις εικόνες. Η σχετικά ομαλή και συνεπής ανοδική πορεία υποδηλώνει ότι το μοντέλο μαθαίνει σταθερά από τα δεδομένα εκπαίδευσης χωρίς σημαντικές διακυμάνσεις, γεγονός που θα μπορούσε να είναι ενδεικτικό ενός σταθερού ρυθμού μάθησης και επαρκούς πολυπλοκότητας του μοντέλου.

Από την άλλη η ακρίβεια (Accuracy) επικύρωσης παρουσιάζει σημαντικές διακυμάνσεις από εποχή σε εποχή με μια μεγάλη μείωση κατά την τέταρτη εποχή . Η μείωση αυτή της ακρίβειας θα μπορούσε να είναι ενδεικτική της αδυναμίας του μοντέλου να γενικεύσει όταν έρχεται αντιμέτωπο με νέα δεδομένα. Ωστόσο, η αύξηση της ακρίβειας μετά τη δωδέκατη εποχή υποδηλώνει ότι το μοντέλο μαθαίνει να γενικεύει καλύτερα τις προβλέψεις του στο σύνολο επικύρωσης με περαιτέρω εκπαίδευση.

Η απόκλιση μεταξύ της ακρίβειας εκπαίδευσης και της ακρίβειας επικύρωσης αποτελεί σημείο ανησυχίας. Μια σταθερά υψηλή ακρίβεια εκπαίδευσης σε συνδυασμό με μια

ασταθή ακρίβεια επικύρωσης υποδεικνύει ένα μοντέλο που μπορεί να μη γενικεύει καλά σε νέα δεδομένα δηλαδή να υπάρχει το πρόβλημα υπερπροσαρμογής.

Στη συνέχεια θα εξετάσουμε την πρόοδο της απώλειας (loss) και την απώλεια της επαλήθευσης (validation loss) κατά τη διάρκεια της εκπαίδευσης.



Εικόνα 15 : Απώλεια Εκπαίδευσης και Επικύρωσης (CNN from scratch)

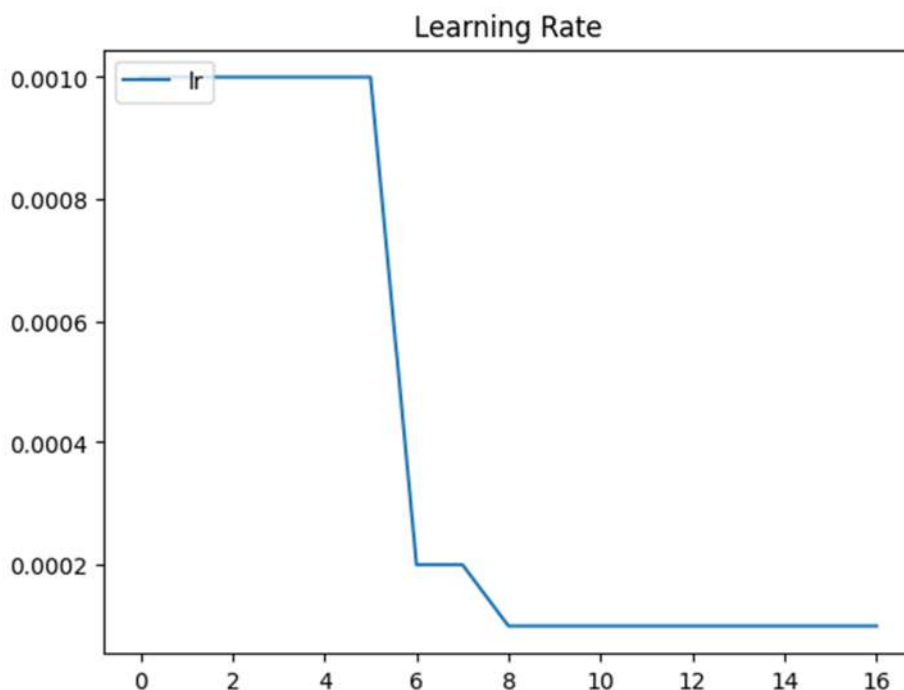
Από την εικόνα 15 βλέπουμε ότι η απώλεια εκπαίδευσης ξεκινά με μια χαμηλή τιμή και παραμένει σχετικά σταθερή κατά τη διάρκεια την εκπαίδευσης. Αυτή η σταθερότητα της απώλειας εκπαίδευσης αποτελεί θετικό σημάδι, υποδεικνύοντας ότι το μοντέλο μαθαίνει σταθερά και βελτιώνει τις προβλέψεις του στο σύνολο εκπαίδευσης κατά τη διάρκεια της διαδικασίας εκπαίδευσης.

Αντίθετα, η απώλεια επικύρωσης εμφανίζει σημαντικές διακυμάνσεις και απότομες μεταβολές. Ξεκινά με μια πολύ υψηλή τιμή υποδεικνύοντας κακή αρχική απόδοση στο σύνολο επικύρωσης υποδεικνύοντας ότι το μοντέλο μπορεί να μην έχει γενικεύσει αμέσως καλά από τα δεδομένα εκπαίδευσης στα δεδομένα επικύρωσης. Στη συνέχεια εμφανίζει ιδιαίτερα έντονες μεταβολές κατά τις εποχές 3, 7 και 11 γεγονός που μπορεί να σημαίνει ότι το μοντέλο συναντά ορισμένα παραδείγματα επικύρωσης τα οποία

διαφέρουν σημαντικά από τα δεδομένα εκπαίδευσης. Ωστόσο, η μείωση των απωλειών μετά την έκτη εποχή δείχνει ότι το μοντέλο άρχισε να βελτιώνει τη γενίκευση του στο σύνολο επικύρωσης καθώς προχωρούσε η εκπαίδευση.

Η απόκλιση μεταξύ της απώλειας εκπαίδευσης και επικύρωσης υποδεικνύει ότι το μοντέλο μπορεί να προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης, κάτι που δεν είναι ασυνήθιστο στα μοντέλα βαθιάς μάθησης. Ωστόσο, δεδομένου ότι η απώλεια επικύρωσης μειώνεται μετά από κάποιες εποχές, υποδηλώνει ότι το μοντέλο είναι ικανό να γενικεύει καλύτερα σε νέα δεδομένα μετά από περαιτέρω εκπαίδευση.

Κατά τη διάρκεια της εκπαίδευσης είναι σημαντικό να εξετάσουμε το πως διαμορφώθηκε ο ρυθμός μάθησης καθώς αυτός είναι υπεύθυνος για το μέγεθος των βημάτων που κάνει ο αλγόριθμος βελτιστοποίησης προς το ελάχιστο της συνάρτησης απώλειας. Από την εικόνα 16 που ακολουθεί βλέπουμε τις τιμές που πήρε ο ρυθμός μάθησης κατά τη διάρκεια της εκπαίδευσης.



Εικόνα 16 : Ρυθμός Μάθησης ανά Εποχή (CNN from scratch)

Η γραφική απεικόνιση του ρυθμού μάθησης μας φανερώνει ότι ο ρυθμός μάθησης στην αρχή μέχρι την πέμπτη εποχή είναι σταθερός και μετά υπάρχει μια αιφνίδια μείωση. Αυτό συμβαίνει διότι ενεργοποιήθηκε το callback ReduceLROnPlateau λόγω

της μη βελτίωσης της τιμής της συνάρτησης κόστους για τα δεδομένα επικύρωσης. Στη συνέχεια μετά από την όγδοη εποχή παραμένει σταθερός επειδή έχει φτάσει στην ελάχιστη τιμή που έχουμε ορίσει $\text{min_lr}=0.0001$.

Από την περίοδο της εκπαίδευσης μπορούμε να συμπεράνουμε ότι η ακρίβεια εκπαίδευσης στις αρχικές εποχές αυξάνεται ενώ η απώλεια μειώνεται κάτι που είναι αναμενόμενο καθώς το μοντέλο αρχίζει να μαθαίνει από τα δεδομένα εκπαίδευσης. Ωστόσο, οι διακυμάνσεις στην ακρίβεια και στην απώλεια επικύρωσης υποδηλώνουν ότι οι προβλέψεις του μοντέλου δεν ευθυγραμμίζονται σταθερά με τα αναμενόμενα αποτελέσματα. Αυτή η διακύμανση είναι ένα πιθανό σημάδι της προσπάθειας του μοντέλου να γενικεύσει, η οποία θα μπορούσε να είναι αποτέλεσμα υπερβολικής προσαρμογής στα δεδομένα εκπαίδευσης, όπως αποδεικνύεται από την υψηλή ακρίβεια εκπαίδευσης σε σχέση με την ακρίβεια επικύρωσης. Οι προσαρμογές του ρυθμού μάθησης δηλαδή ξεκινώντας με έναν υψηλό ρυθμό μάθησης πιθανότατα βοήθησε στη γρήγορη μείωση των απωλειών καθώς το μοντέλο ήταν σε θέση να κάνει σημαντικές προσαρμογές στα βάρη του. Ωστόσο, καθώς ο ρυθμός μάθησης μειωνόταν θα έπρεπε να οδηγήσει σε μια πιο σταθερή αύξηση της ακρίβειας επικύρωσης και μια πιο σταθερή μείωση της απώλειας επικύρωσης, παρατηρούμε όμως ότι έχουμε σημαντικές διακυμάνσεις και στην ακρίβεια και στην απώλεια επικύρωσης κάτι που δεν είναι η αναμενόμενη συμπεριφορά που θα θέλαμε να έχει.

Στη συνέχεια θα αξιολογήσουμε την απόδοση του μοντέλου πρόβλεψης σε ένα νέο σύνολο δεδομένων στα δεδομένα ελέγχου (test dataset). Οι μετρικές των επιδόσεων διαμορφώθηκαν όπως φαίνεται στην εικόνα 17.

	precision	recall	f1-score	support
negative	0.90	0.98	0.94	200
positive	0.98	0.90	0.94	200
accuracy			0.94	400
macro avg	0.94	0.94	0.94	400
weighted avg	0.94	0.94	0.94	400

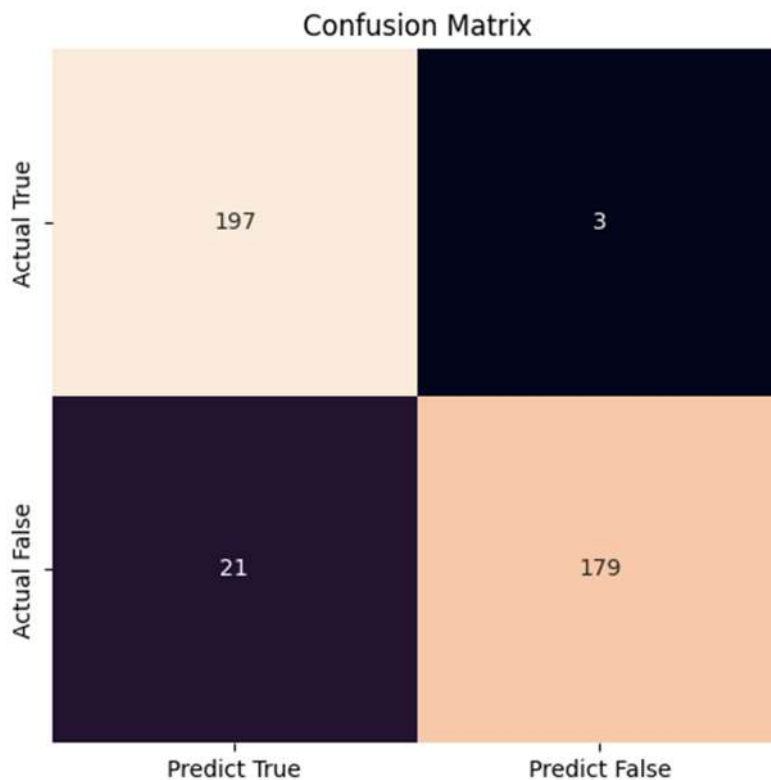
Εικόνα 17 : Μετρήσεις επιδόσεων στο σύνολο δεδομένων επικύρωσης (CNN from scratch)

Η συνολική ακρίβεια (Accuracy) του μοντέλου είναι 0.94 υποδεικνύοντας ότι προβλέπει σωστά την κατηγορία μιας εικόνας στο 94% των περιπτώσεων. Το μοντέλο επιτυγχάνει ακρίβεια (Precision) 0.90 για την αρνητική κλάση γεγονός που σημαίνει ότι όταν προβλέπει μια εικόνα ως αρνητική, είναι σωστό στο 90% των περιπτώσεων. Για τη θετική κλάση, η ακρίβεια (Precision) είναι υψηλότερη 0.98 που σημαίνει ότι το 98% των θετικών προβλέψεων που έκανε το μοντέλο ήταν σωστές. Η ανάκληση (Recall) 0.90 για τη θετική κλάση δείχνει ότι το μοντέλο είναι ιδιαίτερα ικανό να εντοπίσει τις περισσότερες θετικές περιπτώσεις. Η ανάκληση (Recall) 0.98 για την αρνητική κλάση που σημαίνει ότι το μοντέλο αναγνώρισε επιτυχώς το 98% όλων των πραγματικών αρνητικών περιπτώσεων.

Το f1-score (ο αρμονικός μέσος ορός της ακρίβειας (Precision) και της ανάκλησης) είναι 0.92 και για την αρνητική κλάση αλλά και για τη θετική κλάση που σημαίνει ότι υπάρχει μια καλή ισορροπία μεταξύ ακρίβειας και ανάκλησης και για τις δύο κλάσεις. Και σε συνδυασμό με τις μετρικές macro average και weighted average που είναι 0.94 και για την ακρίβεια (Precision) αλλά και για την ανάκληση ενισχύουν τη συνεπή απόδοση του μοντέλου σε όλες τις κλάσεις.

Από τις μετρικές των επιδόσεων πάνω στο σύνολο δεδομένων ελέγχου μπορούμε να συμπεράνουμε ότι το μοντέλο αποδίδει καλά στη διάκριση μεταξύ αρνητικών και θετικών περιπτώσεων Covid-19. Η υψηλή ακρίβεια και ανάκληση τόσο για τη θετική όσο και για την αρνητική κλάση είναι ιδιαίτερα αξιοσημείωτες.

Για να μπορέσουμε να κατανοήσουμε εις βάθος πως διαμορφωθήκαν αυτές οι μετρικές θα αναλύσουμε τον πίνακα σύγχυσης ή σφαλμάτων (Confusion Matrix) από την εικόνα 18 ο οποίος παρέχει μια λεπτομερή ανάλυση των προβλέψεων του μοντέλου σε σύγκριση με τις πραγματικές ετικέτες.



Εικόνα 18 : Πίνακας Σφαλμάτων (CNN from Scratch)

- *True Positive (TP)*: Το μοντέλο έχει υψηλό αριθμό πραγματικών θετικών αποτελεσμάτων (197), υποδεικνύοντα ότι έχει την ικανότητα να ανιχνεύσει το Covid-19 όταν είναι παρόν.
- *False Negative (FN)*: Το μοντέλο πρόβλεψε ψευδώς αρνητικά μόνο 3 από τις 200 περιπτώσεις.
- *False Positive (FP)*: Τα 21 ψευδώς θετικά αποτελέσματα αντιπροσωπεύουν τις περιπτώσεις στις οποίες το μοντέλο απέτυχε να ανιχνεύσει την απουσία Covid-19.
- *True Negative (TN)*: Ο υψηλός αριθμός πραγματικών αρνητικών (179) δείχνει ότι το μοντέλο έχει την ικανότητα επιβεβαίωσης της απουσίας της νόσου όταν αυτή δεν υπάρχει.

Από την ανάλυση του πίνακα σφαλμάτων μπορούμε να συμπεράνουμε ότι το μοντέλο έχει όπως αναφέραμε και προηγουμένως μια ισχυρή ικανότητα στο να ταξινομεί σωστά τόσο τις θετικές όσο και τις αρνητικές περιπτώσεις το οποίο είναι απαραίτητο στην κλινική διάγνωση. Ωστόσο, θα ήταν καλό να επισημάνουμε ότι σε ένα κλινικό πλαίσιο το μοντέλο ότι πρόβλεψε λανθασμένα 3 από τις 200 περιπτώσεις ως υγιείς θα ήταν

ιδιαίτερα ανησυχητικό καθώς η αποτυχία διάγνωσης ενός μολυσμένου ασθενούς μπορεί να οδηγήσει σε περαιτέρω εξάπλωση της νόσου και σε έλλειψη της απαραίτητης ιατρικής παρέμβασης για τον ασθενή. Καθώς επίσης και ότι πρόβλεψε λανθασμένα 21 περιπτώσεις ως θετικές θα μπορούσε να οδηγήσει σε άσκοπη ταλαιπωρία υγιείς ανθρώπους.

3.5 ResNet-50 Model

Η αρχιτεκτονική του μοντέλου ResNet-50 και η στρατηγική εκπαίδευσης επιλέχθηκαν προσεκτικά με βάση ότι τα προ εκπαιδευμένα δίκτυα όπως είναι το ResNet-50 μπορούν να προσαρμοστούν σε νέες απαιτήσεις ταξινόμησης όπως είναι η ανίχνευση Covid-19 από ακτινογραφίες. Όπως και ότι μπορεί να γίνει επανεκπαίδευση ορισμένων επιπέδων για την εκμάθηση χαρακτηριστικών που είναι ειδικά για το πρόβλημα που μελετάμε.

Model: "sequential"		
Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23587712
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 256)	524544
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

Εικόνα 19 : Αρχιτεκτονική ResNet-50

Όπως φαίνεται και από την εικόνα 19, η βάση του μοντέλου είναι το ResNet-50 και τα βάρη του αρχικοποιούνται με εκείνα από την προ εκπαίδευση του στο σύνολο δεδομένων ImageNet. Επίσης, εξαιρέθηκαν τα ανώτερα επίπεδα με την παράμετρο

(include_top=False) ώστε να επιτρέψουμε την εξαγωγή χαρακτηριστικών σε νέα δεδομένα.

Ακολουθεί ένα επίπεδο ομαδοποίησης (GlobalAveragePooling Layer) για τη μείωση του συνολικού αριθμού των παραμέτρων στο μοντέλο και την αποφυγή της υπερπροσαρμογής. Στη συνέχεια ακολουθεί ένα πυκνό πλήρως συνδεδεμένο επίπεδο (Dense Layer) με σκοπό να ερμηνεύσει τα χαρακτηριστικά που έχουν εξαχθεί από τα προηγούμενα επίπεδά και τη συμμετοχή του στη διαδικασία της ταξινόμησης . Μετά από αυτό ακολουθεί ένα επίπεδο αποκοπής νευρώνων (Dropout Layer) με πιθανότητα απόρριψης 0.5 ώστε να μειώσουμε περαιτέρω τον κίνδυνο υπερπροσαρμογής. Και τέλος είναι το επίπεδο εξόδου (Dense). Η συνάρτηση ενεργοποίησης που επιλέχθηκε και για αυτό το μοντέλο είναι η Sigmoid.

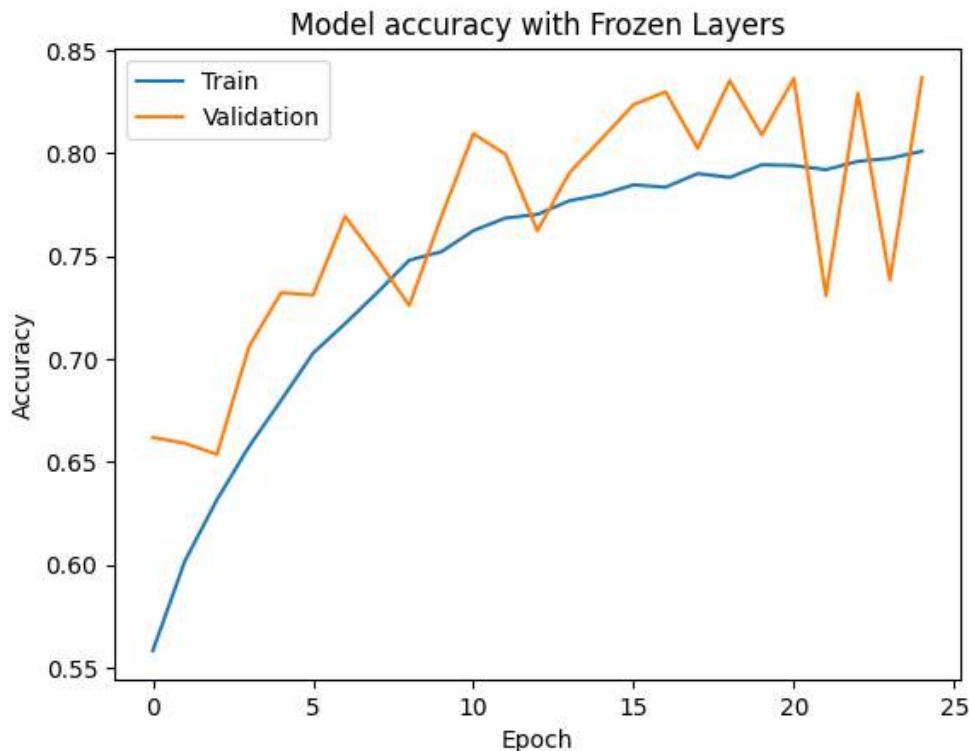
Κατά τη διάρκεια της εκπαίδευσης χρησιμοποιήθηκαν και σε αυτό το μοντέλο διαφορά Callbacks όπως το EarlyStopping και το ReduceLROnPlateau με τις ίδιες παραμέτρους που χρησιμοποιήσαμε και στο μοντέλο CNN from scratch με τη μόνη διαφορά ότι η μικρότερη τιμή που μπορεί να πάρει ο ρυθμός μάθησης είναι το $1e-6$. Ο λόγος που έγινε αυτό είναι διότι η εκπαίδευση θα μπορούσαμε να πούμε ότι χωρίστηκε σε δυο στάδια.

Στο πρώτο στάδιο το μοντέλο μας εκπαιδεύτηκε με τα επίπεδα της βάσης ResNet-50 παγωμένα θέτοντας την ιδιότητα trainable = False, δηλαδή τα επίπεδα του ResNet-50 διατηρούν τα προ-εκπαιδευμένα βάρη και δεν ενημερώνονται κατά τη διάρκεια της εκπαίδευσης και ενημερώνονται μόνο τα βάρη από τα τρία τελευταία επίπεδα . Και ως βελτιστοποιητής (optimizer) επιλέχθηκε ο Adam με ρυθμό εκμάθησης 0.0001. Το μοντέλο εκπαιδεύτηκε για ακριβώς 25 εποχές (epochs).

Στο δεύτερο στάδιο μετά την ολοκλήρωση της εκπαίδευσης έγινε «ξεπάγωμα» των επιπέδων του ResNet-50 με την ιδιότητα trainable = True και εφαρμόστηκε η αντίθετη λογική από το πρώτο στάδιο, δηλαδή επιτρέψαμε τη λεπτομερή ρύθμιση ολόκληρου του δικτύου ResNet-50 και την περαιτέρω εκπαίδευση του για 13 εποχές (epochs) και κρατήσαμε παγωμένα τα τελευταία επίπεδα το οποίο κατά συνέπεια σημαίνει ότι τα βάρη τους δεν ενημερώνονται. Χρησιμοποιήθηκε ένας μειωμένος ρυθμός μάθησης (0.00001) ώστε να πραγματοποιηθούν μικρές σταδιακές αλλαγές στα προ εκπαιδευμένα βάρη χωρίς να διαταράζονται τα ήδη μαθημένα χαρακτηριστικά.

3.5.1 Αξιολόγηση μοντέλου ResNet-50

Για την αξιολόγηση του μοντέλου κατά τη διάρκεια της εκπαίδευσης θα αναλύσουμε δυο σύνολα γραφημάτων ένα για το μοντέλο που εκπαιδεύτηκε στο πρώτο στάδιο και ένα στο δεύτερο στάδιο χρησιμοποιώντας ως μετρήσεις την ακρίβεια (Accuracy) και την απώλεια (Loss) εκπαίδευσης και επικύρωσης.

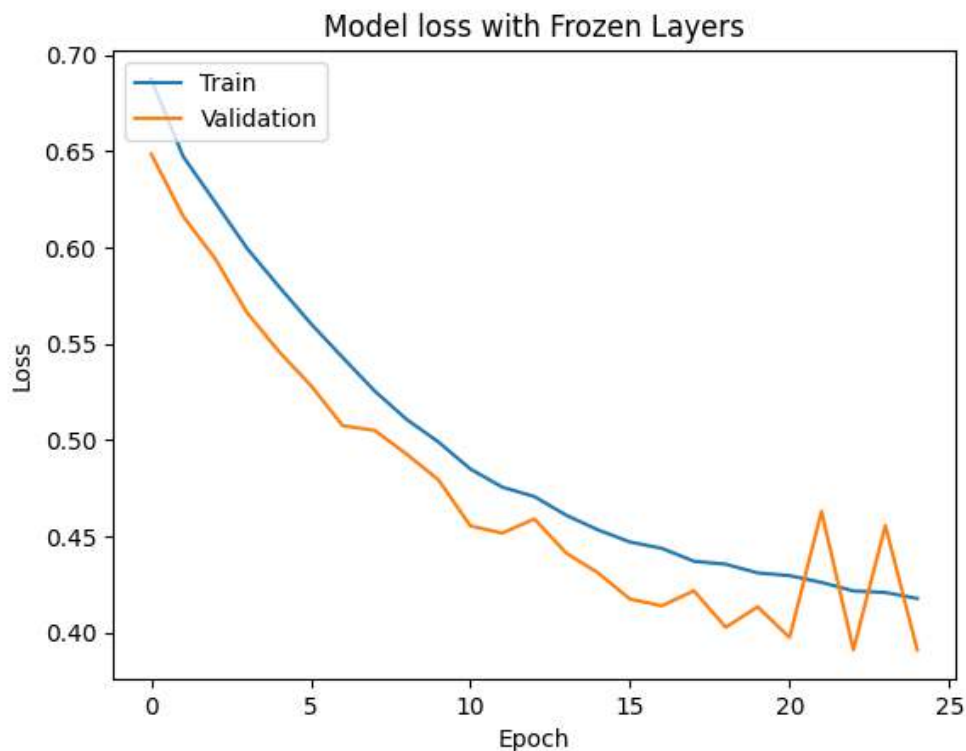


Εικόνα 20 : Training VS Validation Accuracy with frozen layers (ResNet-50)

Για το μοντέλο στο πρώτο στάδιο της εκπαίδευσης με τα παγωμένα επίπεδα η ακρίβεια (Accuracy) εκπαίδευσης όπως φαίνεται και από την εικόνα 20 παρουσιάζει απότομη αύξηση από τις πρώτες εποχές. Αυτή η ταχεία βελτίωση δηλώνει ότι το μοντέλο μαθαίνει γρήγορα τα χαρακτηριστικά που εξάγονται κατά τη διαδικασία της μεταφοράς μάθησης. Με την πάροδο των εποχών συνεχίζει να αυξάνεται αλλά με πιο αργό και σταθερό ρυθμό.

Αντίθετα, η ακρίβεια (Accuracy) επικύρωσης ενώ είναι υψηλότερη από την ακρίβεια εκπαίδευσης στην αρχή που σημαίνει ότι τα χαρακτηριστικά που έχει ήδη μάθει το μοντέλο γενικεύουν καλά στο σύνολο επικύρωσης. Ωστόσο, παρουσιάζει έντονες

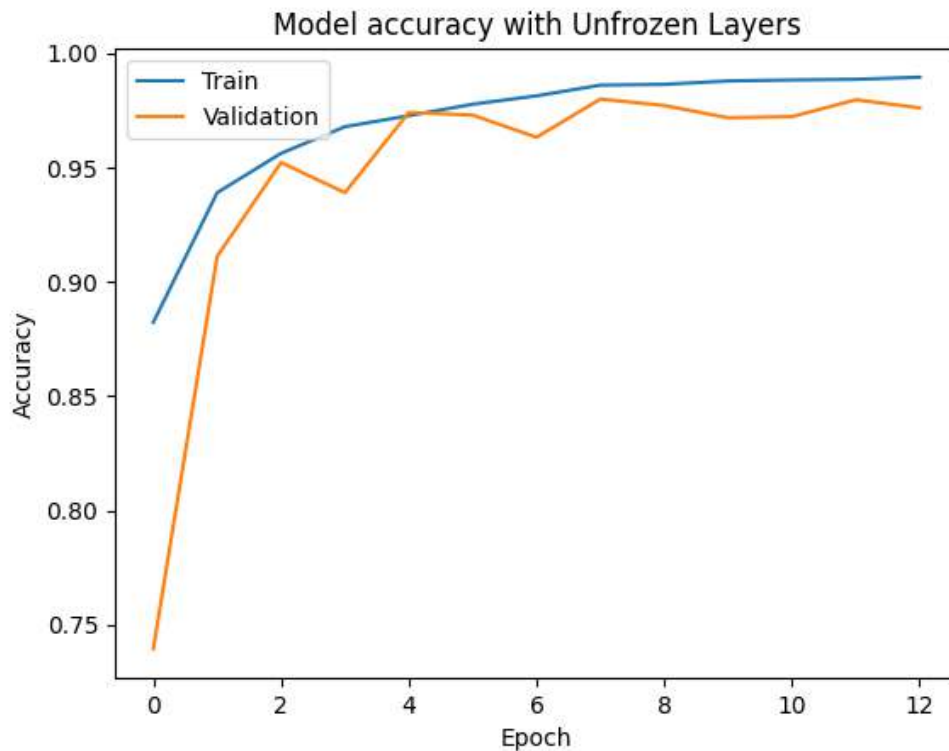
διακυμάνσεις, αυτό θα μπορούσε να σημαίνει ότι το μοντέλο δεν γενικεύει καλά πέρα από τα δεδομένα εκπαίδευσης γεγονός που θα μπορούσε να είναι ένδειξη υπερπροσαρμογής ή ότι τα δεδομένα επικύρωσης δεν είναι πολύ αντιπροσωπευτικά. Είναι επίσης πολύ σημαντικό να επισημανθεί ότι η ακρίβεια (Accuracy) επικύρωσης μετά από τη δέκατη πέμπτη εποχή σταματάει να βελτιώνεται και μάλιστα φαίνεται να μειώνεται γεγονός που μπορεί να δείχνει ότι το μοντέλο έχει μάθει ότι θα μπορούσε να μάθει από την τρέχουσα αρχιτεκτονική και από τα δεδομένα.



Εικόνα 21 : Training VS Validation Loss with frozen layers (ResNet-50)

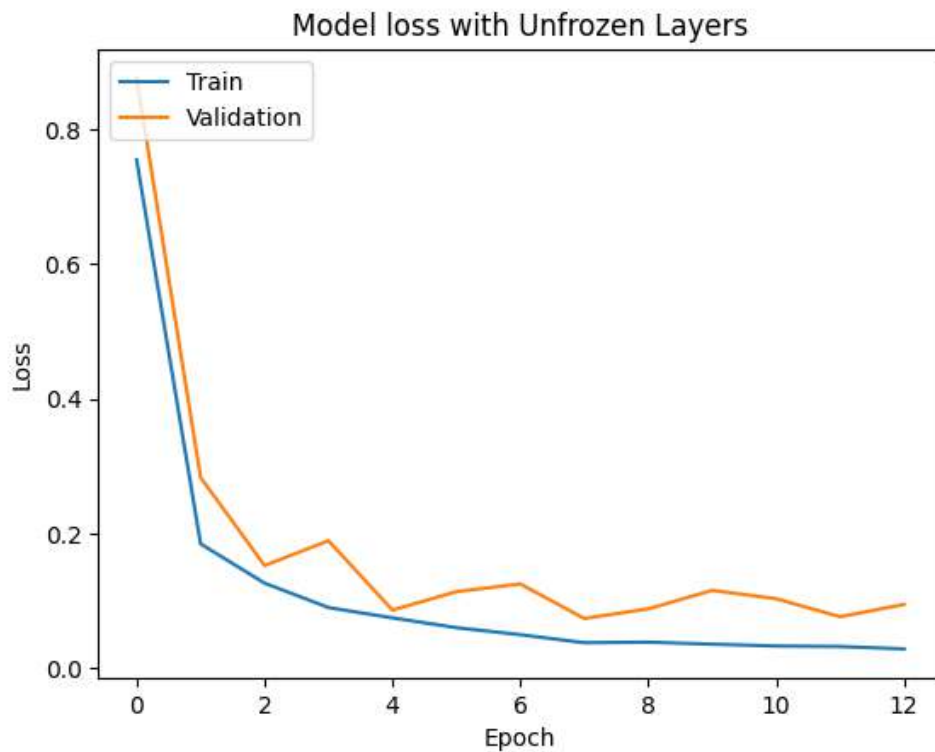
Στην αρχή όπως φαίνεται και από την εικόνα 21 οι τιμές της απώλειας εκπαίδευσης και επικύρωσης ξεκινούν από μια υψηλή τιμή, ωστόσο όσο προχωράει η εκπαίδευση μειώνονται και φαίνεται ότι το μοντέλο αρχίζει να συγκλίνει σε μια ελάχιστη τιμή. Πιο αναλυτικά η απώλεια εκπαίδευσης μειώνεται με έναν σταθερό ρυθμό και σταθεροποιείται γύρω στο 0.40 που σημαίνει ότι μοντέλο ενδεχομένως να μην μπορεί να μειώσει σημαντικά την απώλεια χωρίς να γίνουν κάποιες άλλες προσαρμογές στο μοντέλο ή στη διαδικασία μάθησης.

Αντίθετα, η απώλεια επικύρωσης μειώνεται έως ένα σημείο και στη συνέχεια παρουσιάζει σημαντική διακύμανση ιδιαίτερα μετά τη δέκατη πέμπτη εποχή, συμπεριλαμβανομένης της αύξησης μετά την εικοστή εποχή το οποίο μας προκαλεί ανησυχία καθώς αυτό μπορεί να σημαίνει ότι το μοντέλο μπορεί να μην γενικεύει καλά στα νέα δεδομένα ή ότι τα δεδομένα επικύρωσης δεν είναι πολύ αντιπροσωπευτικά.



Εικόνα 22 : Training VS Validation Accuracy with unfrozen layers (ResNet-50)

Στο δεύτερο στάδιο μετά το «ξεπάγωμα» των επίπεδων και την επέκταση της εκπαίδευσης παρατηρούμε από την εικόνα 22 ότι η βελτίωση της ακρίβειας του μοντέλου είναι αξιοσημείωτη. Η ακρίβεια (Accuracy) εκπαίδευσης έφτασε πολύ γρήγορα σε σχεδόν τέλεια επίπεδα και η ακρίβεια (Accuracy) επικύρωσης ακολουθεί φτάνοντας πάνω από 0.95. Το «ξεπάγωμα» των επίπεδων έδωσε στο μοντέλο τη δυνατότητα να βελτιώσει τα προ εκπαιδευμένα χαρακτηριστικά του με αποτέλεσμα να έχουμε ένα πιο ακριβές μοντέλο που μπορεί να προσαρμόσει τα μαθημένα χαρακτηριστικά του πάνω σε καινούργια δεδομένα.



Εικόνα 23 : Training VS Validation Loss with unfrozen layers (ResNet-50)

Από την εικόνα 23 μπορούμε να δούμε ότι το μοντέλο μετά το «ξεπάγωμα» των επιπέδων όπου είχε τη δυνατότητα να ενημερώσει όλα τα βάρη κατά τη διάρκεια της εκπαίδευσης παρουσίασε μια απότομη μείωση τόσο στην απώλεια εκπαίδευσης όσο και στην απώλεια επικύρωσης. Η απώλεια εκπαίδευσης προσέγγισε σχεδόν μηδενικές τιμές και η απώλεια επικύρωσης σταθεροποιήθηκε σε ένα αρκετά χαμηλό επίπεδο γεγονός που υποδεικνύει την ικανότητα του μοντέλου να γενικεύει από τα δεδομένα εκπαίδευσης στα δεδομένα επικύρωσης καθώς και ότι δεν υπάρχει υπόνοια υπερπροσαρμογής.

Κάνοντας έναν απολογισμό μπορούμε να συμπεράνουμε ότι η στρατηγική να επιτρέψουμε στο μοντέλο να εκπαιδευτεί πλήρως σε όλα τα επίπεδα είναι επωφελής για την ανίχνευση Covid-19 σε ακτινογραφίες. Το πρώτο στάδιο της εκπαίδευσης παρείχε για το μοντέλο ένα καλό σημείο εκκίνησης και με τη μετάβαση στο δεύτερο στάδιο της εκπαίδευσης το μοντέλο κατάφερε να μάθει και να καταγράψει πολύ λεπτά χαρακτηριστικά των εικόνων τα οποία είναι πολύ σημαντικά για τη σωστή διάγνωση.

Στη συνέχεια όπως και για το προηγούμενο μοντέλο (CNN from scratch) θα αξιολογήσουμε την απόδοση του μοντέλου πρόβλεψης ResNet-50 σε ένα νέο σύνολο

δεδομένων στα δεδομένα ελέγχου (test dataset). Οι μετρικές των επιδόσεων που διαμορφώθηκαν όπως φαίνεται στην εικόνα 24 αντικατοπτρίζουν τα συγκεντρωτικά αποτελέσματα μετά και από τα δυο στάδια της εκπαίδευσης.

13/13 [=====] - 1s 78ms/step				
	precision	recall	f1-score	support
Negative	0.98	1.00	0.99	200
Positive	1.00	0.98	0.99	200
accuracy			0.99	400
macro avg	0.99	0.99	0.99	400
weighted avg	0.99	0.99	0.99	400

Εικόνα 24 : Μετρήσεις επιδόσεων στο σύνολο δεδομένων επικύρωσης (ResNet-50)

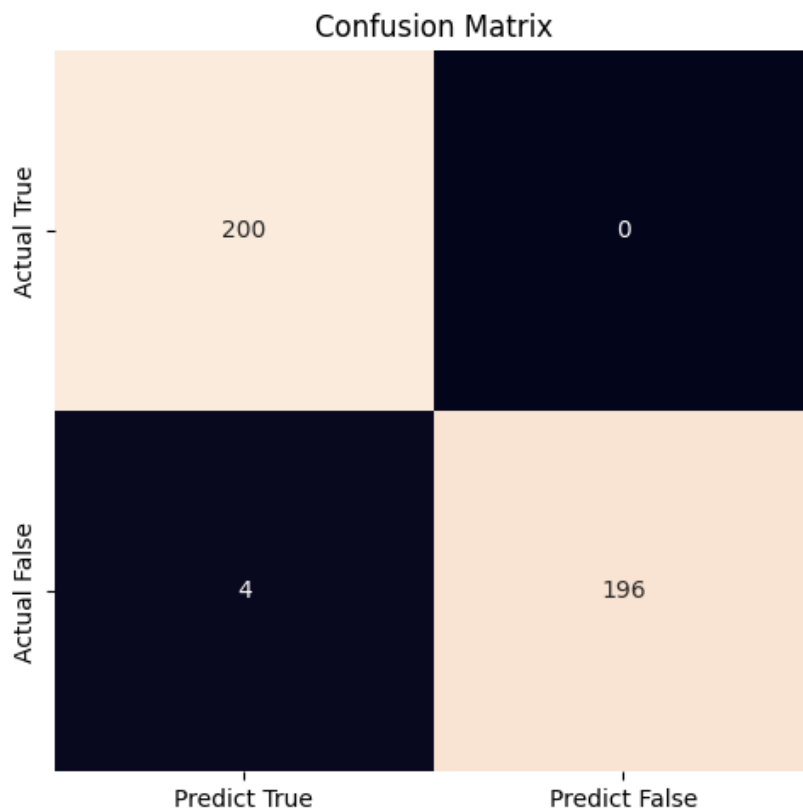
Από τα αποτελέσματα μπορούμε να δούμε ότι το μοντέλο έχει εξαιρετικές επιδόσεις. Πιο συγκριμένα η συνολική ακρίβεια (Accuracy) του μοντέλου είναι 0.99 υποδεικνύοντας ότι ταξινομεί σωστά το 99% όλων των περιπτώσεων και έχει επιτύχει ένα υψηλό επίπεδο αξιοπιστίας.

Το μοντέλο καταφέρνει ακρίβεια (Precision) 0.98 για την αρνητική κλάση που σημαίνει το 98% που πρόβλεψε ως αρνητικές ήταν όντως και για τη θετική κλάση έχει ακρίβεια (Precision) 1.00 που σημαίνει ότι όλες οι περιπτώσεις που πρόβλεψε ως θετικές ήταν και στην πραγματικότητα.

Η ανάκληση (Recall) για την αρνητική κλάση είναι 1.00 που σημαίνει ότι αναγνώρισε σωστά όλες τις αρνητικές περιπτώσεις και η ανάκληση για τη θετική κλάση είναι 0.98 που σημαίνει ότι το 98% των πραγματικών θετικών περιπτώσεων ταξινομήθηκε σωστά.

Το f1-score και για τις δυο κλάσεις είναι 0.99 που δηλώνει ένα ισχυρό μοντέλο με καλή ισορροπία μεταξύ ακρίβειας και ανάκλησης και για τις δύο κλάσεις. Οι μετρικές macro average και weighted average είναι 0.99 για την ακρίβεια (Precision), την ανάκληση (Recall) και για το f1-score που επιβεβαιώνουν την εξαιρετική απόδοση του μοντέλου σε όλες τις κλάσεις.

Εφόσον εξετάσαμε τις μετρικές του μοντέλου πάνω στα δεδομένα ελέγχου παρακάτω με την εικόνα 25 θα εξετάσουμε τον πίνακα σφαλμάτων με τις πραγματικές ετικέτες ο οποίος θα μας δώσει μια πιο σαφή εικόνα για την αποτελεσματικότητα του μοντέλου.



Εικόνα 25 : Πίνακας Σφαλμάτων (ResNet-50)

- **True Positive (TP):** Το μοντέλο αναγνώρισε όλο το σύνολο των πραγματικά θετικών περιπτώσεων (200) αποδεικνύοντας ότι έχει την ικανότητα να αναγνωρίζει σωστά τις περιπτώσεις Covid-19 σε ακτινογραφίες.
- **False Negative (FN):** Το μοντέλο σωστά δεν πρόβλεψε καμία πραγματική θετική περίπτωση ως αρνητική. Αυτό το αποτέλεσμα είναι ιδανικό ιδίως σε μια εργασία όπως αυτή που μελετάμε όπου η λάθος διάγνωση θα μπορούσε να έχει σοβαρές επιπτώσεις.
- **False Positive (FP):** Τα 4 ψευδώς θετικά αποτελέσματα αντιπροσωπεύουν τις λίγες περιπτώσεις στις οποίες το μοντέλο απέτυχε να ανιχνεύσει την απουσία Covid-19.

- *True Negative (TN)*: Ο υψηλός αριθμός πραγματικών αρνητικών (196) δείχνει ότι το μοντέλο έχει την ικανότητα επιβεβαίωσης της απουσίας της νόσου όταν αυτή δεν υπάρχει.

Συνολικά, ο πίνακας σφαλμάτων ενισχύει την υψηλή ακρίβεια του μοντέλου στη διάγνωση του Covid-19, όπως αποδεικνύεται και από τις μετρικές που εξετάσαμε παραπάνω. Καταδεικνύει τη δύναμη του μοντέλου στην ορθή ταξινόμηση των πραγματικών περιπτώσεων με ελάχιστο σφάλμα. Ωστόσο, κατά την κλινική εφαρμογή τέτοιων μοντέλων, οι επιπτώσεις των ψευδώς θετικών αποτελεσμάτων πρέπει να λαμβάνονται εξίσου σοβαρά γιατί μπορεί να οδηγήσουν σε περιττό άγχος και σε άσκοπες εξετάσεις.

ΚΕΦΑΛΑΙΟ 4

Σε αυτό το κεφάλαιο θα παραθέσουμε τα συμπεράσματα στα οποία καταλήξαμε μετά από τις πειραματικές διαδικασίες που διεξήχθησαν σε δυο διαφορετικά μοντέλα για την ανίχνευση Covid-19 πάνω στο ίδιο σύνολο ακτινογραφιών, καθώς και σε μελλοντικές βελτιώσεις που μπορούμε να κάνουμε ώστε να βελτιώσουμε περαιτέρω τα μοντέλα που χρησιμοποιήσαμε. Το πρώτο μοντέλο που χρησιμοποιήσαμε ήταν ένα συνελικτικό νευρωνικό δίκτυο το οποίο σχεδιάστηκε από το μηδέν. Και το δεύτερο μοντέλο ήταν το ResNet-50 με την εξελιγμένη αρχιτεκτονική του που αξιοποιεί τη μεταφορά μάθησης.

4.1 Συμπεράσματα

Το πρώτο μοντέλο (CNN from scratch) παρουσίασε αυξημένη διακύμανση τόσο στην ακρίβεια (accuracy) όσο και στην απώλεια επικύρωσης υποδεικνύοντας πιθανά προβλήματα υπερπροσαρμογής. Οι προσαρμογές του ρυθμού μάθησης φάνηκαν να είναι αντιδραστικές προσπάθειες ώστε να μπορέσει το μοντέλο να σταθεροποιήσει την εκπαίδευση του. Ωστόσο, το μοντέλο στο στάδιο της αξιολόγησης των προβλέψεων πάνω στο σύνολο δεδομένων ελέγχου έδειξε εξαιρετικά υψηλό συνολικό ποσοστό ακρίβειας (accuracy) καθώς και ότι έχει μια ισχυρή ικανότητα να ταξινομεί σωστά τα δεδομένα στις κλάσεις που ανήκουν. Αν και έδειξε αρκετά ενθαρρυντικά αποτελέσματα ανέδειξε και τις προκλήσεις και τα προβλήματα που πρέπει να αντιμετωπίσουμε όταν δημιουργούμε ένα συνελικτικό νευρωνικό δίκτυο από το μηδέν όπως είναι η ανάγκη για μια πιο προσεκτική ρύθμιση των υπερπαραμέτρων και ενός πιο σταθερού ρυθμού μάθησης για τη βελτίωση της γενίκευσης του σε νέα δεδομένα.

Το μοντέλο ResNet-50 έδειξε καλύτερες επιδόσεις και δυνατότητες γενίκευσης από το πρώτο μοντέλο. Οι γραφικές παραστάσεις της ακρίβειας (accuracy) και της απώλειας επικύρωσης απεικόνιζαν ένα αρκετά σταθερό πρότυπο μάθησης και ιδιαίτερα στο στάδιο μετά το «ξεπάγωμα» των επιπέδων το οποίο επέτρεψε στο μοντέλο να κάνει μια πιο λεπτομερή προσαρμογή πάνω στο σύνολο δεδομένων. Το τελικό στάδιο της εκπαίδευσης με τα μη παγωμένα επίπεδα αποδείχθηκε ιδιαίτερα αποτελεσματικό και ενίσχυσε την ικανότητα του μοντέλου να μπορεί να ταξινομεί τα δεδομένα με περισσότερη ακρίβεια.

4.2 Μελλοντικές εργασίες και βελτιστοποιήσεις

Οποιοδήποτε μοντέλο και να αναπτυχθεί στο μέλλον θα χρειαστεί εκτεταμένη κλινική επικύρωση για να διασφαλιστεί η αποτελεσματικότητα του ώστε να μπορέσει να φανεί χρήσιμο σε ένα πραγματικό ιατρικό περιβάλλον. Καθώς και η αξιολόγηση του σε ένα μεγαλύτερο σύνολο ελέγχου με διαφορετικά δημογραφικά χαρακτηριστικά και από διαφορετικά ακτινοδιαγνωστικά μηχανήματα.

Όσον αφορά τις μελλοντικές βελτιστοποιήσεις που θα μπορούσαν να γίνουν, αυτές μπορούν να αφορούν την εφαρμογή ενός πιο εξειδικευμένου ρυθμού μάθησης όπου θα προσαρμόζεται ο ρυθμός με βάση τις μετρικές απώλειας και θα συμβάλλει στην καλύτερη σύγκλιση και σταθερότητα κατά τη διάρκεια της εκπαίδευσης. Επίσης, θα μπορούσε να εξεταστεί το ενδεχόμενο να γίνει επέκταση των δεδομένων εκπαίδευσης είτε με τη χρήση διαφορετικών συνόλων δεδομένων είτε με την επαύξηση του συνόλου δεδομένων που χρησιμοποιήσαμε με διαφορετικές τεχνικές τροποποίησης των δεδομένων.

Συνοψίζοντας ενώ τα αποτελέσματα και από τα δυο μοντέλα είναι αρκετά ελπιδοφόρα, υπάρχει αρκετά μεγάλο περιθώριο βελτίωσης. Οι συνεχείς βελτιώσεις των αρχιτεκτονικών των μοντέλων, μια πιο συνεπής επιμέλεια των δεδομένων εκπαίδευσης και η δοκιμή διαφορετικών στρατηγικών εκπαίδευσης μπορούν να οδηγήσουν σε βελτιώσεις της ακρίβειας και της αξιοπιστίας των μοντέλων για την ανίχνευση Covid-19 από ακτινογραφίες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «World Health Organization,» [Ηλεκτρονικό]. Available: <https://www.who.int/news-room/questions-and-answers/item/coronavirus-disease-covid-19>.
- [2] E. S. 1. B. v. G. K. G. v. L. K. M. Erdi Çallı 1, «Deep learning for chest X-ray analysis: A survey,» *Medical Image Analysis*, 5 June 2021.
- [3] S. Chukekar, Learning Outcomes of Classroom Research, L ORDINE NUOVO PUBLICATION, 2021.
- [4] B. Mahesh, «Machine Learning Algorithms - A Review,» *International Journal of Science and Research (IJSR)*, pp. 381-386, 2020.
- [5] Y. LeCun, Y. Bengio και G. & Hinton, «Deep learning,» p. 436–444, 2015.
- [6] M. Lefkowitz, «Professor's perceptron paved the way for AI – 60 years too soon,» *Cornell Chronicle*, 2019.
- [7] Α. Γεωργούλη, «Τεχνητή Νοημοσύνη» – Μια εισαγωγική προσέγγιση, Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιοβιοθηκών, 2015.
- [8] M. C. Nwadiugwu, «Neural networks, artificial intelligence and the computational brain,» *arXiv preprint arXiv:2101.08635*, 2020.
- [9] D. a. P. F. M. a. S. A. a. G. M. a. M. J. Gandolfi, «Brain-inspired computing: Neuroscience drives the development of new electronics and artificial intelligence,» *Frontiers in Cellular Neuroscience*, 2022.
- [10] C. P. G. B. S. K. Q. W. Nils Bjorck, «Understanding Batch Normalization,» σε *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*.
- [11] N. a. H. G. a. K. A. a. S. I. a. S. R. Srivastava, «Dropout: a simple way to prevent neural networks from overfitting,» *The journal of machine learning research*, pp. 1929--1958, 2014.
- [12] W. a. F. Y.-L. a. Z. D. Xu, «ResNet and Its Application to Medical Image Processing: Research Progress and Challenges,» *Computer Methods and Programs in Biomedicine*, 2023.
- [13] X. Z. S. R. J. S. Kaiming He, «Deep Residual Learning for Image Recognition,» 2015.
- [14] N. Donges, «What Is Transfer Learning? Exploring the Popular Deep Learning Approach.,» 2022. [Ηλεκτρονικό]. Available: <https://builtin.com/data-science/transfer-learning>.
- [15] S. a. S. S. a. A. A. Sharma, «Activation functions in neural networks,» *Towards Data Sci*, pp. 310-316, 2017.

- [16] R. Parmar, «Common Loss functions in machine learning,» [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23>.
- [17] R. Gombru, «Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names,» 2018. [Ηλεκτρονικό]. Available: https://gombru.github.io/2018/05/23/cross_entropy_loss/.
- [18] R.-Y. Sun, «Optimization for deep learning: An overview,» *Journal of the Operations Research Society of China*, pp. 249-294, 2020.
- [19] S. Doshi, «Various Optimization Algorithms For Training Neural Network,» 2019. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6>.
- [20] S. Ruder, «An overview of gradient descent optimization algorithms,» *arXiv*, 2016.
- [21] S. Dev, «Overfitting vs Underfitting,» 27 Δεκέμβριος 2022. [Ηλεκτρονικό]. Available: <https://medium.com/mlearning-ai/overfitting-vs-underfitting-6a41b3c6a9ad>.
- [22] G. H. K. S. S. Nitish Srivastava, «Dropout: A Simple Way to Prevent Neural Networks from Overfitting,» 2014.
- [23] D. E. a. D. R. a. G. R. a. C. Y. Rumelhart, «Backpropagation: The basic theory,» σε *Backpropagation*, Psychology Press, 2013, pp. 1-34.
- [24] [Ηλεκτρονικό]. Available: <https://www.geeksforgeeks.org/vanishing-and-exploding-gradients-problems-in-deep-learning/>.
- [25] [Ηλεκτρονικό]. Available: <https://keras.io/about/>.

ΠΑΡΑΡΤΗΜΑ Ι

```
#Εισαγωγή των βιβλιοθηκών
import cv2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import os
import seaborn as sns
from sklearn.model_selection import train_test_split

#Εισαγωγή των δεδομένων
data_path = os.path.join('data')
train_data_path = os.path.join(data_path, 'train')
test_data_path = os.path.join(data_path, 'test')
train_df = pd.read_csv(os.path.join(data_path, 'train.txt'), sep=" ",
header=None)
test_df = pd.read_csv(os.path.join(data_path, 'test.txt'), sep=" ",
header=None)

#Έλεγχος της δομής των δεδομένων
test_df.head()

train_df.head()

#Αλλαγή των ονομάτων των στηλών
train_df = train_df.rename(columns = {1:'ImageId', 2:'Diagnosis'})
test_df = test_df.rename(columns = {1:'ImageId', 2:'Diagnosis'})
test_df.head()

#Διαγραφή της τρίτης στήλης
train_df.drop(columns = [0,3], axis=1, inplace=True)
test_df.drop(columns = [0,3], axis=1, inplace=True)

train_df.ImageId[9]

test_df.head()

#Αποθήκευση των τροποποιημένων αρχείων txt train και test
train_df.to_csv("data/trainWithDiagnosisColumn.txt", sep='\t', index=False)
test_df.to_csv("data/testWithDiagnosisColumn.txt", sep='\t', index=False)

#Καθορισμός του όγκου των δεδομένων που ανήκουν στη κάθε κλάση
train_df.Diagnosis.value_counts()

test_df.Diagnosis.value_counts()

classes_test = np.unique(test_df.Diagnosis)
```

```

classes_test

classes_train = np.unique(train_df.Diagnosis)
classes_train
#Διαγραμματική απεικόνιση των δεδομένων εκπαίδευσης και ελέγχου
trainData= train_df['Diagnosis'].value_counts()
testData = test_df['Diagnosis'].value_counts()
df = pd.DataFrame({
    'Dataset': ['Train'] * len(trainData) + ['Test'] * len(testData),
    'Class': list(trainData.index) * 2,
    'Count': list(trainData.values) + list(testData.values)})
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
colors = ['blue' if label == 'negative' else 'red' for label in df['Class']]
ax = sns.barplot(x='Dataset', y='Count', hue='Class', data=df, palette=colors)
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2.,
p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset
points', fontsize=10, color='black')
plt.xlabel('Dataset')
plt.ylabel('Count')
plt.show()

#Διαχωρισμός των δεδομένων εκπαίδευσης σε δεδομένα επικύρωσης
train_df, val_df = train_test_split(train_df, test_size=0.2, random_state=42)

#Διαγραμματική απεικόνιση των δεδομένων εκπαίδευσης ,ελέγχου και επικύρωσης
trainData = train_df['Diagnosis'].value_counts()
testData = test_df['Diagnosis'].value_counts()
valData = val_df['Diagnosis'].value_counts()
df = pd.DataFrame({
    'Dataset': ['Train'] * len(trainData)+['Val'] * len(valData) + ['Test'] *
len(testData),
    'Class': list(trainData.index) * 3,
    'Count': list(trainData.values)+list(valData.values) +
list(testData.values)})
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
colors = ['blue' if label == 'negative' else 'red' for label in df['Class']]
ax = sns.barplot(x='Dataset', y='Count', hue='Class', data=df, palette=colors)
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2.,
p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset
points', fontsize=10, color='black')
plt.xlabel('Dataset')
plt.ylabel('Count')
plt.show()

```

```

#Διαγραμματική απεικόνιση μιας τυχαίας εικόνας από το dataset και εκτύπωση του
#μέγεθός της
indx = np.random.randint(0,len(train_df),1)[0]
img = cv2.imread(os.path.join(train_data_path, train_df.ImageId[indx]))
plt.imshow(img)
plt.title(train_df.Diagnosis[indx])
plt.show()
img.shape

#Resizing image 224*224
def imageResize(inputPath,outputPath):
    os.makedirs(outputPath, exist_ok=True)
    for file in os.listdir(inputPath):
        path = os.path.join(inputPath, file)
        image = cv2.imread(path)
        img_resize = cv2.resize(image, (224,224))
        output_path = os.path.join(outputPath, file)
        cv2.imwrite(outputPath,img_resize)
outputPath="/data/testResizing"
inputPath="/data/testResizing"
imageResize(inputPath,outputPath)
data_path = os.path.join('data')
train_data_path = os.path.join(data_path, 'trainResizing')
test_data_path = os.path.join(data_path,'testResizing')

#Έλεγχος του μεγέθους των τροποποιημένων εικόνων
indx = np.random.randint(0,len(train_df),1)[0]
img = cv2.imread(os.path.join(train_data_path, train_df.ImageId[indx]))
plt.imshow(img)
plt.title(train_df.Diagnosis[indx])
plt.show()
img.shape

#Ελάχιστη και μέγιστη τιμή των εικονοστοιχείων μιας εικόνας
img_array = np.array(img)
print("Min pixel value:", np.min(img_array))
print("Max pixel value:", np.max(img_array))

#Κανονικοποίηση μιας εικόνας και απεικόνιση της
img_array = np.array(img)
img_array_normal = img_array / 255.0
print("Min pixel :", np.min(img_array_normal))
print("Max pixel :", np.max(img_array_normal))

plt.imshow(img_array_normal)
plt.title(train_df.Diagnosis[indx])
plt.show()
img_array_normal.shape

```

ΠΑΡΑΡΤΗΜΑ II

```
#Εισαγωγή των βιβλιοθηκών
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
import seaborn as sns
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, BatchNormalization, MaxPooling2D, Dropout,
Flatten, Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint,
ReduceLROnPlateau
from tensorflow.keras.metrics import AUC, Precision, Recall
from tensorflow.keras.models import load_model

#Εισαγωγή των δεδομένων
data_path = os.path.join('data')
train_data_path = os.path.join(data_path, 'trainResizing')
test_data_path = os.path.join(data_path, 'testResizing')
train_df = pd.read_csv(os.path.join(data_path,
'trainWithDiagnosisColumn.txt'), sep="\t", index_col=None)
test_df = pd.read_csv(os.path.join(data_path, 'testWithDiagnosisColumn.txt'),
sep="\t", index_col=None)

#Διαχωρισμός των δεδομένων εκπαίδευσης σε δεδομένα επικύρωσης
train_df, val_df = train_test_split(train_df, test_size=0.2, random_state=42)

#Τεχνικές επεξεργασίας και επαύξησης των δεδομένων εκπαίδευσης ,επικύρωσης
train_data_generator = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    fill_mode='nearest')

batch_size = 32
target_size=(224, 224)

train_generator = train_data_generator.flow_from_dataframe(
    train_df,
    directory=train_data_path,
```

```

    x_col='ImageId',
    y_col='Diagnosis',
    target_size=target_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=True)

print(train_generator.class_indices)

val_data_generator = ImageDataGenerator(rescale=1./255)

val_generator = val_data_generator.flow_from_dataframe(
    val_df,
    directory=train_data_path,
    x_col='ImageId',
    y_col='Diagnosis',
    target_size=target_size,
    batch_size=batch_size ,
    class_mode='binary',
    shuffle=False)

print(val_generator.class_indices)

#Ορισμός του Δικτύου
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

#Compile the network
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy',
metrics=["accuracy", "AUC", "Precision", "Recall"])

model.summary()

```

```

#Ορισμός των Callbacks
checkpoint_model = ModelCheckpoint('model_from_scratch.h5',
save_best_only=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=2,
min_lr=0.0001)
early_stopping = EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True)

#Εκπαίδευση του δικτύου
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=25,
    validation_data=val_generator,
    validation_steps=val_generator.samples // batch_size,
    callbacks=[early_stopping, checkpoint_model, reduce_lr])

#Διαγραμματική απεικόνιση Ακρίβειας επικύρωσης και εκπαίδευσης
print("Plot training & validation accuracy values ")
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy ')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

#Διαγραμματική απεικόνιση απώλειας επικύρωσης και εκπαίδευσης
print("Plot training & validation loss values")
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

#Διαγραμματική απεικόνιση ρυθμού μάθησης
plt.plot(history.history['lr'])
plt.title('Learning Rate')
plt.legend(['lr'], loc='upper left')
plt.show()

# Κανονικοποίηση δεδομένων ελέγχου
test_generator= val_data_generator.flow_from_dataframe(
    test_df,
    directory=test_data_path,
    x_col='ImageId',
    y_col='Diagnosis',

```

```

    target_size=target_size,
    batch_size=batch_size ,
    class_mode='binary',
    shuffle=False)

print(test_generator.class_indices)

model_load = load_model('model_from_scratch.h5')

#Μετρικές αξιολόγησης μοντέλου
evaluation_results= model_load.evaluate(test_generator)
print(f"Test Loss: {evaluation_results[0]}")
print(f"Test Accuracy: {evaluation_results[1]}")

prediction_results = model_load.predict(test_generator)
prediction = np.where(prediction_results > 0.5, 1, 0)
metrics_report = classification_report(test_generator.labels, prediction,
target_names=['Negative', 'Positive'])
print(metrics_report)

#Πίνακας σφαλμάτων
table = confusion_matrix(test_generator.labels,prediction)
matrix_confusion = pd.DataFrame(table, index=['Actual True', 'Actual False'],
                                columns=['Predict True', 'Predict False'])
fig, ax = plt.subplots(figsize=(5,5))
sns.heatmap(data=matrix_confusion, annot=True, fmt='g', ax=ax, cbar=None)
plt.title("Confusion Matrix")
plt.tight_layout()
plt.show()

```

ΠΑΡΑΡΤΗΜΑ ΙΙΙ

```
#Εισαγωγή των βιβλιοθηκών
import os
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.metrics import AUC, Precision, Recall
from keras import Sequential
from tensorflow.keras.models import load_model

#Εισαγωγή των δεδομένων
data_path = os.path.join('data')
train_data_path = os.path.join(data_path, 'trainResizing')
test_data_path = os.path.join(data_path, 'testResizing')
train_df = pd.read_csv(os.path.join(data_path,
'trainWithDiagnosisColumn.txt'), sep="\t", index_col=None)
test_df = pd.read_csv(os.path.join(data_path, 'testWithDiagnosisColumn.txt'),
sep="\t", index_col=None)

#Διαχωρισμός των δεδομένων εκπαίδευσης σε δεδομένα επικύρωσης
train_df, val_df = train_test_split(train_df, test_size=0.2, random_state=42)

batch_size = 32
target_size=(224, 224)

#Τεχνικές επεξεργασίας και επαύξησης των δεδομένων εκπαίδευσης ,επικύρωσης
train_data_generator = ImageDataGenerator(
    rescale=1./255,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    fill_mode='nearest'
)

train_generator = train_data_generator.flow_from_dataframe(
```

```

train_df,
directory=train_data_path,
x_col='ImageId',
y_col='Diagnosis',
target_size=target_size,
batch_size=batch_size,
class_mode='binary',
shuffle=True)
val_data_generator = ImageDataGenerator(rescale=1./255)
val_generator = val_data_generator.flow_from_dataframe(
    val_df,
    directory=train_data_path,
    x_col='ImageId',
    y_col='Diagnosis',
    target_size=target_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False
)

#Ορισμός του Δικτύου
model = Sequential([
    ResNet50(include_top=False, weights='imagenet', input_shape=(224, 224,
3)),
    GlobalAveragePooling2D(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')])

#Πάγωμα των επιπέδων
for layer in model.layers[:-3]:
    layer.trainable = False

#Compile the network
model.compile(optimizer=Adam(learning_rate=1e-4), loss='binary_crossentropy',
metrics=['accuracy', AUC(), Precision(), Recall()])

#Ορισμός των Callbacks
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=2,
min_lr=1e-7, verbose=1)
early_stopping = EarlyStopping(monitor='val_loss',
patience=5, restore_best_weights=True)

#Εκπαίδευση του δικτύου
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=25,
    validation_data=val_generator,

```

```

validation_steps=val_generator.samples // val_generator.batch_size,
callbacks=[reduce_lr, early_stopping])

#Ξεπάγωμα των επιπέδων
for layer in model.layers[:-3]:
    layer.trainable = True

#Compile the network
model.compile(optimizer=Adam(learning_rate=1e-5), loss='binary_crossentropy',
metrics=['accuracy',AUC(), Precision(), Recall()])

#Εκπαίδευση του δικτύου
history_unfreeze = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=15,
    validation_data=val_generator,
    validation_steps=val_generator.samples // val_generator.batch_size,
    callbacks=[reduce_lr, early_stopping])
model.summary()

#Διαγραμματική απεικόνιση Ακρίβειας επικύρωσης και εκπαίδευσης με τα παγωμένα
#επίπεδα
print("Plot training & validation accuracy values with Frozen Layers")
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy with Frozen Layers')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
print("Plot training & validation loss values")
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss with Frozen Layers')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

#Διαγραμματική απεικόνιση Ακρίβειας επικύρωσης και εκπαίδευσης μετά το
#ξεπάγωμα των επιπέδων

print("Plot training & validation accuracy values")
plt.plot(history_unfreeze.history['accuracy'])
plt.plot(history_unfreeze.history['val_accuracy'])
plt.title('Model accuracy with Unfrozen Layers')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')

```

```

plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
print("Plot training & validation loss values")
plt.plot(history_unfreeze.history['loss'])
plt.plot(history_unfreeze.history['val_loss'])
plt.title('Model loss with Unfrozen Layers')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

#Αποθήκευση του εκπαιδευμένου μοντέλου
model.save('model_res_fr_unfr.h5')
# Κανονικοποίηση δεδομένων ελέγχου
test_generator = val_data_generator.flow_from_dataframe(
    test_df,
    directory=test_data_path,
    x_col='ImageId',
    y_col='Diagnosis',
    target_size=target_size,
    batch_size=batch_size,
    class_mode='binary',
    shuffle=False)
print(test_generator.class_indices)

#Μετρικές αξιολόγησης μοντέλου
evaluation_results = model_load.evaluate(test_generator,
steps=test_generator.samples // test_generator.batch_size)
print(f"Test Loss: {evaluation_results[0]}")
print(f"Test Accuracy: {evaluation_results[1]}")

prediction_results = model_load.predict(test_generator)
prediction = np.where(prediction_results > 0.5, 1, 0)
metrics_report = classification_report(test_generator.labels, prediction,
target_names=['Negative', 'Positive'])
print(metrics_report)

#Πίνακας σφαλμάτων
table = confusion_matrix(test_generator.labels,prediction)
matrix_confusion = pd.DataFrame(table, index=['Actual True', 'Actual False'],
columns=['Predict True', 'Predict False'])

fig, ax = plt.subplots(figsize=(5,5))

sns.heatmap(data=matrix_confusion, annot=True, fmt='g', ax=ax, cbar=None)
plt.title("Confusion Matrix")
plt.tight_layout()
plt.show()

```