



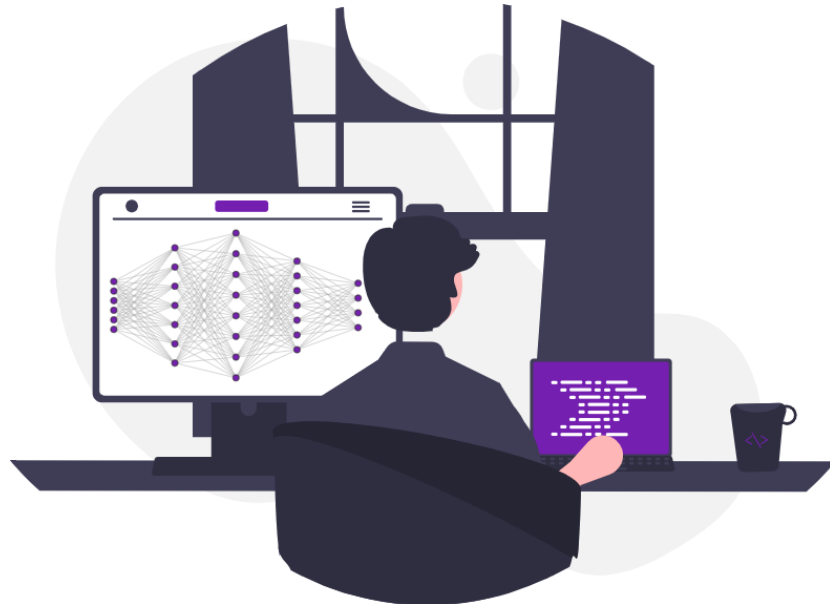
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Σχεδιασμός και υλοποίηση διαδικτυακής εφαρμογής για την κατηγοριοποίηση
όγκων του μαστού με χρήση Συνελκτικών Νευρωνικών Δικτύων



Φοιτητής: Χριστόφορος Κόρμπος
ΑΜ: 19387115

Επιβλέπον μέλος ΔΕΠ

Δρ. Γρηγόριος Κουλούρας
Αναπληρωτής Καθηγητής

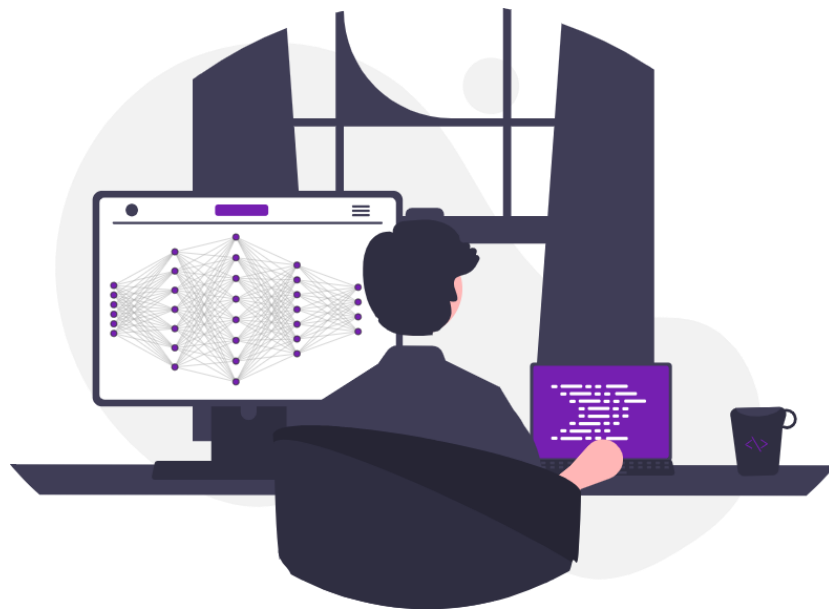
ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, Ιούλιος 2024



UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

Design and Implementation of a Web Application for Breast Tumors Classification through Convolutional Neural Networks



Student: Christopher Kormpos
Registration Number: 19387115

Supervisor

Dr. Grigorios Koulouras
Associate Professor

ATHENS-EGALEO, July 2024

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

Γρηγόριος Κουλούρας Αναπληρωτής Καθηγητής (Επιβλέπον μέλος ΔΕΠ)	Αλέξανδρος Αλεξανδρίδης Καθηγητής	Ηλίας Ζωής Αναπληρωτής Καθηγητής
(Υπογραφή)	(Υπογραφή)	(Υπογραφή)

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Χριστόφορος Κόρμπος,
Ιούλιος, 2024**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα του και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Χριστόφορος Κόρμπος του Φωτίου, με αριθμό μητρώου 19387115 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο Δηλών
Χριστόφορος Κόρμπος



A king's portrait,
Knowledge is the mind's fortress,
Patient as watching the growing orchard.

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, αρχικά θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Γρηγόριο Κουλούρα, για τις πολύτιμες συμβουλές, την καθοδήγηση και την εμπιστοσύνη που μου έδειξε από την αρχή. Η ελευθερία στην επιλογή ενός θέματος που συνδύασε κάποιες τεχνολογίες που είχα στο μυαλό μου και η στήριξή του κατά τη διάρκεια της έρευνας και της συγγραφής αυτής εργασίας αποτέλεσαν τους βασικούς πυλώνες για την εξαιρετική συνεργασία μας. Θερμές ευχαριστίες απευθύνω επίσης στους φίλους και συμφοιτητές μου για τη συνεργασία και τη βοήθειά τους σε αυτή την πορεία. Οι συζητήσεις και οι κοινές μας προσπάθειες συνέβαλαν σημαντικά στην ανάδειξη αυτού του κεφαλαίου της ζωής μας που ονομάζεται πανεπιστήμιο. Τέλος, θα ήθελα να εκφράσω την αμέριστη ευγνωμοσύνη μου στην οικογένειά μου για την απεριόριστη αγάπη τους και την υποστήριξη που παρείχαν κατά τη διάρκεια των σπουδών μου, καθώς και για την καθοριστική επίδρασή τους στη διαμόρφωση της προσωπικότητάς μου.

Περίληψη

Η ραγδαία εξέλιξη της τεχνητής νοημοσύνης, σε συνδυασμό με την αύξηση της ταχύτητας των υπολογιστικών συστημάτων, καθώς και της ακρίβειας των μοντέλων μηχανικής μάθησης, έχουν επωφελήσει αρκετούς τομείς, ένας εκ των οποίων είναι και αυτός της ιατρικής και βιοϊατρικής τεχνολογίας. Συγκεκριμένα, η αναγνώριση εικόνων και προτύπων μέσω της βαθιάς μάθησης και η εκπαίδευση των Νευρωνικών Δικτύων με μεγάλο όγκο δεδομένων έχουν φέρει επαναστατικές αλλαγές στην αναγνώριση και κατηγοριοποίηση ασθενειών κάθε είδους. Η ικανότητα των Νευρωνικών Δικτύων να αναγνωρίζουν πρότυπα και να κάνουν κατηγοριοποιήσεις συχνά μπορεί να ξεπερνάει την ανθρώπινη ακρίβεια, επιτρέποντας με αυτόν τον τρόπο την προτεραιοποίηση σοβαρών περιστατικών ή ακόμα και την αντιμετώπιση περίπλοκων ιατρικών προκλήσεων.

Στα πλαίσια της Διπλωματικής Εργασίας θα αναπτυχθεί ένα μοντέλο τεχνητής νοημοσύνης χρησιμοποιώντας Συνελικτικά Νευρωνικά Δίκτυα (CNNs) για την αναγνώριση όγκων από υπέρηχους στον μαστό και την ταξινόμηση αυτών ως καλοήθεις ή κακοήθεις με τη βοήθεια των βιβλιοθηκών Tensorflow και Keras. Επιπλέον, θα υλοποιηθεί μια διαδικτυακή εφαρμογή, με Python και ReactJS που θα δίνει έμφαση στο user experience και θα επιτρέπει την ευκολότερη πρόσβαση και χρήση από τον καθένα. Η αυτονομία που θα προσφέρει η διαδικτυακή εφαρμογή συνδυάζεται με τη δυνατότητα αλληλεπίδρασης και μέσω κινητών τηλεφώνων, παρέχοντας ολοκληρωμένη λύση για την άμεση ανίχνευση και προσφέροντας τελικά ένα πολύτιμο εργαλείο για τη διάγνωση και παρακολούθηση των παθήσεων του μαστού.

Λέξεις – κλειδιά

Τεχνητή Νοημοσύνη, Συνελικτικά Νευρωνικά Δίκτυα, Υπολογιστική Όραση, Βιοιατρική Τεχνολογία, Κατηγοριοποίηση Όγκων, Διαδικτυακή Εφαρμογή, ReactJS

Abstract

The rapid growth of artificial intelligence, coupled with an increase in computational speed, as well as the precision of machine learning models nowadays, has benefited several sectors, such as the medical and biomedical technology field. Specifically, image and pattern recognition through deep learning, along with Neural Networks being trained on large datasets, have brought revolutionary changes to the recognition and categorization for diseases of all kinds. The proficiency of Deep Neural Networks in pattern recognition and precise classifications that often surpass human accuracy enables the correct prioritization of critical cases and effective addressing of complex medical challenges.

As part of the Thesis, an artificial intelligence model will be developed using Convolutional Neural Networks (CNNs) for identifying breast tumors from ultrasound images and then characterizing them as benign or malignant using the Tensorflow and Keras libraries. Additionally, a web application that emphasizes in user experience and easier access for everyone will be developed using Python and ReactJS. The autonomy offered by the web application, combined with the ability to interact through mobile phones, will provide a comprehensive solution for direct detections and will ultimately offer a valuable tool for the diagnosis and monitoring of breast cancer.

Keywords

Artificial Intelligence, Convolutional Neural Networks, Computer Vision, Biomedical Technology, Tumor Classification, Web Application, ReactJS

Περιεχόμενα

Κατάλογος Πινάκων.....	9
Κατάλογος Εικόνων	9
Αλφαβητικό Ευρετήριο.....	11
Εισαγωγή.....	12
Αντικείμενο της διπλωματικής εργασίας.....	13
Σκοπός και στόχοι	13
Μεθοδολογία.....	13
Καινοτομία	14
Δομή Διπλωματικής Εργασίας	14
ΚΕΦΑΛΑΙΟ 1^ο : Τεχνητή Νοημοσύνη και Νευρωνικά Δίκτυα.....	15
1.1 Τεχνητή Νοημοσύνη.....	15
1.1.1 Η εξέλιξη της τεχνητής νοημοσύνης.....	16
1.1.2 Υπολογιστική Νοημοσύνη	17
1.1.3 Μηχανική Μάθηση.....	18
1.2 Νευρωνικά Δίκτυα	19
1.2.1 Τεχνητοί Νευρώνες.....	19
1.2.2 Συναρτήσεις ενεργοποίησης	21
1.2.3 Υπερπαράμετροι δικτύου	24
1.2.4 Αρχιτεκτονικές Νευρωνικών Δικτύων	26
1.3 Συνελικτικά Νευρωνικά Δίκτυα	28
1.3.1 Συνέλιξη Πινάκων για την ανίχνευση ακρών	28
1.3.2 Επίπεδα Συνελικτικών Νευρωνικών Δικτύων.....	30
1.3.3 Προεκπαιδευμένα μοντέλα και Μεταφορά Μάθησης	32
1.3.4 Αρχιτεκτονικές διαφόρων μοντέλων.....	33
1.4 Συμπεράσματα Κεφαλαίου	37
ΚΕΦΑΛΑΙΟ 2^ο : Ανάλυση δεδομένων και κατασκευή μοντέλου.....	38
2.1 Τεχνολογίες Μοντέλου.....	38
2.1.1 Εργαλεία ανάπτυξης λογισμικού.....	38
2.1.2 Υπολογιστικό περιβάλλον.....	39
2.2 Επισκόπηση προηγούμενης έρευνας	40
2.3 Προεπεξεργασία δεδομένων.....	42
2.3.1 Διαθέσιμο σύνολο δεδομένων.....	42
2.3.2 Επαύξηση συνόλου δεδομένων	43
2.3.3 Μετασχηματισμός δεδομένων.....	46
2.4 Εκπαίδευση Νευρωνικών Δικτύων	47
2.5 Παρουσίαση και σύγκριση αποτελεσμάτων	49
2.6 Συμπεράσματα Κεφαλαίου	53
ΚΕΦΑΛΑΙΟ 3^ο : Ανάπτυξη διαδικτυακής εφαρμογής.....	54
3.1 Διακομιστής.....	54
3.2 Γραφικό περιβάλλον χρήστη	56
3.2.1 Σελίδα παρουσίασης εφαρμογής.....	56
3.2.2 Σελίδα εφαρμογής.....	57
3.3 Φιλοξενία εφαρμογής.....	58
3.4 Τελικό αποτέλεσμα.....	59
ΚΕΦΑΛΑΙΟ 4^ο : Συμπεράσματα - Βελτιώσεις	61

Βιβλιογραφία – Αναφορές.....	63
Παράρτημα Α – Κώδικας για την υλοποίηση μοντέλου και διακομιστή.....	65
data_preprocessing.ipynb.....	65
nasnet.ipynb	70
predict_app.py	72
Παράρτημα Β – Κώδικας διαδικτυακής εφαρμογής	74
index.html.....	74
index.js file.....	74
App.js file	75
Application.jsx.....	75
package.json	78

Κατάλογος Πινάκων

Πίνακας 1: Διαδικασία επαύξησης και συνολικός αριθμός δεδομένων μετά από κάθε βήμα	44
Πίνακας 2: Επίπεδα και υπερπαράμετροι δικτύων.....	47
Πίνακας 3: Αποτελέσματα εκπαίδευσης των δικτύων από τους W. Al-Dhabyani et al. για την αναγνώριση τριών κατηγοριών σε εικόνες υπερήχων μαστού [18].....	48
Πίνακας 4: Δείκτες αξιολόγησης κάθε μοντέλου στα δεδομένα δοκιμής.....	51
Πίνακας 5: Χρόνος εκπαίδευσης και χρόνος πρόβλεψης κάθε μοντέλου	52

Κατάλογος Εικόνων

Εικόνα 1: Τα διάφορα ενθλακωμένα επίπεδα της τεχνητής νοημοσύνης	15
Εικόνα 2: Δομή Βιολογικού Νευρώνα.....	19
Εικόνα 3: Δομή τεχνητού Νευρώνα [4]	20
Εικόνα 4: Γραμμική συνάρτηση ενεργοποίησης	21
Εικόνα 5: Σιγμοειδής συνάρτηση ενεργοποίησης	22
Εικόνα 6: Συνάρτηση ενεργοποίησης ReLU	22
Εικόνα 7: Συνάρτηση ενεργοποίησης Softmax	23
Εικόνα 8: Μεγάλος και μικρός ρυθμός μάθησης για την εύρεση ελαχίστου συνάρτηση κόστους.....	24
Εικόνα 9: Διαδικασία ενημέρωσης βαρών δικτύου.....	25
Εικόνα 10: FeedForward Net ενός επιπέδου [4].....	26
Εικόνα 11: Fully connected FeedForward Net πολλών επιπέδων [4]	27
Εικόνα 12: Αναδρομικό δίκτυο με κρυφούς νευρώνες [4]	27
Εικόνα 13: Διαδικασία μάθησης συνελικτικού νευρωνικού δικτύου	28
Εικόνα 14: Παράδειγμα συνέλιξης πινάκων για την ανίχνευση ακρών	29
Εικόνα 15: Φίλτρο 3x3 με τα επιμέρους στοιχεία του ως παράμετροι ενός δικτύου.....	29
Εικόνα 16: Διαδικασίες υποδειγματοληψίας Max Pooling και Average Pooling [8]	30
Εικόνα 17: Πλήρως συνδεδεμένα επίπεδα συνελικτικού νευρωνικού δικτύου.....	31
Εικόνα 18: Μεταφορά μάθησης από το σύνολο δεδομένων ImageNet σε σύνολο ιατρικών εικόνων [9].....	32
Εικόνα 19: Αρχιτεκτονική δικτύου LeNet-5	33

Εικόνα 20: <i>Residual Block</i> [14]	34
Εικόνα 21: Αρχιτεκτονική <i>ResNet18</i> [15]	35
Εικόνα 22: Απόδοση και σύνολο παραμέτρων <i>NASNet</i> σε σχέση με άλλα μοντέλα [16]	36
Εικόνα 23: Ο υπολογιστής υψηλών προδιαγραφών <i>NVIDIA DGX Workstation</i>	39
Εικόνα 24: Δείγματα εικόνων υπερήχων του μαστού από το σύνολο δεδομένων και της αντίστοιχης μάσκα τους [17]	42
Εικόνα 25: Οι διάφοροι τύποι παραμόρφωσης για την διεύρυνση του συνόλου δεδομένων των υπερήχων	44
Εικόνα 26: Αριθμός των εικόνων πριν και μετά την αύξηση του συνόλου δεδομένων χρησιμοποιώντας τεχνικές παραδοσιακής επαύξησης δεδομένων για την εξισορρόπηση αυτού.	45
Εικόνα 27: Οπτικοποίηση της μεταβολής των τιμών <i>accuracy</i> και <i>loss</i> για τα δεδομένα εκπαίδευσης και τα δεδομένα επαλήθευσης για τα δίκτυα <i>VGG16</i> , <i>ResNet</i> και <i>NASNet</i> (10 epochs)	50
Εικόνα 28: Πίνακες Σύγκρισης των Δικτύων <i>VGG16</i> , <i>ResNet</i> και <i>NASNet</i> στα δεδομένα δοκιμής	51
Εικόνα 29: Κύκλος αιτήματος και απάντησης μεταξύ μιας διαδικτυακής εφαρμογής και ενός διακομιστή	54
Εικόνα 30: Μπλοκ διάγραμμα των <i>containers</i> της εφαρμογής	58
Εικόνα 31: Σελίδα παρουσίασης της εφαρμογής σε υπολογιστή και κινητό	59
Εικόνα 32: <i>Input Box</i> με το οποίο ο χρήστης επιλέγει μια εικόνα στην σελίδας εφαρμογής	59
Εικόνα 33: Εμφάνιση εικόνας στον χρήστη και ενεργοποίηση κουμπιού “ <i>Predict</i> ”	60
Εικόνα 34: Εμφάνιση πρόβλεψης μοντέλου με συνοπτική αναφορά και αναλυτικό γράφημα	60

Αλφαβητικό Ευρετήριο

Ακρωνύμιο	Περιγραφή
TN	Τεχνητή Νοημοσύνη
ΔΕ	Διπλωματική Εργασία
ΥΟ	Υπολογιστική Όραση
ΣΝΔ	Συνελκτικό Νευρωνικό Δίκτυο
ΝΔ	Νευρωνικό Δίκτυο
ΣΕ	Συνάρτηση Ενεργοποίησης
CNN	Convolutional Neural Network
GPU	Graphical Processing Unit
CPU	Central Processing Unit
JSON	JavaScript Object Notation
HTML	Hyper Text Markup Language
API	Application Programming Interface
CSS	Cascading Style Sheets
HTTPS	Hypertext Transfer Protocol Secure
SSL	Secure Sockets Layer
UI	User Interface

Εισαγωγή

Η υπεροχή της τεχνητής νοημοσύνης (TN) σε κάθε πτυχή της καθημερινότητάς μας είναι πλέον αδιαμφισβήτητη. Η συνεχής πρόοδος στον τομέα της μηχανικής μάθησης έχει θέσει νέες και ισχυρές βάσεις στην ανάπτυξη της σχέσης της ανθρωπότητας με την τεχνολογία, επιλύοντας ποικίλα προβλήματα όχι μόνο στην καθημερινή ζωή, αλλά και σε πεδία μείζονος ενδιαφέροντος, όπως η ιατρική περίθαλψη. Οι νέες θεωρητικές και πρακτικές μέθοδοι που έχουν εισαχθεί στην επεξεργασία και ανάλυση εικόνων, αλλά και δεδομένων, σε συνδυασμό με την άνευ προηγουμένου ανάπτυξη των υπολογιστικών πόρων, όπως οι επεξεργαστές και οι κάρτες γραφικών, έχουν επιφέρει κυρίως θετικό αντίκτυπο στην κοινωνία και στον τομέα της βιοϊατρικής τεχνολογίας.

Η πρόβλεψη και κατηγοριοποίηση ασθενειών, όπως οι όγκοι, συνήθως απαιτούν ειδική εκτίμηση από γιατρούς ειδικευμένους σε καρκίνους. Σε καθημερινή βάση, νέα περιστατικά πιθανών καρκίνων του μαστού καταφθάνουν σε όλα τα μεγάλα ογκολογικά νοσοκομεία της χώρας, όπου πολλοί εξ' αυτών είναι καλοήθεις, ενώ άλλοι είναι κακοήθεις. Η έγκαιρη διάγνωση ενός τέτοιου καρκίνου είναι ζωτικής σημασίας, καθώς εάν ανιχνευθεί σε πρώιμο στάδιο, υπάρχει μεγάλη πιθανότητα να αντιμετωπιστεί και να θεραπευτεί πλήρως. Σύμφωνα με έρευνες, στην Ελλάδα το ποσοστό των περιστατικών καρκίνου του μαστού που διαγιγνώσκεται σε πρώιμο στάδιο είναι πολύ χαμηλότερο από το μέσο ποσοστό στην Ευρώπη, το οποίο είναι κοντά στο 60%¹.

Για την αντιμετώπιση αυτού του προβλήματος, απαιτούνται εργαλεία TN που μπορούν να παρέχουν μια δεύτερη γνώμη σε έναν γιατρό που τα χρησιμοποιεί ή ακόμα και να λειτουργούν αυτόνομα χωρίς την ανάγκη εξειδικευμένου προσωπικού. Τα μοντέλα που παρέχουν μια πρόβλεψη ακρίβειας σε ένα πρόβλημα όπως η κατηγοριοποίηση όγκων σε καλοήθεις και κακοήθεις δίνουν τη δυνατότητα σε ένα μεγαλύτερο πλήθος γυναικών να διαγνώσουν εγκαίρως μια κακοήθεια και να αυξήσουν τις πιθανότητες αντιμετώπισης του πιο συχνού καρκίνου στο γυναικείο φύλο.

Στην παρούσα διπλωματική εργασία (ΔΕ), παρουσιάζεται ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής που αποσκοπεί στην ακριβή αντιμετώπιση της ανάγκης αυτής. Από τη δημιουργία ενός μοντέλου για ακριβείς προβλέψεις σε εικόνες υπερήχων καλοήθων και κακοήθων όγκων, μέχρι την ανάπτυξη της διαδικτυακής εφαρμογής με κεντρικό προσανατολισμό τον χρήστη, χρησιμοποιώντας πολλές νέες τεχνολογίες, σκοπός μας είναι να συμβάλουμε στις προσπάθειες προτεραιοποίησης περιστατικών καρκίνου του μαστού στις γυναίκες αλλά και στην περαιτέρω ενσωμάτωση της TN στον τομέα της ιατρικής περίθαλψης.

¹ <https://www.anti-cancer.gr/pages/1> [Accessed: 12 July 2024]

Αντικείμενο της διπλωματικής εργασίας

Το κύριο θέμα και αντικείμενο της παρούσας ΔΕ είναι, αρχικά, η ανάπτυξη ενός μοντέλου αναγνώρισης εικόνας με υψηλή ακρίβεια για την ορθή κατηγοριοποίηση όγκων ως καλοήθεις ή κακοήθεις. Η αξιολόγηση του μοντέλου θα πραγματοποιηθεί μέσω ενός συνόλου δεδομένων αξιολόγησης, το οποίο θα διαχωριστεί από τα υπάρχοντα δεδομένα. Σε δεύτερη φάση, θα αναπτυχθεί ένας διακομιστής που θα παρέχει το μοντέλο μέσω μιας ιστοσελίδας, όπου οι χρήστες θα μπορούν να μεταφορτώνουν εικόνες. Οι χρήστες μέσα σε ένα διάστημα λίγων δευτερολέπτων, θα λαμβάνουν μια πρόβλεψη-απάντηση από το μοντέλο σχετικά με τη συγκεκριμένη εικόνα μέσα σε ένα εύρωστο και ευχάριστα διαμορφωμένο διαδικτυακό περιβάλλον.

Το συγκεκριμένο θέμα αποτελεί πηγή ενδιαφέροντος και σημασίας για πολλούς λόγους. Καταρχάς το θέμα αγγίζει τον ευρέως διαδεδομένο χώρο της TN, ο οποίος έχει αποκτήσει σημαντική θέση και προβολή στον παγκόσμιο χώρο την τελευταία διετία. Αυτό καθιστά το θέμα εξαιρετικά επίκαιρο, καθώς ολοένα και περισσότεροι άνθρωποι εμπλέκονται στη μελέτη και την εξερεύνηση μοντέλων και εφαρμογών που σχετίζονται με τη βαθιά μάθηση και την υπολογιστική νοημοσύνη είτε επίτηδες είτε ακούσια. Πέραν αυτού, η εργασία αντιμετωπίζει μια πάντα επίκαιρη και σημαντική ανάγκη στον τομέα της υγείας. Συγκεκριμένα, αποσκοπεί στην ορθή διάγνωση καρκίνου, καθώς και στην αποσυμπίεση του φόρτου εργασίας του ήδη επιβαρυσμένου ιατρικού προσωπικού. Η έγκαιρη διάγνωση των όγκων αποτελεί ένα κεντρικό μέρος της προσπάθειας για τη βελτίωση της ποιότητας ζωής των γυναικών σε όλο τον κόσμο και επομένως, η αντιμετώπιση αυτής της ανάγκης μέσω της TN αναδεικνύεται ως υψίστης σημασίας.

Σκοπός και στόχοι

Το ερευνητικό πλαίσιο και ο βασικός στόχος της εργασίας επικεντρώνονται στη δημιουργία ενός εργαλείου που θα παρέχει στους ιατρούς μια αξιόπιστη πρόβλεψη, χρησιμοποιώντας το είτε σε συνδυασμό με τη δική τους κρίση, είτε ως αυτόνομο εργαλείο. Η διαδικτυακή εφαρμογή στοχεύει στη διευκόλυνση της πρόσβασης του μοντέλου πρόβλεψης από τον καθένα, προσφέροντας μια απλή λύση χωρίς εμπλοκή περίπλοκων συστημάτων. Η έρευνα συνολικά επικεντρώνεται στο να συμβάλει στην έγκαιρη διάγνωση περιπτώσεων κακοηθών όγκων και στην προτεραιοποίηση των αντίστοιχων περιστατικών, είτε λειτουργώντας αυτόνομα, είτε σε συνεργασία με το ιατρικό προσωπικό.

Μεθοδολογία

Για τη μεθοδολογία της διπλωματικής, αρχικά πραγματοποιήθηκε μελέτη της βιβλιογραφίας για την TN και τα συνελικτικά νευρωνικά δίκτυα (ΣΝΔ). Στη συνέχεια, σχεδιάστηκε η αρχιτεκτονική της εφαρμογής και καθορίστηκαν τα βήματα για την εκτέλεση του έργου εντός του προκαθορισμένου χρονικού πλαισίου. Ακολούθησε η προεπεξεργασία και ο μετασχηματισμός των δεδομένων με τη χρήση βιβλιοθηκών επεξεργασίας εικόνας. Εκπαιδεύτηκαν προεκπαιδευμένοι αλγόριθμοι για τη δυαδική κατηγοριοποίηση εικόνων υπερήχων όγκων του μαστού. Έπειτα, πραγματοποιήθηκε ανάλυση των αποτελεσμάτων για την εκτίμηση της απόδοσης των νευρωνικών δικτύων (ΝΔ) και την επιλογή του βέλτιστου αλγορίθμου. Τέλος, υλοποιήθηκε ο διακομιστής της διαδικτυακής εφαρμογής και στη συνέχεια η εφαρμογή η ίδια, η οποία τέθηκε σε λειτουργία και είναι διαθέσιμη σε πραγματικό περιβάλλον. Ο κώδικας επίσης είναι διαθέσιμος στο GitHub, προσφέροντας εύκολη πρόσβαση σε όλους τους ενδιαφερόμενους για περαιτέρω ανάπτυξη και έρευνα.

Καινοτομία

Η καινοτομία της παρούσας ΔΕ αναδεικνύεται μέσα από τον συνδυασμό δύο διακριτών τεχνολογιών που αρχικά φαίνονται ασύνδετες μεταξύ τους. Στην εργασία, η αξιοποίηση της TN και των πιο αξιόπιστων μοντέλων ΣΝΔ για την αναγνώριση εικόνων και προτύπων συνδυάζεται με την ανάπτυξη διαδικτυακών εφαρμογών. Το αποτέλεσμα αυτού του συνδυασμού είναι μια εφαρμογή που δίνει έμφαση στον σχεδιασμό με επίκεντρο τον χρήστη, η οποία χρησιμοποιεί έναν πρόσφατα ανεπτυγμένο αλγόριθμο για τη δυαδική κατηγοριοποίηση όγκων του μαστού, προσφέροντας παράλληλα και μια αυξημένη ακρίβεια και αποτελεσματικότητα στη διάγνωση. Επιπλέον, η καινοτομία της εφαρμογής ενισχύεται από το γεγονός ότι ο κώδικας είναι ανοιχτός και προσβάσιμος από όλους (open source), επιτρέποντας την ελεύθερη χρήση και βελτίωσή της από την επιστημονική κοινότητα και το ιατρικό προσωπικό. Αυτός ο συνδυασμός πρωτοποριακών προσεγγίσεων δημιουργεί ένα εργαλείο που όχι μόνο αναδεικνύει μια καινοτόμο μέθοδο στη διάγνωση περιστατικών καρκίνου του μαστού, αλλά ενισχύει το έργο των γιατρών και στέκεται επάξια στο πλευρό του ιατρικού προσωπικού που ασχολείται με την αντιμετώπιση περιστατικών καρκίνου του μαστού

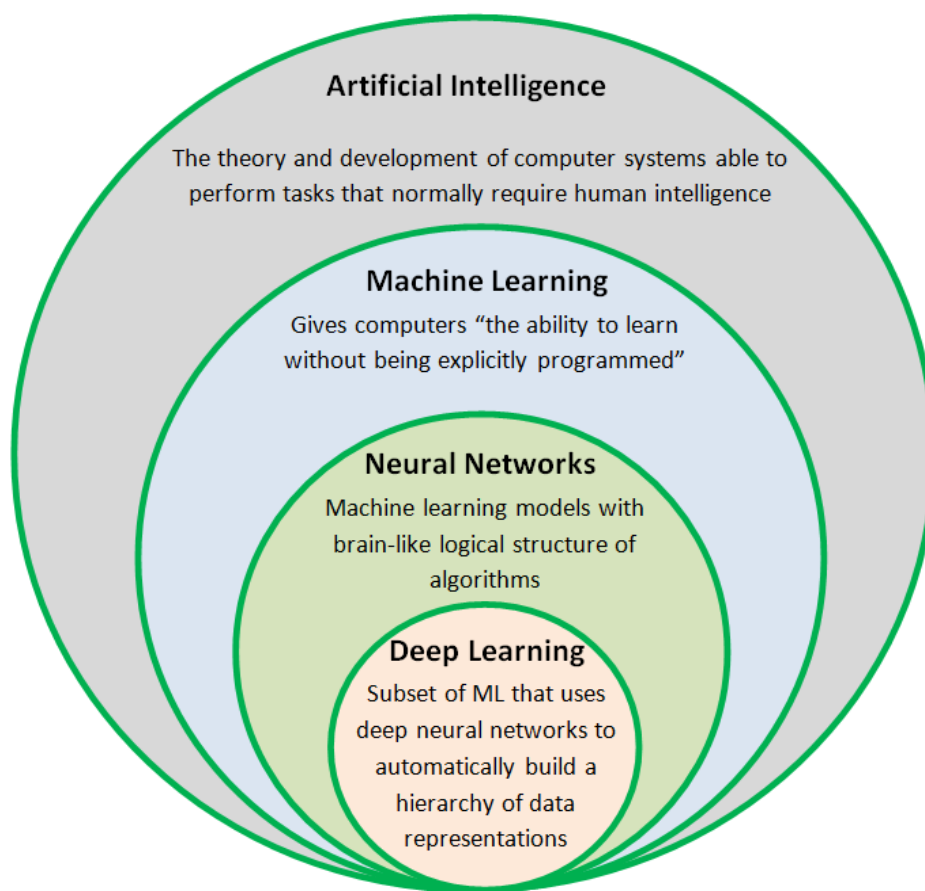
Δομή Διπλωματικής Εργασίας

Η εργασία αποτελείται από τρία βασικά κεφάλαια, καθένα από τα οποία συμβάλλει στη συνολική κατανόηση και υλοποίηση του έργου. Στο πρώτο κεφάλαιο, το οποίο είναι χωρισμένο σε τέσσερα υποκεφάλαια, αναλύεται το θεωρητικό υπόβαθρο που απαιτείται για την κατανόηση των εννοιών που χρησιμοποιούνται γύρω από την TN. Αρχικά, γίνεται μια εισαγωγή στους γενικούς όρους και τις βασικές λειτουργίες που πρέπει να είναι γνωστές, εμβαθύνοντας σταδιακά σε πιο ειδικές έννοιες και μοντέλα, τα οποία είναι απαραίτητα για την υλοποίηση του μοντέλου της εφαρμογής. Ιδιαίτερη έμφαση δίνεται στις θεμελιώδεις αρχές της TN, στα ΝΔ, καθώς και στα σύγχρονα μοντέλα βαθιάς μάθησης που χρησιμοποιούνται ευρέως από ερευνητές του κλάδου. Το δεύτερο κεφάλαιο επικεντρώνεται στην ανάλυση και την προεπεξεργασία των δεδομένων, έτσι ώστε αυτά να μπορούν να τροφοδοτηθούν στα ΝΔ. Αρχικά, παρουσιάζονται οι τεχνικές επαύξησης των δεδομένων, οι οποίες βελτιώνουν σημαντικά την ακρίβεια των προβλέψεων ενός μοντέλου ενώ στη συνέχεια, εκπαιδεύονται τρία ΣΝΔ χρησιμοποιώντας τα ίδια δεδομένα. Αφού ολοκληρωθεί η εκπαίδευση, συγκρίνονται διάφοροι δείκτες ακρίβειας για να αξιολογηθεί και να επιλεγεί το καλύτερο συνολικά μοντέλο. Στο τρίτο κεφάλαιο, περιγράφεται αναλυτικά η διαδικασία δημιουργίας της διαδικτυακής εφαρμογής και του διακομιστή που θα την φιλοξενήσει. Από τη δημιουργία του γραφικού περιβάλλοντος όλων των σελίδων με τη βοήθεια της βιβλιοθήκης React, μέχρι τη χρήση τεχνολογιών νεφοϋπολογιστικής για τη φιλοξενία της εφαρμογής, κάθε βήμα παρουσιάζεται με λεπτομέρεια. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση του τελικού αποτελέσματος το οποίο δικαιολογεί τη χρήση όλων αυτών των προηγμένων και περίπλοκων τεχνολογιών. Η συνδυαστική χρήση αυτών των τεχνολογιών επέτρεψε τη δημιουργία μιας καινοτόμου και αποδοτικής εφαρμογής, η οποία είναι σε θέση να σταθεί επάξια δίπλα σε υπάρχοντα επαγγελματικά λογισμικά, προσφέροντας αξιόπιστες υπηρεσίες στους ιατρούς και βελτιώνοντας τη συνολική εμπειρία χρήστη. Τέλος, παρουσιάζονται τα συμπεράσματα που προέκυψαν από την υλοποίηση αυτής της εφαρμογής, καθώς και πιθανές μελλοντικές βελτιώσεις που μπορούν να πραγματοποιηθούν σε κάθε επίπεδο.

ΚΕΦΑΛΑΙΟ 1^ο : Τεχνητή Νοημοσύνη και Νευρωνικά Δίκτυα

1.1 Τεχνητή Νοημοσύνη

Η αυξημένη προσοχή και το έντονο ενδιαφέρον που δίνεται τα τελευταία χρόνια στον τομέα της ΤΝ από ειδικούς και μη, την έχουν καταστήσει έναν από τους πλέον δυναμικούς και επεκτεινόμενους τομείς στην επιστημονική και τεχνολογική κοινότητα. Ο ενθουσιασμός αυτός και η αφοσίωση που επιδεικνύουν οι επιστήμονες ανά τον κόσμο αποτελούν ένδειξη της σημαντικότητάς της στη σύγχρονη κοινωνία. Με τη διάδοσή της σε πολλά καθημερινά εργαλεία και την ενσωμάτωσή της σε πολλούς τομείς της ζωής μας, η ΤΝ έχει εμπνεύσει ένα κύμα αυξημένης έρευνας και επιστημονικής δραστηριότητας. Αποτελώντας μάλιστα έναν από τους πιο νέους ερευνητικούς τομείς [1], η ΤΝ εξελίσσεται με έναν ρυθμό που συχνά εντυπωσιάζει, καθιστώντας σαφές το ότι ήρθε για να μείνει.



Εικόνα 1: Τα διάφορα ενθυλακωμένα επίπεδα της τεχνητής νοημοσύνης

1.1.1 Η εξέλιξη της τεχνητής νοημοσύνης

Από την αρχαιότητα μέχρι τον 20ό αιώνα, η ανθρωπότητα εξέταζε την έννοια της νοημοσύνης. Από τον Αριστοτέλη που πρώτος ανέδειξε τη σημασία της λογικής και της διαίσθησης, η ανθρώπινη επιθυμία να μιμηθεί τη νοημοσύνη μέσω μηχανών άρχισε να παίρνει σάρκα και οστά. Αυτή η επιθυμία κορυφώθηκε το 1943, όταν ο νευροφυσιολόγος Warren McCulloch με τον Walter Pitts δημοσίευσαν την πρώτη εργασία που προσεγγίζει την ΤΝ, μοντελοποιώντας τον ανθρώπινο εγκέφαλο με βάση τη λογική των νευρώνων. Οι έρευνες αυτών των δύο σε συνδυασμό με την υλοποίηση και την προσομοίωση ενός μοντέλου τεχνητών νευρώνων μερικά χρόνια αργότερα στο Πανεπιστήμιο του Princeton από τους Minsky και Edmonds, ανοίγουν το δρόμο για τον Frank Rosenblatt το 1957, ο οποίος δημιουργεί το perceptron, μια πρωτοποριακή ιδέα που επιτρέπει στις μηχανές να μαθαίνουν να κατηγοριοποιούν δεδομένα, ανοίγοντας νέους ορίζοντες στην ΤΝ. Μετά από δεκαετίες ανάπτυξης και συνεχούς έρευνας, και αφού η ΤΝ έχει θεσπιστεί ως ένα νέο επιστημονικό πεδίο από τον John McCarthy, δημιουργούνται συστήματα επεξεργασίας γλώσσας, αναγνώρισης εικόνας καθώς και αυτοματοποιημένης εκμάθησης. Φτάνοντας στον 21ο αιώνα και στη δεκαετία που διανύουμε, η οποία αποτελεί μια δεκαετία ορόσημο για την ΤΝ, πανεπιστήμια αλλά και πολλές εταιρείες κάνουν αγώνα δρόμου για το ποια θα αναπτύξει το πιο καινοτόμο και επικερδές σύστημα, είτε αυτό είναι ένα μοντέλο Large Language όπως το chatGPT της OpenAI, είτε αυτό είναι ένα αυτόματο εργαλείο ανάπτυξης λογισμικού – προγραμματιστής όπως το πρόσφατο DevinAI της Cognition.

Όσον αφορά τον ορισμό της ΤΝ, ο επικρατέστερος ορισμός που έχει δοθεί προέρχεται από τους Barr και Feigenbaum στο βιβλίο με τίτλο “The handbook of Artificial Intelligence”. Αυτοί ορίζουν την ΤΝ ως “τον τομέα που ασχολείται με τη σχεδίαση ευφών υπολογιστικών συστημάτων, δηλαδή συστημάτων που επιδεικνύουν χαρακτηριστικά που σχετίζονται με τη νοημοσύνη στην ανθρώπινη συμπεριφορά”. Πολλές φορές ως έννοια σχετίζεται με την ικανότητα των ανθρώπων να επιλύουν προβλήματα και να μαθαίνουν από την εμπειρία τους και τα λάθη τους, καθώς και να εξάγουν συμπεράσματα με βάση τις ήδη υπάρχουσες γνώσεις τους [2]. Η ΤΝ ως σύνολο αποτελεί σημείο τομής επιστημών όπως η φιλοσοφία, τα μαθηματικά, η μηχανική, η νευρολογία, η ψυχολογία και φυσικά η πληροφορική [1].

1.1.2 Υπολογιστική Νοημοσύνη

Η υπολογιστική νοημοσύνη αποτελεί το μεγαλύτερο και το σημαντικότερο κομμάτι της ΤΝ. Αντιπροσωπεύει τη σύνθεση και την εφαρμογή αλγορίθμων και μεθόδων που επιτρέπουν στους υπολογιστές να μιμούνται μηχανισμούς που παρατηρούνται στη φύση από τα έμβια και κυρίως ευφυή όντα, με στόχο να επιλύσει πολύπλοκα προβλήματα που απαιτούν ανθρώπινη ευφυΐα, όπως η αναγνώριση προτύπων και η λήψη αποφάσεων, προβλήματα τα οποία δεν θα μπορούσαν να λυθούν με χρήση συμβατικών μεθόδων [3]. Βασίζεται κυρίως στη μάθηση μέσω επαναληπτικών διαδικασιών για τη ρύθμιση διαφόρων παραμέτρων και στα εμπειρικά δεδομένα, ενώ η επίλυση προβλημάτων γίνεται με τρεις βασικές μεθόδους:

- Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)

Τα τεχνητά νευρωνικά δίκτυα είναι πολύ ισχυρά υπολογιστικά συστήματα που αντλούν έμπνευση από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου. Αποτελούνται από διάφορα επίπεδα νευρώνων και χρησιμοποιούνται για προβλήματα αναγνώρισης προτύπων, πρόβλεψης, αναγνώρισης φωνής, κ.α. Ο τρόπος λειτουργίας τους θα αναλυθεί εκτενέστερα σε επόμενα κεφάλαια.

- Συστήματα ασαφούς λογικής (Fuzzy Logic Systems)

Τα συστήματα ασαφούς λογικής χρησιμοποιούνται για την αντιμετώπιση της αβεβαιότητας και της ασάφειας σε δεδομένα και πληροφορίες. Αντί να εργάζονται με δυαδικές τιμές (όπως παραδείγματος χάριν αληθής/ψευδής), τα συστήματα αυτά λειτουργούν με βαθμούς αλήθειας μέσω ασαφών συνόλων και κανόνων που ορίζει ο προγραμματιστής. Η ασαφής λογική μπορεί να προσφέρει λύσεις σε προβλήματα που δεν μπορούν να περιγραφούν με ακρίβεια μαθηματικών εξισώσεων ή μοντέλων, αλλά απαιτούν την ανθρώπινη εμπειρία και κρίση [3].

- Εξελικτική Υπολογιστική (Evolutionary Computation)

Η εξελικτική υπολογιστική αποτελεί ένα πεδίο της ΤΝ που εστιάζει στη χρήση εξελικτικών αλγορίθμων για την επίλυση προβλημάτων. Αυτοί οι αλγόριθμοι εμπνέονται από τη φυσική εξέλιξη και την επιλογή του πιο προσαρμοστικού. Η εξελικτική υπολογιστική εφαρμόζεται σε πολλούς τομείς, όπως αυτόν της μαθηματικής βελτιστοποίησης και της αναγνώρισης προτύπων.

1.1.3 Μηχανική Μάθηση

Ένα από τα παρακλάδια της τεχνητής και της υπολογιστικής νοημοσύνης που έχει επηρεαστεί θετικά από την διαρκή εξέλιξη και άνοδο των υπολογιστικών συστημάτων και αλγορίθμων, είναι αυτό της μηχανικής μάθησης. Ο ορισμός που διατυπώθηκε το 1959 από τον Arthur Samuel περιγράφει τη μηχανική μάθηση ως "ένα πεδίο μελέτης που παρέχει στους υπολογιστές τη δυνατότητα να μαθαίνουν χωρίς να έχουν ρητά προγραμματιστεί". Το όνομα της μηχανικής μάθησης είναι στενά συνυφασμένο με την υπολογιστική στατιστική και την μαθηματική βελτιστοποίηση και βασίζεται στην ιδέα ότι οι υπολογιστές μπορούν να μαθαίνουν από τα δεδομένα που συλλέγουν με σκοπό να αναγνωρίζουν μοτίβα και να λαμβάνουν αποφάσεις. Στη σημερινή εποχή, η μηχανική μάθηση αποτελεί βασικό πυλώνα για την ανάπτυξη και την αυτοματοποίηση διαδικασιών σε ολόένα και περισσότερους οργανισμούς και επιχειρήσεις. Η επίλυση προβλημάτων που επιτυγχάνεται για την αυτοματοποίηση αυτών των διεργασιών περιλαμβάνει την προσέγγιση συναρτήσεων, την ταξινόμηση και την συσταδοποίηση ενώ μπορεί να επιτευχθεί μέσω τριών βασικών κατηγοριών μηχανικής μάθησης:

- Επιβλεπόμενη Μάθηση (Supervised Learning)

Στην επιβλεπόμενη μάθηση, το μοντέλο εκπαιδεύεται χρησιμοποιώντας ένα σύνολο δεδομένων που περιλαμβάνει τις εισόδους, αλλά και τις αντίστοιχες ετικέτες ή κλάσεις που αναμένεται το μοντέλο να προβλέψει. Στόχος ενός μοντέλου επιβλεπόμενης μάθησης είναι να προβλεφθούν οι ετικέτες εξόδου για νέα δεδομένα, βασιζόμενο στην εκπαίδευση που έχει ήδη πραγματοποιηθεί.

- Μη επιβλεπόμενη μάθηση (Unsupervised Learning)

Η μη επιβλεπόμενη μάθηση αναφέρεται σε ένα είδος μάθησης όπου το μοντέλο εκπαιδεύεται χωρίς να γνωρίζει τις πραγματικές ετικέτες των εξόδων. Προκειμένου να κάνει μια πρόβλεψη, το μοντέλο αναλύει τα δεδομένα και εξάγει πληροφορίες, όπως μοτίβα ή δομές, δηλαδή χρησιμοποιεί ένα ανεξάρτητο μέτρο συγκρίσεων για τη βελτιστοποίηση των ελεύθερων παραμέτρων του [4].

- Ενισχυτική μάθηση (Reinforcement Learning)

Στην ενισχυτική μάθηση, η απόκτηση γνώσης σχετίζεται με τη συνεχή αλληλεπίδραση του μοντέλου με το περιβάλλον, με στόχο την ελαχιστοποίηση ενός βαθμωτού δείκτη απόδοσης. Κατά τη διαδικασία εκπαίδευσης, το μοντέλο λαμβάνει ανταμοιβές ή ποινές για τις ενέργειές του, με στόχο τη μείωση μιας συνολικής συνάρτησης κόστους.

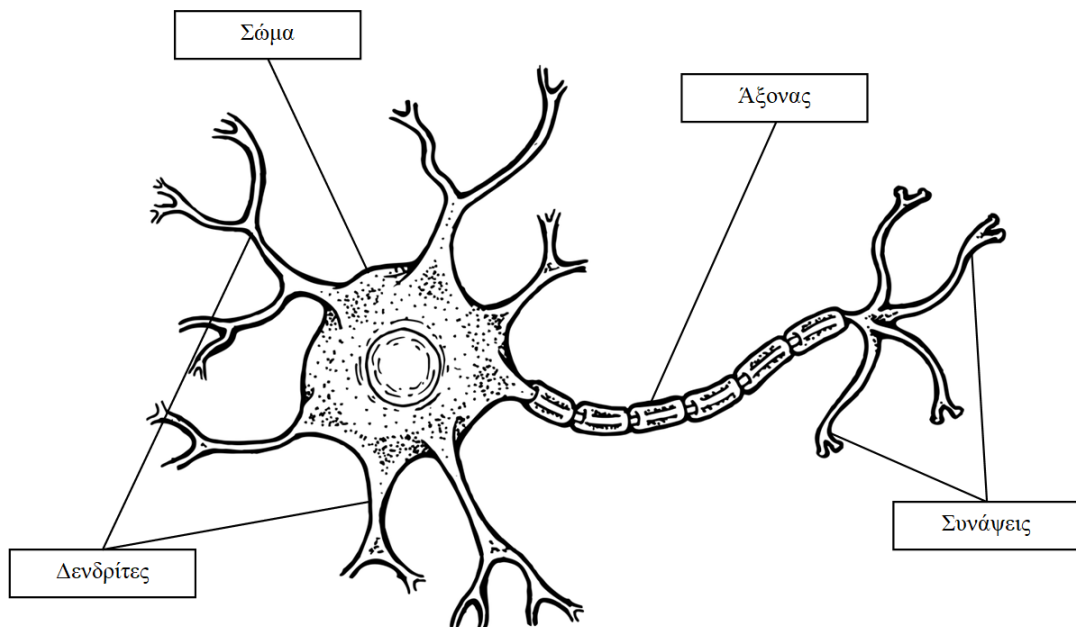
Οι διαφορετικές μορφές μάθησης που περιγράφονται, υπάγονται κατά σειρά σε δύο μεγαλύτερα σύνολα, στη μάθηση με εκπαιδευτή (επιβλεπόμενη μάθηση) και στην μάθηση χωρίς εκπαιδευτή (μη επιβλεπόμενη και ενισχυτική μάθηση), αντιπροσωπεύοντας τρόπους που προσομοιώνουν λίγο η πολύ την ανθρώπινη διαδικασία μάθησης [4]. Με τη συνεχή αύξηση των δεδομένων και την ευρύτερη πρόσβαση σε αυτά, καθώς και την πολυπλοκότητα στις συσχετίσεις που μπορεί να παρουσιάζουν, τα μοντέλα μηχανικής μάθησης έχουν πλέον αρχίσει να εξελίσσονται σε μοντέλα που απαιτούν σημαντικά περισσότερα επίπεδα ή μονάδες επεξεργασίας για την παραγωγή αξιόπιστων προβλέψεων, ευρέως γνωστά και ως μοντέλα βαθιάς μάθησης.

1.2 Νευρωνικά Δίκτυα

1.2.1 Τεχνητοί Νευρώνες

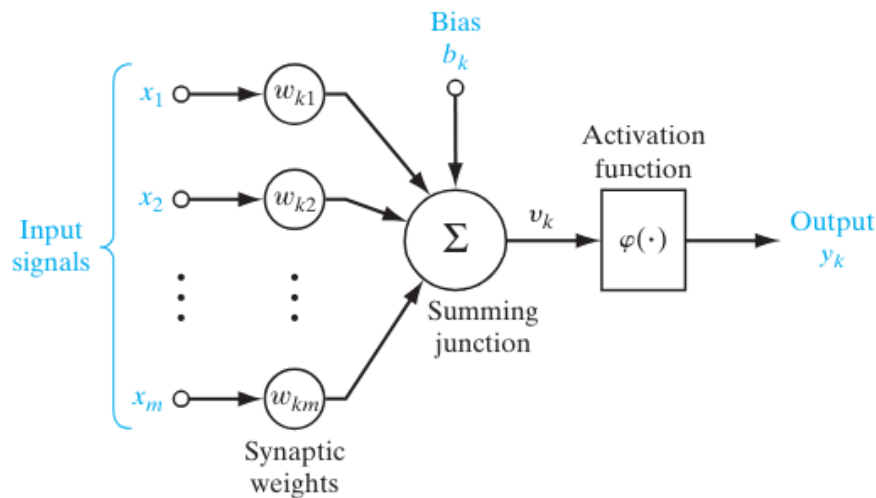
Παρά το γεγονός ότι ο ανθρώπινος εγκέφαλος εκτελεί αριθμητικές πράξεις με πολύ πιο αργό ρυθμό από ότι ένας υπολογιστής, η αποτελεσματικότητά του στην αντιμετώπιση πολύπλοκων εργασιών παραμένει εντυπωσιακή. Αυτός είναι ο λόγος που για πολλά χρόνια το κύριο μέλημα των ερευνητών και των επιστημόνων που ασχολούταν με την ΤΝ ήταν να απομονώσουν και να προσομοιώσουν τους νευρώνες του ανθρώπινου εγκεφάλου όσο το δυνατόν καλύτερα. Τα τεχνητά ΝΔ αποτελούν πλέον την πιο διαδεδομένη και χρησιμοποιούμενη μέθοδο επίλυσης προβλημάτων ΤΝ και συνδέονται άμεσα με τη μηχανική και τη βαθιά μάθηση. Η στοχευμένη διαφήμιση στο διαδίκτυο βάσει των προτιμήσεων του χρήστη και πιο πολύπλοκες εφαρμογές όπως η αναγνώριση εικόνων ή η αναγνώριση ήχου αποτελούν μόνο μερικά από τα προβλήματα που μπορούν να επιλύσουν τα ΝΔ κάθε είδους.

Βασική έμπνευση στη δημιουργία τεχνητών ΝΔ αποτελεί η ίδια η δομή των νευρώνων του εγκεφάλου. Η διαδικασία μάθησης σε έναν βιολογικό οργανισμό και συγκεκριμένα στον εγκέφαλο γίνεται μέσω της μετάδοσης μιας πληροφορίας από τον έναν νευρώνα στον άλλον υπό τη μορφή ηλεκτρικού σήματος. Αυτή η μετάδοση του ηλεκτρικού σήματος πραγματοποιείται μέσω κάποιων βασικών τμημάτων του νευρώνα. Αρχικά από τις συνάψεις του προηγούμενου νευρώνα και μέσω μιας χημικής διαδικασίας που τροποποιεί τη συχνότητα του εν λόγω σήματος, οι νευρικοί δενδρίτες προσλαμβάνουν το σήμα, ενώ στη συνέχεια, το προωθούν στο σώμα που είναι υπεύθυνο για την συνολική άθροιση όλων των σημάτων που λαμβάνει από γειτονικούς δενδρίτες. Έπειτα, το σήμα που παράγεται μεταφέρεται μέσω του άξονα στις συνάψεις, οι οποίες τροφοδοτούν, τελικά, τους επόμενους νευρώνες που συνδέονται με αυτές [5]. Η δομή ενός τέτοιου βιολογικού νευρώνα, όπως περιγράφηκε, παρουσιάζεται στην Εικόνα 2.



Εικόνα 2: Δομή Βιολογικού Νευρώνα

Ένας τεχνητός νευρώνας, όπως και ένας βιολογικός, αντιπροσωπεύει μια μονάδα επεξεργασίας πληροφοριών ουσιαστική για τη λειτουργία ενός ΝΔ. Ακολουθώντας τη δομή των πραγματικών νευρώνων, τα σήματα εισόδου (Input signals) που εισέρχονται συνδέονται με τις συνάψεις των νευρώνων, καθεμία εκ των οποίων φέρει ένα χαρακτηριστικό συναπτικό βάρος (Synaptic weight). Τα σήματα αυτά πολλαπλασιάζονται με τα αντίστοιχα συναπτικά βάρη και στη συνέχεια αθροίζονται μαζί με μια εξωτερικά εφαρμοσμένη πόλωση (bias) στον αθροιστή (Summing junction). Τέλος, η πορεία των σημάτων ολοκληρώνεται μέσω της διέλευσης τους από μια συνάρτηση ενεργοποίησης (Activation function), γνωστή και ως συνάρτηση περιορισμού, προκειμένου να διαμορφωθεί η τελική έξοδος [4]. Η Εικόνα 3 παρουσιάζει γραφικά τη δομή ενός τέτοιου μη γραμμικού μοντέλου νευρώνα όπως περιγράφηκε.



Εικόνα 3: Δομή τεχνητού Νευρώνα [4]

Η λειτουργία ενός τεχνητού νευρώνα και η Εικόνα 3 μπορεί επίσης να περιγραφεί με χρήση μαθηματικών όρων με την παρακάτω εξίσωση:

$$y_k = \phi\left(\sum_{j=1}^m w_{kj}x_j + b_k\right)$$

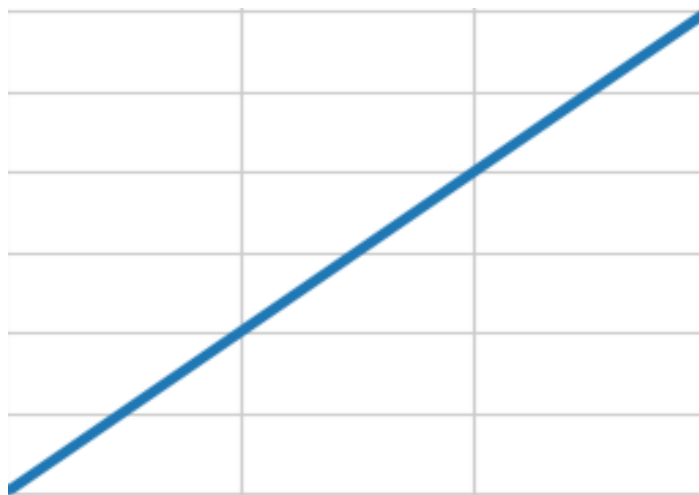
Η εξίσωση περιγράφει αρχικά τον πολλαπλασιασμό των σημάτων εισόδου x_j με τα συναπτικά βάρη των νευρώνων w_{kj} , καθώς και τη συνολική αθροιστική λειτουργία, όπου προστίθεται ένας σταθερός όρος πόλωσης b_k . Τέλος, εφαρμόζεται στο αποτέλεσμα μια συνάρτηση ενεργοποίησης (ΣΕ) ϕ , και έτσι προκύπτει η έξοδος y_k .

1.2.2 Συναρτήσεις ενεργοποίησης

Οι συναρτήσεις ενεργοποίησης αποτελούν βασικά στοιχεία και διαδραματίζουν κρίσιμο ρόλο στη διαμόρφωση ενός ΝΔ. Αποτελούν τους μηχανισμούς που δέχονται ως είσοδο το συνολικό αθροισμένο σήμα συναπτικών βαρών και σημάτων εισόδου και το μετασχηματίζουν κατάλληλα για την παραγωγή της εξόδου. Τα τελευταία χρόνια έχουν εμφανιστεί ορισμένες συναρτήσεις ενεργοποίησης που έχουν ξεχωρίσει και χρησιμοποιούνται για διαφορετικές διεργασίες, καθεμία από αυτές επιτυγχάνοντας το επιθυμητό αποτέλεσμα.

- Γραμμική ΣΕ

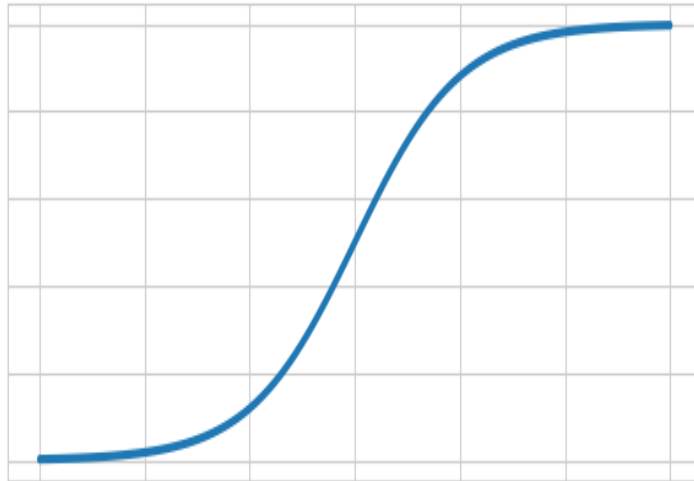
Η γραμμική ΣΕ είναι η πιο απλή μορφή ενεργοποίησης στα ΝΔ. Αν και απλή, παραμένει εξαιρετικά σημαντική και χρησιμοποιείται σε προβλήματα όπου υπάρχει γραμμική αναπαράσταση όπως παραδείγματος χάριν είναι η πρόβλεψη των τιμών του χρηματιστηρίου. Τα τελευταία χρόνια λόγω της απλότητας τους και λόγω της ανόδου της πολυπλοκότητας των διαθέσιμων δεδομένων αποφεύγεται ευρέως η χρήση των γραμμικών συναρτήσεων στις κρυφές στοιβάδες ενός ΝΔ.



Εικόνα 4: Γραμμική συνάρτηση ενεργοποίησης

- Σιγμοειδής ΣΕ

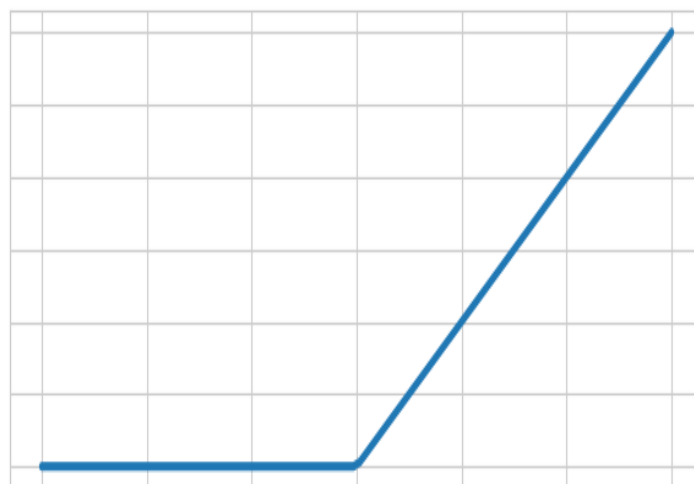
Η σιγμοειδής ΣΕ είναι η πιο διαδεδομένη μορφή ΣΕ στην κατασκευή ΝΔ. Αυτή η συνάρτηση παρουσιάζει μια πολύ κομψή ισορροπία μεταξύ γραμμικής και μη γραμμικής συμπεριφοράς και χρησιμοποιείται συχνότερα σε δίκτυα που ασχολούνται με προβλήματα δυαδικής ταξινόμησης [4], αλλά όπως και η γραμμική αποφεύγεται η χρήση τους στις κρυφές στοιβάδες ενός δικτύου καθώς ένα μοντέλο μπορεί να παρουσιάσει προβλήματα κατά την διάρκεια της εκπαίδευσης (vanishing gradients).



Εικόνα 5: Σιγμοειδής συνάρτηση ενεργοποίησης

- ΣΕ ReLU

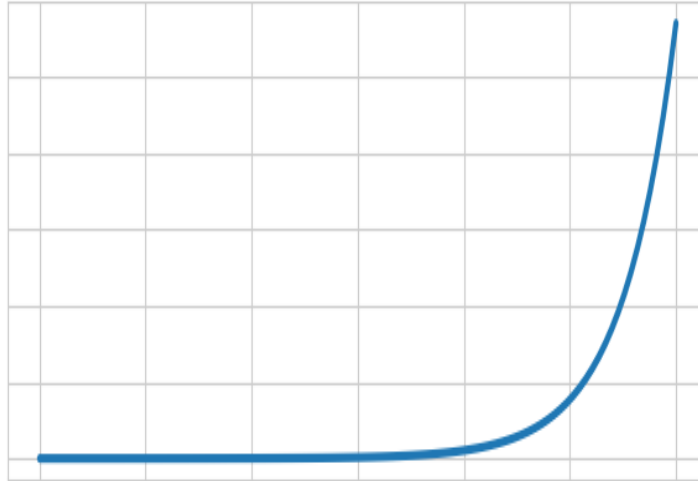
Η ΣΕ ReLU (Rectified Linear Unit) είναι μια μη γραμμική συνάρτηση που χρησιμοποιείται ευρέως στα ΝΔ και σχεδόν αποκλειστικά στις κρυφές στοιβάδες ενός δικτύου. Υπολογίζει την έξοδο ως το μέγιστο μεταξύ της εισόδου και του μηδενός [$g(z) = \max(0,z)$] και χρησιμοποιείται για την εισαγωγή μη γραμμικότητας στο δίκτυο.



Εικόνα 6: Συνάρτηση ενεργοποίησης ReLU

- ΣΕ Softmax

Η ΣΕ Softmax είναι μια μη γραμμική συνάρτηση που χρησιμοποιείται συνήθως στην τελευταία στοιβάδα ενός ΝΔ σε προβλήματα κατηγοριοποίησης πολλών κλάσεων. Παράγει μια κατανομή πιθανοτήτων πάνω στις διαφορετικές κατηγορίες που του έχει δοθεί να ταξινομήσει, με αποτέλεσμα η άθροιση των εξόδων να είναι ίση με τη μονάδα.



Εικόνα 7: Συνάρτηση ενεργοποίησης Softmax

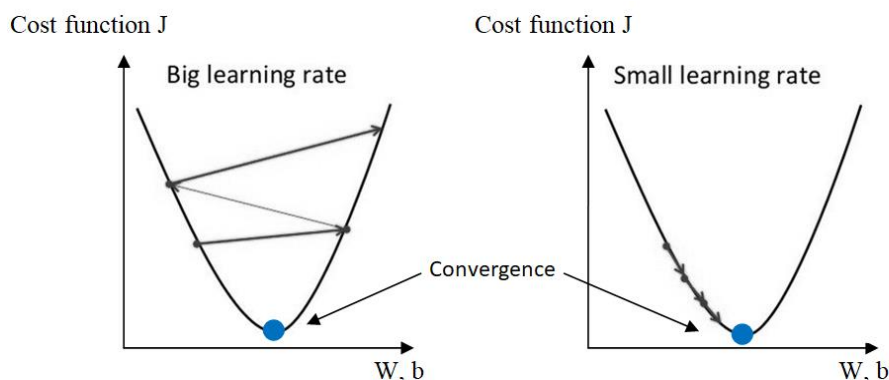
Αξίζει επίσης να γίνει μια αναφορά και στις συναρτήσεις ενεργοποίησης Leaky ReLU και tanh, οι οποίες χρησιμοποιούνται για διαφορετικούς λόγους. Η Leaky ReLU επιτρέπει μια πολύ μικρή αρνητική τιμή για αρνητικές εισόδους, αντιμετωπίζοντας έτσι το πρόβλημα της ReLU που μηδενίζει τις αρνητικές τιμές. Η συνάρτηση tanh προσφέρει μια συμμετρική ΣΕ με εύρος τιμών από -1 έως 1. Η tanh χρησιμοποιούταν πολύ περισσότερο στο παρελθόν αλλά παρά τις εναλλακτικές που έχουν αναπτυχθεί, εξακολουθεί ακόμα και σήμερα να είναι χρήσιμη σε περιπτώσεις όπου απαιτείται “κεντράρισμα” των δεδομένων γύρω από το μηδέν.

1.2.3 Υπερπαράμετροι δικτύου

Η επιλογή και η ρύθμιση των υπερπαραμέτρων ενός ΝΔ αποτελούν κρίσιμα βήματα στη διαδικασία ανάπτυξης και εκπαίδευσης μοντέλων μηχανικής και βαθιάς μάθησης. Υπερπαραμέτροι όπως ο ρυθμός εκμάθησης, οι εποχές (epochs), η ύπαρξη ή όχι της τεχνικής αποκοπής (dropout), καθώς και οι αλγόριθμοι βελτιστοποίησης που χρησιμοποιούνται από το μοντέλο, επηρεάζουν άμεσα την απόδοση των ΝΔ. Η σωστή ρύθμιση αυτών των παραμέτρων είναι απαραίτητη για την επίτευξη του βέλτιστου δυνατού αποτελέσματος, διασφαλίζοντας την αποτελεσματική εκπαίδευση και γενίκευση του μοντέλου.

- Ρυθμός μάθησης (Learning rate)

Ο ρυθμός μάθησης καθορίζει τον ρυθμό με τον οποίο προσαρμόζονται τα βάρη του μοντέλου κατά την εκπαίδευση έτσι ώστε αυτό να βρει το ελάχιστο της συνάρτησης κόστους. Χρησιμοποιώντας έναν μεγάλο ρυθμό μάθησης το μοντέλο μπορεί να οδηγηθεί σε αστάθεια (overshooting) και τελικά σε αδυναμία σύγκλισης, ενώ αντίθετα ένας πολύ μικρός ρυθμός μάθησης μπορεί να οδηγήσει σε πολύ αργή σύγκλιση ή ακόμα και σε εγκλωβισμό σε τοπικά ελάχιστα [6].



Εικόνα 8: Μεγάλος και μικρός ρυθμός μάθησης για την εύρεση ελαχίστου συνάρτησης κόστους

- Εποχές (Epochs)

Η υπερπαραμέτρος "epochs" καθορίζει τον αριθμό των επαναλήψεων που εκτελούνται κατά την εκπαίδευση του μοντέλου με το σύνολο δεδομένων εκπαίδευσης και περιγράφει την ενημέρωση των βαρών του δικτύου βάσει της υπολογιζόμενης κλίσης. Μεγαλύτερες τιμές είναι γενικά ευεργετικές για την ποιότητα του μοντέλου, αλλά μπορεί να αυξήσουν τον απαιτούμενο χρόνο εκτέλεσης. Ωστόσο, μεγαλύτεροι χρόνοι εκτέλεσης μπορεί να αυξήσουν τον κίνδυνο υπερβολικής προσαρμογής στα δεδομένα (overfitting), εάν δεν ληφθούν κάποια μέτρα προφύλαξης [6].

- Αποκοπή (Dropout)

Το dropout είναι μια τεχνική που χρησιμοποιείται για να αποτραπεί η υπερεκπαίδευση και η υπερβολική προσαρμογή στα δεδομένα εκπαίδευσης. Χρησιμοποιώντας την τεχνική αποκοπής κατά τη διάρκεια της εκπαίδευσης, επιλέγονται τυχαία κάποιοι νευρώνες (η πιθανότητα επιλογής ορίζεται από το dropout rate) και μαζί με τις αντίστοιχες συνδέσεις τους απενεργοποιούνται με αποτέλεσμα τη μείωση της πιθανότητας υπερεκπαίδευσης του μοντέλου καθώς και στην αύξηση της απόδοσής του [6].

- Αλγόριθμοι βελτιστοποίησης (Gradient descent optimization algorithms)

Η καθοδική κλίση (gradient descent) αποτελεί μια μέθοδο βελτιστοποίησης που χρησιμοποιείται για την εύρεση του ελάχιστου μιας συνάρτησης. Στο συγκεκριμένο πρόβλημα που θα αναλυθεί στη ΔΕ, η πρόβλεψη y θα λαμβάνει τιμές μόνο 0 ή 1 οπότε στόχος είναι να βρεθεί το ελάχιστο της συνάρτησης κόστους του ΝΔ [4], η οποία ορίζεται ως εξής:

$$J(w, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(f_{w,b}(x^{(i)})) + (1 - y^{(i)}) \log(1 - f_{w,b}(x^{(i)}))]$$

Στην εξίσωση αυτή, το $y^{(i)}$ αντιπροσωπεύει την πραγματική τιμή της κλάσης για το i -οστό παράδειγμα, ενώ το $f_{w,b}(x^{(i)})$ είναι η προβλεπόμενη πιθανότητα ότι το i -οστό παράδειγμα ανήκει στην κατηγορία 1. Η ελαχιστοποίηση αυτής της συνάρτησης κόστους επιτυγχάνεται με συνεχή ενημέρωση των βαρών w και b έως ότου επιτευχθεί σύγκλιση (convergence), ακριβώς όπως και στα προβλήματα γραμμικής παλινδρόμησης με την διαφορά ότι εδώ χρησιμοποιείται σιγμοειδής ΣΕ². Η παρακάτω εικόνα παρουσιάζει την υλοποίηση αυτής της διαδικασίας:

$$\left. \begin{aligned} w_j &= w_j - a \left[\frac{1}{m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)}) x_j^{(i)} \right] \\ b &= b - a \left[\frac{1}{m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)}) \right] \end{aligned} \right\} \begin{array}{l} \text{Repeat until} \\ \text{convergence} \end{array}$$

Εικόνα 9: Διαδικασία ενημέρωσης βαρών δικτύου

Η κύρια λειτουργία των αλγορίθμων βελτιστοποίησης είναι η ενημέρωση των βαρών του ΝΔ μέσω της διαδικασίας της οπισθοδιάδοσης (backpropagation), είτε διορθώνοντας τον ρυθμό μάθησης (a), είτε διορθώνοντας τη συνιστώσα της κλίσης. Υπάρχει μια πληθώρα διαθέσιμων αλγορίθμων βελτιστοποίησης, με τους πιο γνωστούς να είναι ο Adam και ο SGD (Stochastic Gradient Descent) που παρά τις διαφορές στη λειτουργία τους, ξεχωρίζουν για την υψηλή αποδοτικότητα τους. Ο Adam χρησιμοποιεί προσαρμοστικούς ρυθμούς εκμάθησης και υπολογίζει εκθετικά κινούμενες μέσες τιμές των πρώτων και δεύτερων gradient moments, ενώ ο SGD ανανεώνει τα βάρη του δικτύου χρησιμοποιώντας ένα μικρό δείγμα δεδομένων (mini-batch).

Η βελτιστοποίηση των υπερπαραμέτρων που αναλύθηκαν απαιτεί συνήθως προσαρμογή των τιμών τους, μια διαδικασία που μπορεί να γίνει είτε εμπειρικά μέσω χειροκίνητης ρύθμισης, είτε με χρήση αυτοματοποιημένων μεθόδων, όπως η αναζήτηση πλέγματος, η τυχαία δειγματοληψία ή ο βελτιστοποιητής Bayesian. Αυτές οι μέθοδοι αξιοποιούν αλγόριθμους για να εξερευνήσουν τον χώρο των υπερπαραμέτρων και να εντοπίσουν τις βέλτιστες τιμές, βελτιώνοντας έτσι την απόδοση και την ικανότητα του μοντέλου να γενικεύει στα δεδομένα.

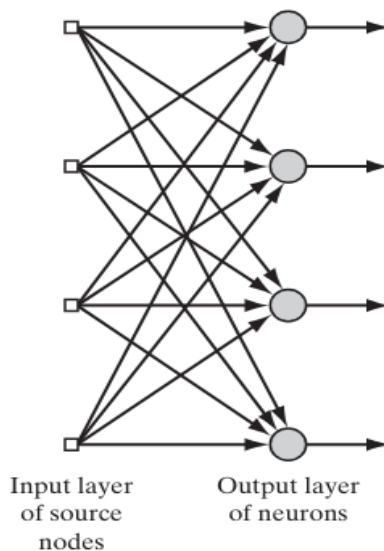
² <https://www.coursera.org/learn/neural-networks-deep-learning> [Accessed: 12 July 2024]

1.2.4 Αρχιτεκτονικές Νευρωνικών Δικτύων

Μια ακόμα σημαντική πτυχή στην παραγωγή αξιόπιστων προβλέψεων αλλά και για την εξασφάλιση της βέλτιστης απόδοσης ενός ΝΔ είναι η επιλογή της σωστής αρχιτεκτονικής. Για τη δημιουργία μιας δομής με τη δυνατότητα να μαθαίνει, να αποθηκεύει γνώση και τελικά να προβλέπει, απαιτείται συχνά η σύνδεση ενός μεγάλου αριθμού νευρώνων σε στοιβάδες. Σε ένα μοντέλο υπολογιστικής όρασης (ΥΟ) για παράδειγμα, όπου απαιτούνται πολλοί νευρώνες και σκοπός είναι η αναγνώριση εικόνας σε ανθρώπινα πρόσωπα, σε μια πρώτη στοιβάδα νευρώνων το μοντέλο μπορεί να ξεκινήσει να διακρίνει γραμμές και σχήματα, ενώ στις επόμενες στοιβάδες να είναι σε θέση να αναγνωρίζει συγκεκριμένα χαρακτηριστικά όπως μια μύτη ή ένα στόμα. Αυτή η σταδιακή προσέγγιση και σύνδεση πολλών νευρώνων σε στοιβάδες επιτρέπει ένα σύστημα να αναπτύξει σύνθετες συσχετίσεις στα δεδομένα εισόδου, βοηθώντας το να επιτύχει υψηλή ακρίβεια σε οποιαδήποτε διεργασία του ανατεθεί. Υπάρχουν τρεις βασικές κατηγορίες αρχιτεκτονικών δικτύων που διαφέρουν θεμελιωδώς μεταξύ τους [4] και που μπορούμε να διακρίνουμε:

1. Δίκτυα πρόσθιας τροφοδότησης (FeedForward Net) ενός επίπεδου

Τα δίκτυα πρόσθιας τροφοδότησης ενός επίπεδου αποτελούν την απλούστερη μορφή ενός δικτύου, με τον όρο "ενός επίπεδου" να αναφέρεται στο επίπεδο εξόδου των υπολογιστικών κόμβων. Τα δίκτυα αυτού του είδους αποτελούνται από ένα μόνο στρώμα νευρώνων με το επίπεδο εισόδου να μην λαμβάνεται υπόψη καθώς δεν πραγματοποιείται κανένας υπολογισμός σε αυτό. Η πληροφορία ρέει μόνο από την είσοδο προς την έξοδο, χωρίς αντίστροφη ροή.

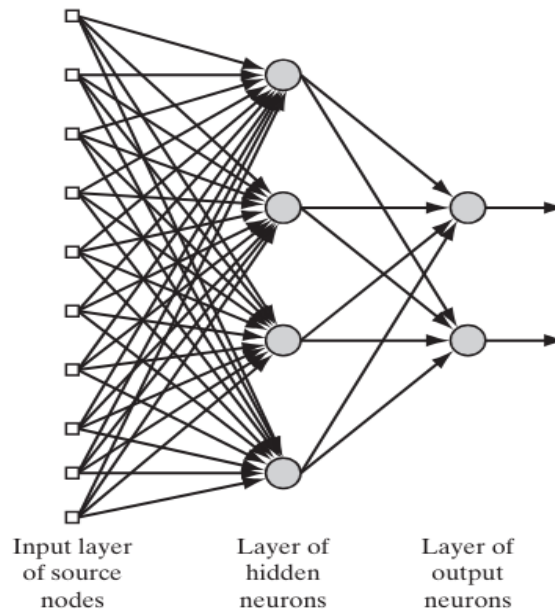


Εικόνα 10: FeedForward Net ενός επίπεδου [4]

2. Δίκτυα πρόσθιας τροφοδότησης (FeedForward Net) πολλών επιπέδων

Αυτή η κατηγορία χαρακτηρίζεται από την ύπαρξη ενός ή περισσότερων κρυφών στοιβάδων, των οποίων οι υπολογιστικοί κόμβοι αποκαλούνται κρυφοί νευρώνες. Η έννοια αυτή προκύπτει από το γεγονός ότι οι στοιβάδες που παρεμβαίνουν μεταξύ του επίπεδου εισόδου και του επίπεδου εξόδου δεν είναι αμέσως ορατές ούτε από την είσοδο ούτε από την έξοδο

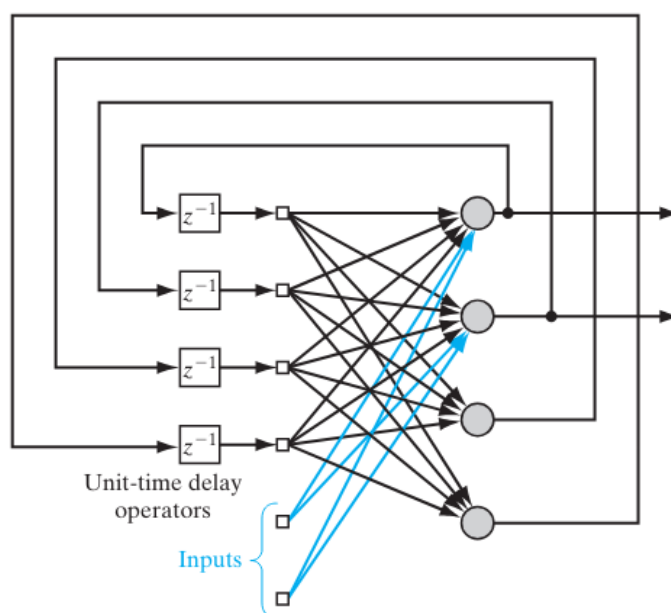
του δικτύου αυτού καθαυτές. Είναι πολύ πιο πολύπλοκα από τα δίκτυα πρόσθιας τροφοδότησης ενός επίπεδου και επιτρέπουν την ανάδειξη πιο σύνθετων χαρακτηριστικών.



Εικόνα 11: Fully connected FeedForward Net πολλών επιπέδων [4]

3. Αναδρομικά δίκτυα (Recurrent Neural Network)

Τα αναδρομικά ΝΔ διαφέρουν από τα δίκτυα πρόσθιας τροφοδότησης από το γεγονός ότι έχουν τουλάχιστον έναν βρόχο ανάδρασης. Η παρουσία των βρόχων αναδράσεως έχει βαθιά επίδραση στη δυνατότητα μάθησης και την απόδοση του δικτύου, καθώς επιτρέπει την ενσωμάτωση προηγούμενων πληροφοριών, την εύρεση δομικών σχέσεων στα δεδομένα και την αντιμετώπιση πολύπλοκων προβλημάτων.



Εικόνα 12: Αναδρομικό δίκτυο με κρυφούς νευρώνες [4]

1.3 Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks) αποτελούν μια εξειδικευμένη κατηγορία perceptron πολλών επιπέδων και αποδεικνύονται ιδιαίτερα αποτελεσματικά στην ταξινόμηση προτύπων. Η ουσιαστική διαφορά μεταξύ ενός ΣΝΔ και ενός παραδοσιακού ΝΔ είναι ότι τα πρώτα δεν αντιμετωπίζουν σημαντικές δυσκολίες με την υπολογιστική πολυπλοκότητα που απαιτείται για την επεξεργασία δεδομένων εικόνων και για αυτόν τον λόγο χρησιμοποιούνται ευρέως στην ανίχνευση αντικειμένων (Object detection) ,στην ταξινόμηση εικόνων (Image classification) καθώς και στην μεταφορά του στυλ του ΝΔ σε εικόνες (Neural Style Transfer) [7]. Η ανάπτυξη δικτύων τέτοιου είδους βασίζεται στη μελέτη των βιολογικών νευρώνων του οπτικού φλοιού της γάτας, όπως μελετήθηκε από τους Hebel και Wiesel τη δεκαετία του 1960. Τα ΣΝΔ είναι ειδικά σχεδιασμένα για την αναγνώριση δισδιάστατων μοτίβων χωρίς να είναι ευαίσθητα σε κλιμάκωση, στρέβλωση ή άλλες παραμορφώσεις μιας εικόνας [4].

1.3.1 Συνέλιξη Πινάκων για την ανίχνευση ακρών

Σε ένα ΝΔ, το πρώτο βήμα για την αναγνώριση προτύπων είναι η εύρεση όλων των ακμών μέσα σε μια εικόνα. Στα πρώτα επίπεδα του δικτύου, αναγνωρίζονται συνήθως χαρακτηριστικά όπως είναι οι ακμές και οι άκρες, ενώ καθώς το δίκτυο εμβαθύνει και αλλάζουν τα συναπτικά του βάρη, το δίκτυο αρχίζει να αναγνωρίζει πιο γενικά χαρακτηριστικά καταλήγοντας τελικά στις τελευταίες του στοιβάδες που είναι σε θέση να αναγνωρίζει ολοκληρωμένα αντικείμενα. Στην Εικόνα 13 παρουσιάζεται αυτή ακριβώς η διαδικασία μάθησης ενός ΣΝΔ για την αναγνώριση εικόνων από πρόσωπα ανθρώπων.



Εικόνα 13: Διαδικασία μάθησης συνελικτικού νευρωνικού δικτύου³

Η διαδικασία της συνέλιξης πινάκων αποτελεί τον βασικό πυλώνα στην επεξεργασία μιας εικόνας και ιδιαίτερα στην ανίχνευση ακρών, η οποία αποτελεί το αρχικό βήμα για την αναγνώριση προτύπων. Κατά τη διάρκειά της, στην εικόνα που αναπαρίσταται υπό την μορφή πίνακα, εφαρμόζεται ένα μικρό φίλτρο (Filter ή Kernel) για να εντοπιστούν τα σημεία όπου υπάρχουν αλλαγές στη φωτεινότητα ή το χρώμα. Αυτές οι αλλαγές, συνήθως, υποδεικνύουν τα σύνορα ή τις ακμές των αντικειμένων στην εικόνα. Μια παρουσίαση ενός απλού παραδείγματος αυτής της διαδικασίας παρέχεται στην Εικόνα 14.

³ <https://www.coursera.org/learn/convolutional-neural-networks> [Accessed: 12 July 2024]

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$

6×6 3×3 4×4

Vertical filter

Εικόνα 14: Παράδειγμα συνέλιξης πινάκων για την ανίχνευση ακρών

Το παραπάνω παράδειγμα ανίχνευσης ακρών και ακμών μέσω συνέλιξης παρουσιάζει μια ασπρόμαυρη εικόνα 6×6 pixels ενός καναλιού (channel), αναπαριστώντας την ως ένα πίνακα 6×6 . Στην οπτική αναπαράσταση της εικόνας, τα πιο φωτεινά pixel αντιστοιχούν στον αριθμό 10, ενώ τα πιο σκούρα στον αριθμό 0. Ο σκοπός ενός ΝΔ όπως προαναφέρθηκε είναι να δημιουργήσει συσχετίσεις στα δεδομένα. Αυτό επιτυγχάνεται αρχικά εντοπίζοντας τα όρια μεταξύ δύο χρωμάτων με τη χρήση ενός 3×3 φίλτρου, το οποίο μπορεί να ανιχνεύσει τις κατακόρυφες γραμμές. Η διαδικασία της συνέλιξης πινάκων για τα ΣΝΔ περιγράφεται ως εξής: κάθε υποπίνακας- διάνυσμα 3×3 (ίδιες διαστάσεις με του φίλτρου) της εικόνας πολλαπλασιάζεται ένα προς ένα με τα στοιχεία του πίνακα του φίλτρου και στη συνέχεια αυτά αθροίζονται. Ο πίνακας που προκύπτει περιλαμβάνει όλα τα αποτελέσματα της συνέλιξης κάθε υποπίνακα 3×3 με το κατακόρυφο φίλτρο. Όπως παρατηρείται και από την εικόνα, όπου τα pixel με την τιμή 0 αναπαριστούν τα λιγότερο φωτεινά σημεία και τα pixel με τιμή 30 τα πιο φωτεινά, μέσω της συνέλιξης, η περιοχή (region) όπου υπάρχει ένα όριο ή μια ακμή μεταξύ των χρωμάτων εντοπίζεται με επιτυχία.

Η διαδικασία που περιγράφηκε παραπάνω φυσικά μπορεί να επεκταθεί και να εφαρμοστεί σε φωτογραφίες με πολύ μεγαλύτερες διαστάσεις και ανάλυση, καθώς και σε εικόνες πολλών καναλιών χρωμάτων (RGB). Αυτό σημαίνει ότι μπορούμε να εφαρμόσουμε την ίδια διαδικασία για παράδειγμα σε μια εικόνα $6 \times 6 \times 3$ άπλα αλλάζοντας αντίστοιχα το φίλτρο από έναν πίνακα $3 \times 3 \times 1$ σε έναν πίνακα $3 \times 3 \times 3$. Είναι συχνό στα ΣΝΔ το φίλτρο να αντιμετωπίζεται ως παράμετρος του δικτύου και οι τιμές του να προσαρμόζονται σύμφωνα με τη διαδικασία της οπισθοδιάδοσης (back propagation)³.

$$\begin{pmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{pmatrix}$$

Εικόνα 15: Φίλτρο 3×3 με τα επιμέρους στοιχεία του ως παράμετροι ενός δικτύου

³ <https://www.coursera.org/learn/convolutional-neural-networks> [Accessed: 12 July 2024]

1.3.2 Επίπεδα Συνελκτικών Νευρωνικών Δικτύων

Για να είναι σε θέση τα ΣΝΔ να αναγνωρίζουν χαρακτηριστικά και τελικά να καταλήγουν σε συμπεράσματα όπως η πρόβλεψη της πιθανότητας μιας κλάσης σε προβλήματα ταξινόμησης, αποτελούνται από μια πληθώρα νευρώνων που λειτουργούν συμπληρωματικά. Σε ένα τέτοιο δίκτυο συνήθως διακρίνουμε τρία βασικά επίπεδα:

1. Επίπεδο Συνέλιξης (Convolutional Layer)

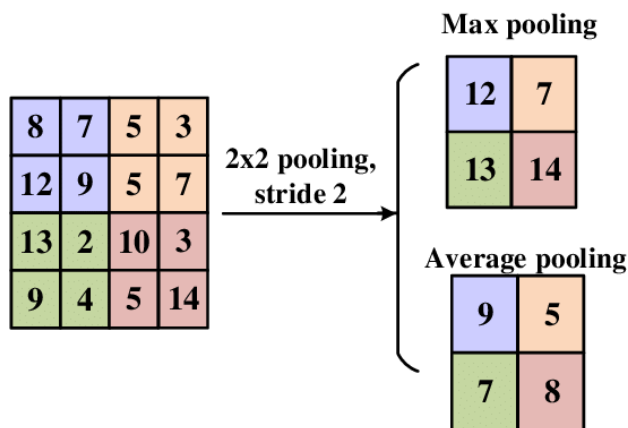
Τα επίπεδα συνέλιξης όπως αναφέρθηκε και στο παράδειγμα της Εικόνα 14 είναι υπεύθυνα για την εξαγωγή χαρακτηριστικών από τις εικόνες μέσω της συνέλιξης με φίλτρα (kernels). Κάθε φίλτρο εκτός από άκρες ή ακμές μπορεί να ανιχνεύει και διάφορα σχήματα δημιουργώντας με αυτόν τον τρόπο έναν "χάρτη χαρακτηριστικών" (feature map) που απεικονίζει την παρουσία αυτών των χαρακτηριστικών στην εικόνα αποτελεί την είσοδο για τα επόμενα επίπεδα [7]. Ένα επίπεδο συνέλιξης μπορεί να περιγραφεί με την χρήση μαθηματικών όρων χρησιμοποιώντας την παρακάτω εξίσωση:

$$a^{[1]} = g(w^{[1]} \times a^{[0]} + b^{[1]})$$

Σύμφωνα με το παράδειγμα που παρουσιάστηκε, ο πίνακας $a^{[0]}$ αναφέρεται στον πίνακα με διαστάσεις 6x6 που αντιστοιχεί στην εικόνα εισόδου. Ο πίνακας βαρών $w^{[1]}$ αντιστοιχεί στο φίλτρο διαστάσεων 3x3, ενώ ο πίνακας $b^{[1]}$ είναι μια εξωτερική πόλωση που προστίθεται. Μετά την εφαρμογή μιας ΣΕ, το αποτέλεσμα αντιστοιχεί στον πίνακα $a^{[1]}$, ο οποίος αποτελεί και το τελικό αποτέλεσμα που θα περάσει στο επόμενο επίπεδο την ίδια διαδικασία.

2. Επίπεδο υποδειγματοληψίας (Pooling Layer)

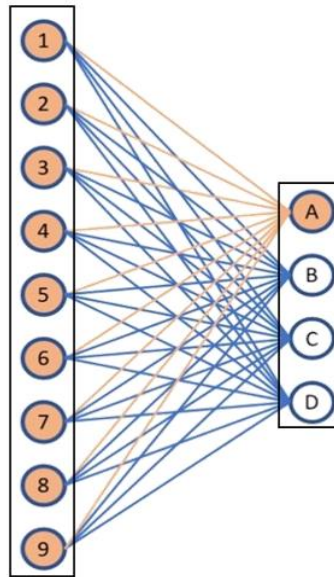
Τα επίπεδα υποδειγματοληψίας, επίσης γνωστά ως επίπεδα ομαδοποίησης, μειώνουν τον αριθμό των συντελεστών και απλοποιούν τα διανύσματα των χαρακτηριστικών που εξάγονται από τα επίπεδα συνέλιξης μειώνοντας τις διαστάσεις τους. Χρησιμοποιούνται για τη μείωση του αριθμού των παραμέτρων του δικτύου και, κυρίως, για τη βελτίωση της απόδοσής του. Η λειτουργία τους είναι παρόμοια με αυτή των επιπέδων συνέλιξης. Τα πιο διαδεδομένα είδη υποδειγματοληψίας είναι το Max Pooling και το Average Pooling, όπως παρουσιάζονται στην Εικόνα 16 [8].



Εικόνα 16: Διαδικασίες υποδειγματοληψίας Max Pooling και Average Pooling [8]

3. Πλήρως Συνδεδεμένο Επίπεδο (Fully Connected Layer)

Σε αυτό το επίπεδο περιλαμβάνονται νευρώνες που είναι άμεσα συνδεδεμένοι με τους νευρώνες των δύο γειτονικών επιπέδων, χωρίς να συνδέονται με οποιαδήποτε άλλα επίπεδα εντός αυτών, όπως ακριβώς και στην παραδοσιακή μορφή ενός ΝΔ [7]. Συνήθως οι περισσότεροι παράμετροι ενός δικτύου εντοπίζονται στα πλήρως συνδεδεμένα επίπεδα. Χρησιμοποιούνται για την τελική επεξεργασία των χαρακτηριστικών και την παραγωγή των τελικών αποτελεσμάτων πριν την εφαρμογή της τελικής ΣΕ, η οποία για τα ΣΝΔ είναι συνήθως μια Softmax ή μια Sigmoid.



Εικόνα 17: Πλήρως συνδεδεμένα επίπεδα συνελικτικού νευρωνικού δικτύου⁴

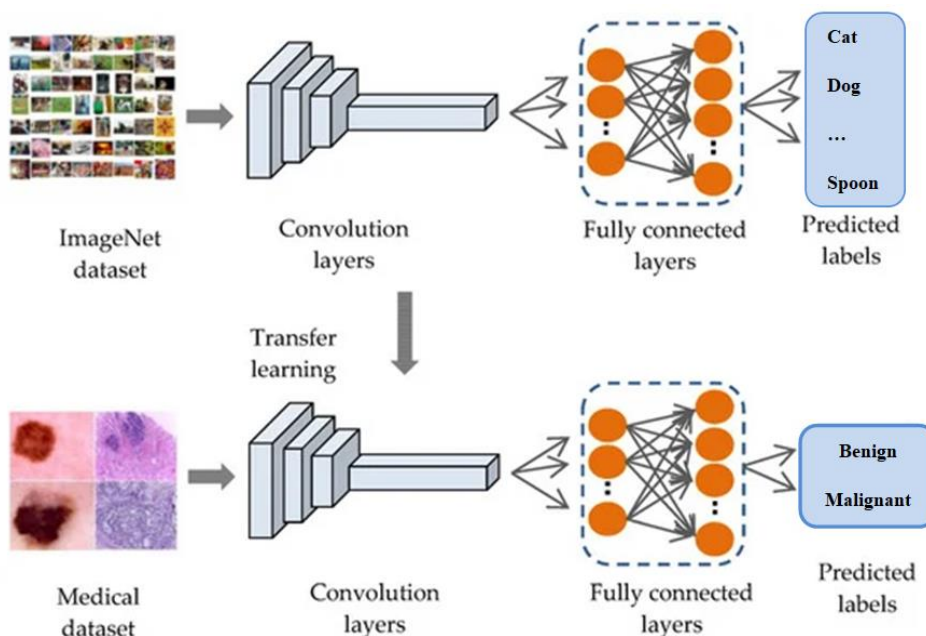
Αξίζει επιπλέον να σημειωθεί ότι σε ένα ΝΔ, ένα επίπεδο (layer) θεωρείται ως τέτοιο μόνο εάν διαθέτει παραμέτρους. Αυτό σημαίνει πρακτικά ότι ένα επίπεδο υποδειγματοληψίας δεν μπορεί από μόνο του να αποτελέσει ένα επίπεδο. Μόνο ο συνδυασμός αυτού και ενός επιπέδου συνέλιξης μπορεί να χαρακτηριστεί ως ένα επίπεδο, όπως γίνεται αναφορά σε πολλές αρχιτεκτονικές και σχεδόν σε όλες τις δημοσιεύσεις πάνω στα ΣΝΔ.

⁴ <https://builtin.com/machine-learning/fully-connected-layer> [Accessed: 12 July 2024]

1.3.3 Προεκπαιδευμένα μοντέλα και Μεταφορά Μάθησης

Η ΥΟ αποτελεί ένα πεδίο όπου η μεταφορά μάθησης και γνώσης αποδεικνύεται πολύ χρήσιμη και γίνεται πλέον από σχεδόν όλους τους ερευνητές, εκτός αν υπάρχουν στη διάθεσή τους εξαιρετικά μεγάλα σύνολα δεδομένων. Στη μεταφορά μάθησης (transfer learning), αντί να εκπαιδεύεται ένα μοντέλο και να εξάγονται βάρη κάθε φορά από το μηδέν μέσω αρχικοποίησης, είναι σύνηθες να χρησιμοποιούνται τα βάρη που έχει ήδη εκπαιδεύσει κάποιος άλλος σε μια συγκεκριμένη αρχιτεκτονική ενός δικτύου. Η εκπαίδευση ενός ΣΝΔ χρησιμοποιώντας σύνολα δεδομένων εικόνων μπορεί σε πολλές περιπτώσεις να διαρκέσει αρκετές εβδομάδες και πιθανότατα να απαιτήσει πολλούς πόρους, ειδικά από πλευράς κάρτας γραφικών (GPU). Όταν, όμως, χρησιμοποιείται ένα ήδη εκπαιδευμένο μοντέλο, τα βάρη, οι παράμετροι, καθώς και η γνώση του μεταφέρονται για να πραγματοποιηθεί μια διαφορετική διεργασία από αυτήν για την οποία είχε αρχικά εκπαιδευτεί το μοντέλο αυτό [9].

Οι περισσότερες αρχιτεκτονικές ΣΝΔ έχουν εκπαιδευτεί σε σύνολα δεδομένων όπως το ImageNet, το οποίο περιλαμβάνει 1000 κλάσεις αντικειμένων. Όταν λοιπόν δεν υπάρχουν διαθέσιμες εικόνες σε μεγάλο αριθμό για την εκπαίδευση, μια συνηθισμένη πρακτική είναι να "παγώνουμε" τις παραμέτρους σε όλα τα αρχικά επίπεδα του μοντέλου και να εκπαιδεύουμε μόνο τις παραμέτρους στο τελευταίο επίπεδο που μπορεί να είναι μια συνάρτηση Softmax ή για τις περισσότερες εφαρμογές ιατρικής διάγνωσης, μια Sigmoid. Αυτή η προσέγγιση αλλάζει λίγο όταν διαθέτουμε ένα πολύ μεγαλύτερο σύνολο δεδομένων, καθώς σε αυτήν την περίπτωση συνήθως παγώνουμε μόνο τα πρώτα επίπεδα και στη συνέχεια εκπαιδεύουμε τα περισσότερα από τα μετέπειτα επίπεδα. Σε κάθε περίπτωση, έχει παρατηρηθεί ότι με αυτόν τον τρόπο, ακόμα και με ένα μικρό σύνολο δεδομένων, μπορεί να επιτευχθεί μια πολύ καλή ακρίβεια και εξαιρετικά ικανοποιητικά αποτελέσματα.



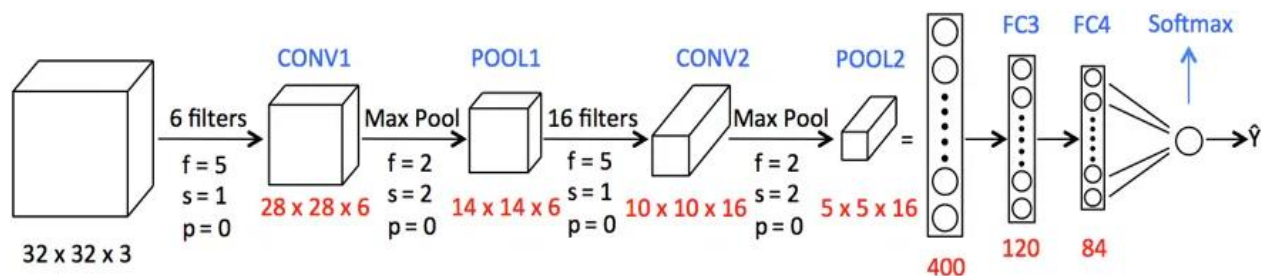
Εικόνα 18: Μεταφορά μάθησης από το σύνολο δεδομένων ImageNet σε σύνολο ιατρικών εικόνων [9]

1.3.4 Αρχιτεκτονικές διαφορών μοντέλων

1.3.4.1 Η απλή αρχιτεκτονική του LeNet-5

Αν και κατά τη διάρκεια των χρόνων έχουν αναπτυχθεί πολλά μοντέλα και αρχιτεκτονικές ΣΝΔ, το LeNet-5 θεωρείται το πιο απλό και βασικό, συχνά χρησιμοποιούμενο μοντέλο για να επεξηγηθεί η λειτουργία των συνελικτικών δικτύων. Προτάθηκε και δημιουργήθηκε το 1989, με σκοπό να χρησιμοποιηθεί σε προβλήματα αναγνώρισης χειρόγραφων αριθμών από εικόνες, όπως αυτές του γνωστού συνόλου δεδομένων MNIST [10]. Αποτέλεσε ένα από τα πρώτα επιτυχημένα μοντέλα ΣΝΔ για εφαρμογές τέτοιου είδους και η δομή του μπορεί να περιγραφεί ως εξής:

Αρχικά, η είσοδος μπορεί να είναι μια εικόνα 32x32 είτε με ένα κανάλι (grayscale) είτε με 3 κανάλια (RGB). Ακολουθεί ένα συνελικτικό επίπεδο με φίλτρο 5x5 και βήμα 1, ακολουθούμενο από max pooling με μέγεθος φίλτρου 2x2 και βήμα 2. Τα χαρακτηριστικά που εξάγονται συνελίσσονται με ένα φίλτρο 5x5 και βήμα 1 και υποδειγματοληπτούνται μέσω Max Pooling με μέγεθος φίλτρου ξανά 2x2 και βήμα 2. Στη συνέχεια, τα χαρακτηριστικά εισάγονται στα πλήρως συνδεδεμένα επίπεδα του δικτύου. Το πρώτο πλήρως συνδεδεμένο επίπεδο έχει 400 νευρώνες εισόδου και 120 νευρώνες εξόδου, ενώ το επόμενο έχει αντίστοιχα 120 νευρώνες εισόδου και 84 νευρώνες εξόδου. Το τελευταίο πλήρως συνδεδεμένο επίπεδο αποτελείται από 84 νευρώνες εισόδου και 10 νευρώνες εξόδου, αντιπροσωπεύοντας τις δέκα δυνατές κλάσεις των δεδομένων εξόδου (ψηφία από το μηδέν μέχρι το εννέα). Το διάνυσμα εξόδου υποβάλλεται τελικά σε μια ΣΕ Softmax για την πρόβλεψη της τελικής κλάσης [10]. Η δομή που περιγράφηκε αντιστοιχεί στην Εικόνα 19 και απεικονίζει την αρχιτεκτονική του LeNet-5 με τα συνελικτικά και πλήρως συνδεδεμένα επίπεδα, καθώς και τις συνδέσεις μεταξύ τους.



Εικόνα 19: Αρχιτεκτονική δικτύου LeNet-5⁵

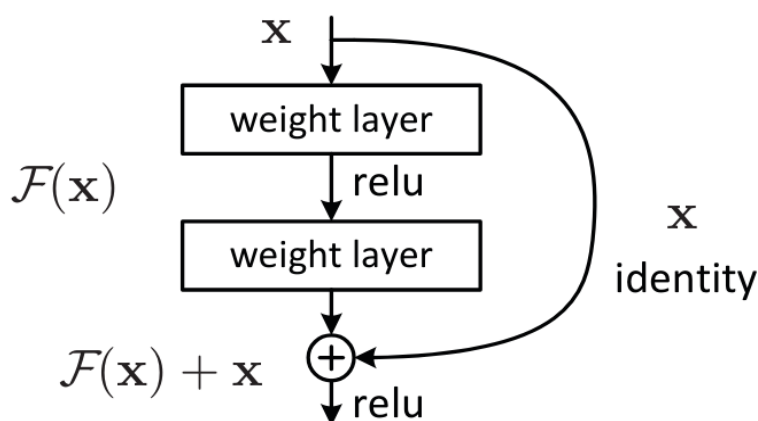
Πάνω σε αυτή την αρχιτεκτονική βασίστηκαν μοντέλα που βγήκαν μεταγενέστερα και θριάμβευσαν στους διαγωνισμούς ImageNet, όπως είναι το AlexNet και το VGG-16. Το AlexNet, είναι πολύ μεγαλύτερο και πολύ πιο βαθύ από το LeNet-5, καθώς χρησιμοποιεί πέντε συνελικτικά επίπεδα και τρία πλήρως συνδεδεμένα επίπεδα. Πρωτοπόρησε με τη χρήση ReLU συναρτήσεων ενεργοποίησης, οι οποίες επιτάχυναν σημαντικά την εκπαίδευση του δικτύου, καθώς και με την τεχνική Dropout η οποία μείωνε το πρόβλημα της υπερεκπαίδευσης (overfitting) [11]. Το VGG-16, που δημοσιεύτηκε το 2014, ακλουθώντας μια παρόμοια δομή, αποτελείται από δεκατρία συνελικτικά επίπεδα και τρία πλήρως συνδεδεμένα επίπεδα. Σε αντίθεση με προηγούμενα μοντέλα, το δίκτυο αυτό χρησιμοποιεί μικρά συνελικτικά φίλτρα (3x3) σε όλα τα επίπεδα επιτυγχάνοντας έτσι καλύτερη εξαγωγή χαρακτηριστικών και καλύτερη συνολική απόδοση [12].

⁵ <https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524>
[Accessed: 12 July 2024]

1.3.4.2 Το πρόβλημα που λύνει το δίκτυο ResNet

Μετά την εμφάνιση και την εντυπωσιακή νίκη του AlexNet στον διαγωνισμό ImageNet το 2012, η επιστημονική κοινότητα έστρεψε την προσοχή της στη δημιουργία όσο γίνεται περισσότερων μοντέλων με πολλά επίπεδα, όπως τα μοντέλα VGG που προαναφέρθηκαν, με την πεποίθηση ότι η αύξηση του βάθους των δικτύων θα οδηγούσε σε βελτίωση της απόδοσής τους. Ωστόσο, σε αντίθεση με ότι αρχικά πιστευόταν, τα δίκτυα αυτά καθώς γίνονταν βαθύτερα παρουσίαζαν σημαντικά προβλήματα και η απόδοσή τους μειωνόταν δραστικά. Αυτό το φαινόμενο οφείλεται σε ένα πρόβλημα γνωστό ως “vanishing gradients” όπου οι παράγωγοι που χρησιμοποιούνται από τους αλγόριθμους βελτιστοποίησης για την εκπαίδευση των δικτύων “εξαφανίζονται” με αποτέλεσμα τα δίκτυα αυτά να μην είναι ικανά πλέον να μάθουν από τα δεδομένα [13].

Στο πρόβλημα των “vanishing gradients” ήρθε να δώσει λύση το ResNet (Residual Neural Network), μια νέα αρχιτεκτονική που παρουσιάστηκε από ερευνητές της Microsoft και κέρδισε τον διαγωνισμό κατηγοριοποίησης ILSVRC το 2015, επιτυγχάνοντας μάλιστα ακρίβεια 96.43% στο σύνολο δεδομένων ImageNet [14]. Βασισμένοι σε μια τεχνική που χρησιμοποιείται και στα δίκτυα πρόσθιας τροφοδότησης, οι ερευνητές δημιούργησαν τα Residual Blocks, τα οποία χρησιμοποιούν συνδέσεις παράκαμψης (skip connections) για να επιτρέπουν στην πληροφορία να παρακάμπτει ένα ή περισσότερα επίπεδα και να μεταφέρεται κατευθείαν στο επόμενο μπλοκ [14]. Αυτό έχει ως αποτέλεσμα να επιλύεται το πρόβλημα των “vanishing gradients”, καθώς ακόμα και αν οι αλλαγές των βαρών είναι μηδενικές, η συνάρτηση εξόδου του μπλοκ δεν θα είναι ποτέ μηδενική, καθώς προστίθεται με την είσοδο του.

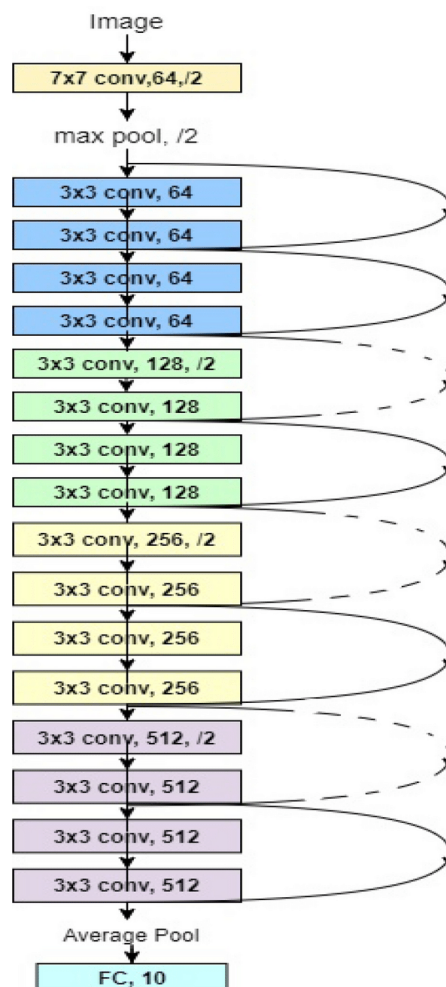


Εικόνα 20: Residual Block [14]

Η πιο απλή δομή για να περιγραφεί η αρχιτεκτονική ενός ResNet είναι αυτή του ResNet-18. Η εικόνα σε αυτή την αρχιτεκτονική αρχικά εισάγεται στο πρώτο συνελκτικό επίπεδο (7x7 kernel και 64 φίλτρα) και αφού της εφαρμοστεί Batch normalization και ΣΕ ReLU, στη συνέχεια περνάει από ένα Max pooling (3x3). Στη συνέχεια, η είσοδος προωθείται στα επίπεδα που περιέχουν τα Residual Blocks με την παρακάτω σειρά:

1. Conv2_x: Περιέχει 2 Residual Blocks, καθένα από τα οποία αποτελείται από 2 συνελκτικά επίπεδα (3x3 kernel, 64 φίλτρα).
2. Conv3_x: Περιέχει 2 Residual Blocks, καθένα από τα οποία αποτελείται από 2 συνελκτικά επίπεδα (3x3 kernel, 128 φίλτρα). Περιλαμβάνει επιπλέον Downsampling με βήμα 2.
3. Conv4_x: Περιέχει 2 Residual Blocks, καθένα από τα οποία αποτελείται από 2 συνελκτικά επίπεδα (3x3 kernel, 256 φίλτρα). Και εδώ περιλαμβάνεται Downsampling με βήμα 2.
4. Conv5_x: Περιέχει 2 Residual Blocks, καθένα με 2 συνελκτικά επίπεδα (3x3 kernel, 512 φίλτρα) ενώ περιλαμβάνεται και Downsampling με βήμα 2.

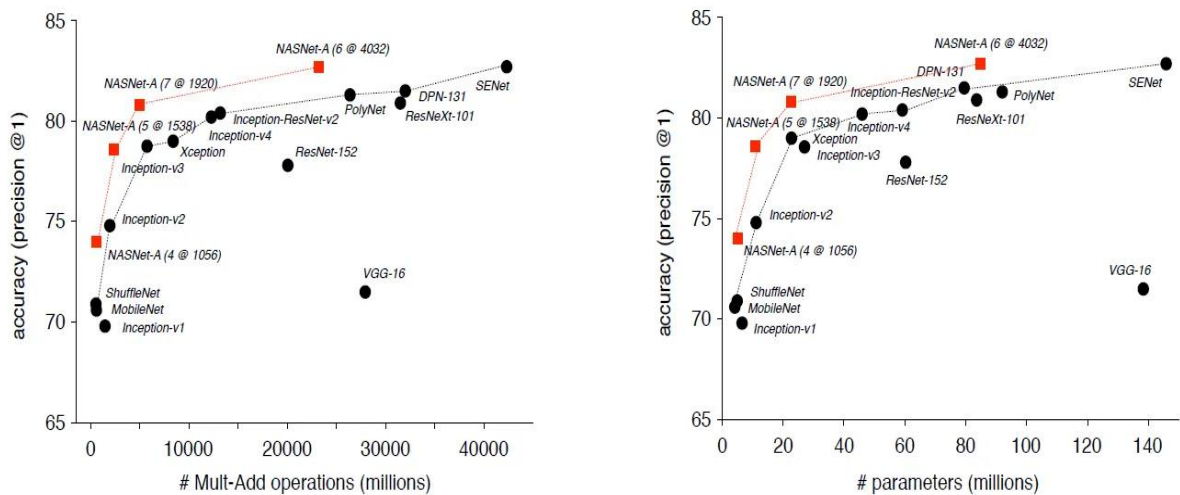
Στο τέλος, η έξοδος περνάει από ένα επίπεδο υποδειγματοληψίας Average pooling (7x7) και προωθείται στο τελικό επίπεδο, το οποίο είναι ένα πλήρως συνδεδεμένο επίπεδο που χρησιμοποιεί μια ΣΕ Softmax ή μια Sigmoid για την ταξινόμηση πολλών ή μόνο δυο κλάσεων αντίστοιχα. Η αρχιτεκτονική του ResNet18 όπως περιγράφηκε, παρουσιάζεται και στην Εικόνα 21 [15]:



Εικόνα 21: Αρχιτεκτονική ResNet18 [15]

1.3.4.3 Η προσέγγιση της αρχιτεκτονικής NASNet

Σε αντίθεση με τα προηγούμενα μοντέλα, των οποίων η αρχιτεκτονική είναι σταθερή και βασίζεται σε παραδοσιακές πρακτικές για την εμβάθυνση του δικτύου, έρχεται το NASNet (Neural Architecture Search Network) το οποίο αποτελεί ένα από τα πιο πρόσφατα ανεπτυγμένα και προηγμένα μοντέλα ΣΝΔ. Δημιουργήθηκε από την ομάδα της Google Brain το 2018 και ξεχωρίζει για τη μοναδική του αρχιτεκτονική, η οποία δεν είναι συγκεκριμένη, αλλά διαμορφώνεται αυτόματα μέσω της αναζήτησης νευρωνικών δομών. Αυτή η διαδικασία επιτυγχάνεται μέσω ενός ελεγκτή RNN, ο οποίος διεξάγει αναζήτηση αρχικά για το σύνολο δεδομένων CIFAR-10 και αφού προκύψουν κάποια βάρη, στην συνέχεια τα χρησιμοποιεί για να κάνει αναγνώριση στο σύνολο ImageNet, επιλέγοντας και προσαρμόζοντας δομές που βελτιστοποιούν την απόδοση του μοντέλου [16]. Η δομή του NASNet αποτελείται από πολλαπλά βαθιά συνελκτικά επίπεδα, ακολουθούμενα από επίπεδα υποδειγματοληψίας, τα οποία συμβάλλουν στην εξαγωγή χαρακτηριστικών από τις εισόδους. Το NASNet μετά από 4 ημέρες εκπαίδευσης σε 500 μονάδες GPU, πετυχαίνει εντυπωσιακή ακρίβεια και διαθέτει σχετικά λιγότερες παραμέτρους σε σύγκριση με άλλες αρχιτεκτονικές [16].



Εικόνα 22: Απόδοση και σύνολο παραμέτρων NASNet σε σχέση με άλλα μοντέλα [16]

Με την ανάλυση όλων των βασικών αρχιτεκτονικών, είτε χρησιμοποιούνται παραδοσιακοί τρόποι για τη βελτιστοποίηση της απόδοσης είτε εφαρμόζονται καινοτόμες μέθοδοι, καταλήγουμε στο συμπέρασμα ότι υπάρχουν πολλές διαθέσιμες αρχιτεκτονικές και δομές που μπορούν να δοκιμαστούν για να επιτευχθεί το επιθυμητό αποτέλεσμα και η καλύτερη ακρίβεια σε ένα μοντέλο που πραγματοποιεί κατηγοριοποίηση και ταξινόμηση. Για τη δική μας περίπτωση, η επιλογή του σωστού μοντέλου για την ανάπτυξη της εφαρμογής αναγνώρισης όγκων μόνο τυχαία δεν θα μπορούσε να είναι. Στηριζόμενοι στο ευρέως γνωστό και μόνο διαθέσιμο ανοιχτό σύνολο δεδομένων υπερήχων του μαστού “Dataset of breast ultrasound images” [17], οι ερευνητές και συγγραφείς αυτού δημοσίευσαν μια επιπλέον μελέτη στα τέλη του 2019 που αφορούσε την εκπαίδευση διαφόρων μοντέλων πάνω στο σύνολο δεδομένων BUSI μαζί με κάποιες μεθόδους τροποποίησης και αύξησης των εικόνων χρησιμοποιώντας τεχνικές μεταφοράς μάθησης. Η μελέτη απέδειξε ότι, σε κάθε περίπτωση, το NASNet επέδειξε την καλύτερη απόδοση, είτε με επέκταση του συνόλου δεδομένων με διάφορους τρόπους είτε χωρίς αυτή [18]. Στα επόμενα κεφάλαια, θα εξεταστούν και θα συγκριθούν οι αρχιτεκτονικές που δοκιμάστηκαν από αυτούς τους ερευνητές ενώ τελικά θα επιλεγεί το μοντέλο που θα έχει την καλύτερη συνολική απόδοση, επιβεβαιώνοντας ή διαψεύδοντας τα διαθέσιμα ευρήματα για το συγκεκριμένο σύνολο δεδομένων.

1.4 Συμπεράσματα Κεφαλαίου

Στο εν λόγω κεφάλαιο πραγματοποιήθηκε μια εμπειρισταωμένη ανάλυση της ιστορίας και της εξέλιξης της ΤΝ, από τις ιδέες του Αριστοτέλη μέχρι τη σύγχρονη εποχή και την άνοδο των μοντέλων ΤΝ με ηγέτες μεγάλους κολοσσούς στον χώρο της τεχνολογίας. Έγινε επίσης μια αναφορά στις διάφορες κατηγορίες της υπολογιστικής νοημοσύνης και της μηχανικής μάθησης, με έμφαση στα σημαντικότερα χαρακτηριστικά τους και στις διαφορές που έχουν. Επιπλέον, αναδείχθηκε η δομή των νευρώνων του εγκεφάλου και η επίδρασή αυτών στη έμπνευση και την δημιουργία των τεχνητών ΝΔ, καθώς και ο τρόπος που αυτά αναπαρίστανται μαθηματικά. Στη συνέχεια, εξετάστηκε διεξοδικά η διαδικασία της συνέλιξης πινάκων και παρουσιάστηκαν λεπτομερώς τα διάφορα επίπεδα των ΣΝΔ. Ακόμα, αναλύθηκε η σημασία της μεταφοράς μάθησης στην εκπαίδευση μοντέλων για μικρά σύνολα δεδομένων και παρουσιάστηκε αναλυτικά η πιο απλή δομή ενός ΣΝΔ (LeNet-5) και των αρχιτεκτονικών που αυτή ενέπνευσε. Τέλος, πραγματοποιήθηκε εκτενής ανάλυση διαφόρων αρχιτεκτονικών μοντέλων ΣΝΔ, όπως είναι το ResNet και το NASNet, τα οποία θα εξεταστούν πειραματικά στα επόμενα κεφάλαια προκειμένου να αναδειχθεί και να αξιολογηθεί η απόδοσή τους στο διαθέσιμο σύνολο δεδομένων.

ΚΕΦΑΛΑΙΟ 2^ο : Ανάλυση δεδομένων και κατασκευή μοντέλου

Προκειμένου να αναπτυχτεί ένα μοντέλο βαθιάς μάθησης που θα μπορεί να είναι σε θέση να διακρίνει τους όγκους του μαστού και να τους κατηγοριοποιεί ως καλοήθεις ή ως κακοήθεις με μεγάλη ακρίβεια, θα πρέπει να ακολουθηθεί μια συγκεκριμένη διαδικασία. Στο κεφάλαιο αυτό αρχικά, θα παρουσιαστούν οι τεχνολογίες καθώς και τα εργαλεία λογισμικού και εξοπλισμού που θα χρησιμοποιηθούν για την ανάπτυξη των μοντέλων. Στη συνέχεια, θα αναλυθούν οι μέθοδοι προεπεξεργασίας και τροποποίησης των δεδομένων, με στόχο την επέκταση του διαθέσιμου συνόλου δεδομένων και τη βελτιστοποίηση των προβλέψεων καθώς και της συνολικής απόδοσης. Αφού ολοκληρωθεί η κανονικοποίηση του επεκταμένου συνόλου δεδομένων, αυτά θα τροφοδοτηθούν στα προεκπαιδευμένα μοντέλα που επιλέχθηκαν για την ακρίβεια τους και των οποίων οι υπερπαραμέτροι θα ρυθμιστούν με στόχο τη βελτιστοποίηση των αποτελεσμάτων. Τέλος, τα μοντέλα θα συγκριθούν ως προς την ακρίβεια τους και ως προς κάποιους άλλους δείκτες μέτρησης, ώστε να επιλεγεί το καλύτερο από αυτά, το οποίο και θα ενσωματωθεί στην διαδικτυακή εφαρμογή που θα αναπτυχθεί στο επόμενο κεφάλαιο.

2.1 Τεχνολογίες Μοντέλου

2.1.1 Εργαλεία ανάπτυξης λογισμικού

Η ανάπτυξη της TN τα τελευταία χρόνια έχει κάνει χωρίς αμφιβολία κάποιες γλώσσες προγραμματισμού και κάποια εργαλεία μηχανικής μάθησης να ξεχωρίσουν στον χώρο αυτό. Μία από τις γλώσσες προγραμματισμού που χρησιμοποιείται ευρέως στην ανάλυση δεδομένων και στην υπολογιστική μάθηση είναι η δημοφιλής Python. Η Python είναι μια γλώσσα προγραμματισμού γενικής χρήσης, με την οποία μπορεί να υλοποιηθεί κάθε είδους εφαρμογή με σχετική ευκολία, ακόμη και αν ο σχεδιαστής της εφαρμογής δεν είναι πολύ εξοικειωμένος με τον προγραμματισμό. Στην περίπτωση μας, όλα τα μοντέλα που θα υλοποιηθούν θα γίνουν με τη χρήση Python μέσω μιας πολύ διαδεδομένης πλατφόρμας στην επιστήμη των δεδομένων, γνωστής με το όνομα Jupyter Notebook. Το Jupyter Notebook είναι ένα περιβάλλον προγραμματισμού που επιτρέπει τη συνδυασμένη χρήση κελιών κώδικα, καθιστώντας τη διαδικασία ανάπτυξης, ανάλυσης και παρουσίασης των δεδομένων πιο διαδραστική.

Αξίζει να γίνει επιπλέον μια μικρή αναφορά και σε κάποιες από τις βασικές βιβλιοθήκες που θα χρησιμοποιηθούν κατά τη διάρκεια της προεπεξεργασίας των δεδομένων αλλά και της ανάπτυξης του μοντέλου:

- TensorFlow: Η TensorFlow είναι μια βιβλιοθήκη ανοιχτού κώδικα που αναπτύχθηκε από τους μηχανικούς της Google Brain, με σκοπό την ανάπτυξη συστημάτων μηχανικής και βαθιάς μάθησης. Το στοιχείο που την κάνει να ξεχωρίζει είναι ο τρόπος με τον οποίο πραγματοποιεί υπολογισμούς πολυδιάστατων πινάκων χρησιμοποιώντας ταυνοστές (tensors), καθώς και η ικανότητα της βιβλιοθήκης να τρέχει υπολογισμούς σε διαφορετικά μέρη του υπολογιστή, όπως στην CPU ή στην GPU.
- Keras: Η Keras είναι μια υψηλού επιπέδου βιβλιοθήκη ΝΔ που είναι ενσωματωμένη στο TensorFlow. Μέσω της Keras, η ανάπτυξη μοντέλων μηχανικής μάθησης πραγματοποιείται με τη χρήση λίγων γραμμών κώδικα, επιτρέποντας στον χρήστη να επικεντρωθεί περισσότερο στη δομή και τη λειτουργία του μοντέλου παρά στην υλοποίηση αυτού από το μηδέν.

- NumPy: Η NumPy είναι μια από τις πιο ισχυρές βιβλιοθήκες της Python καθώς χρησιμοποιείται ευρέως για την ανάλυση και επεξεργασία δεδομένων λόγω του αποτελεσματικού τρόπου με τον οποίο πραγματοποιεί πράξεις σε πολυδιάστατους πίνακες.
- Matplotlib: Το Matplotlib είναι η πιο γνωστή βιβλιοθήκη για τη δημιουργία γραφημάτων καθώς και για την οπτικοποίηση και παρουσίαση δεδομένων και αποτελεσμάτων με ευκολία.

2.1.2 Υπολογιστικό περιβάλλον

Το περιβάλλον στο οποίο θα αναπτυχθεί και θα εκτελεστεί το μοντέλο, για την ελαχιστοποίηση του χρόνου εκτέλεσης και την παραγωγή βέλτιστων αποτελεσμάτων, δεν θα μπορούσε να είναι άλλο από ένα σύστημα υψηλών προδιαγραφών. Ο υπερυπολογιστής NVIDIA DGX Workstation αποτελεί ένα τέτοιο σύστημα, προσαρμοσμένο στις ανάγκες της ανάπτυξης και της έρευνας συστημάτων TN. Διαθέτει 4 κάρτες γραφικών Tesla V100 Tensor Core, γεγονός που το καθιστά ιδανικό για την αποτελεσματική εκπαίδευση μοντέλων βαθιάς μάθησης. Το σύστημα διαθέτει επιπλέον 4 NVLink, τα οποία αποτελούν ειδικούς διαύλους επικοινωνίας μεταξύ των GPU. Κάθε ένας παρέχει bandwidth έως 50GBps για την ανταλλαγή πληροφοριών μεταξύ των GPU. Επιπλέον, η μνήμη RAM του workstation είναι διπλάσια από την μνήμη των GPU, βελτιώνοντας τη διαχείριση των προσωρινών δεδομένων κατά την εκπαίδευση μοντέλων βαθιάς μάθησης. Πιο συγκριμένα τα χαρακτηριστικά του υπερυπολογιστή είναι:

- Επεξεργαστές: Intel Xeon E5-2698 v4 (2.2 GHz, 20 πυρήνες)
- Κάρτες Γραφικών: 4x NVIDIA Tesla V100 32GB, 500 TFLOPS με 20480 CUDA Cores και 2560 Tensor Cores
- Μνήμη: 256GB RDIMM DDR4
- Αποθηκευτικός Χώρος: 3x 1.92TB SSD (Data) + 1x 1.92TB SSD (OS)



Εικόνα 23: Ο υπολογιστής υψηλών προδιαγραφών NVIDIA DGX Workstation

2.2 Επισκόπηση προηγούμενης έρευνας

Η αναγνώριση και η διάγνωση διαφόρων περιστατικών καρκίνου του μαστού μέσω μοντέλων TN δεν αποτελεί πρόσφατη εξέλιξη στον επιστημονικό τομέα, καθώς είναι ένας κλάδος στον οποίο οι έρευνες αυξάνονται συνεχώς. Προτού προχωρήσουμε λοιπόν στην προεπεξεργασία και την ανάλυση των δεδομένων, αξίζει να γίνει μια σύντομη επισκόπηση και να αναφερθούν οι πιο γνωστές προηγούμενες έρευνες που έχουν πραγματοποιηθεί.

Οι περισσότερες έρευνες που έχουν γίνει γύρω από τη διάγνωση παθήσεων του μαστού μέσω εικόνων έχουν επικεντρωθεί κυρίως στη χρήση στιγμιότυπων από μαστογραφίες και τη φόρτωση αυτών σε μοντέλα μεγάλης ακρίβειας, είτε για την εξαγωγή χρήσιμων χαρακτηριστικών και δεδομένων από την εικόνα, όπως είναι η μάσκα βάσης του όγκου, είτε για την κατηγοριοποίηση των εικόνων αυτούσιες ως καλοήθειες ή κακοήθειες [19]. Παρόλα αυτά, εκτός του γεγονότος ότι κάποια νοσοκομεία ή ιατρικά κέντρα μπορεί να μην διαθέτουν τον απαραίτητο εξοπλισμό για την πραγματοποίηση ψηφιακής μαστογραφίας, σε μια μαστογραφία ενδέχεται συχνά, σε περίπτωση πυκνότερων ιστών στο στήθος, να μην μπορεί να διακριθεί ένας όγκος. Η μέθοδος που χρησιμοποιείται συχνά συμπληρωματικά στη μαστογραφία ή ως υποκατάστατό αυτής, είναι το υπερηχογράφημα μαστού, το οποίο αποτελεί μια πολύ πιο θορυβώδη ως προς την παραγόμενη εικόνα, αλλά εξαιρετικά ασφαλή μέθοδο διάγνωσης που χρησιμοποιούν οι ακτινολόγοι [20].

Γενικά, έχουν δημοσιευθεί πολλές εργασίες που εξετάζουν τη χρήση μεθόδων βαθιάς μάθησης για την αναγνώριση όγκων από εικόνες υπερήχων. Το 2017 οι M. Yar et al. [21] εκπαίδευσαν διάφορα ΝΔ για τον ακριβή εντοπισμό όγκων και κατέληξαν στο συμπέρασμα ότι τα μοντέλα LeNet και AlexNet με μεταφορά μάθησης πέτυχαν τα καλύτερα αποτελέσματα στα δεδομένα τους. Σε επόμενη δουλειά το 2020, οι Fang et al. [22] πραγματοποίησαν προεπεξεργασία των δεδομένων, εξάγοντας χαρακτηριστικά όπως το σχήμα και τα όρια των όγκων, τα οποία στη συνέχεια χρησιμοποίησαν σε ένα μοντέλο SVM για την πρόβλεψη και την κατηγοριοποίηση. Αξιοσημείωτη είναι και η εργασία των Alzubaidi et al. [23], στην οποία μελετήθηκε το πώς η ρύθμιση των υπερπαραμέτρων προεκπαιδευμένων μοντέλων με τελείως διαφορετικά σύνολα δεδομένων μπορεί να βελτιώσει σημαντικά την απόδοση ενός μοντέλου.

Όσον αφορά την υλοποίηση ενός μοντέλου χρησιμοποιώντας το ίδιο σύνολο δεδομένων που θα χρησιμοποιηθεί στη συνέχεια, υπάρχει μια πληθώρα προεκπαιδευμένων και μη δημοσιευμένων μοντέλων διαθέσιμα στην ιστοσελίδα που φιλοξενεί το συγκεκριμένο σύνολο δεδομένων⁶. Από το δημοσιευμένο υλικό που αφορά τη δυαδική κατηγοριοποίηση ή την αναγνώριση αντικειμένων με χρήση εικόνων υπερήχων του μαστού, έχουν προκύψει σημαντικά ευρήματα με την πάροδο των ετών. Το 2019, οι W. Al-Dhabyani et al. [18], δημιουργοί του συνόλου δεδομένων BUSI, δημοσίευσαν μια εργασία που ασχολούνταν με τη σύγκριση μοντέλων στο συγκεκριμένο σύνολο δεδομένων. Εκτός από την παραδοσιακή αύξηση δεδομένων, χρησιμοποίησαν και ένα παραγωγικό αντιπαραθετικό δίκτυο (DAGAN), το οποίο μέσω ενός generator και ενός discriminator παράγει φωτογραφίες βάσει αυτών που έχει τροφοδοτηθεί, αυξάνοντας έτσι τα διαθέσιμα δεδομένα. Στόχος ήταν μέσω αυτής της επαύξησης των δεδομένων να βελτιωθεί η ακρίβεια των μοντέλων. Τελικό συμπέρασμα ήταν ότι η αύξηση των δεδομένων με τον συνδυασμό των δύο μεθόδων παρήγαγε πολύ καλύτερα αποτελέσματα από τη χρήση είτε της μιας είτε της άλλης μεθόδου. Το 2022, οι R.C.

⁶ <https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset/code> [Accessed: 12 July 2024]

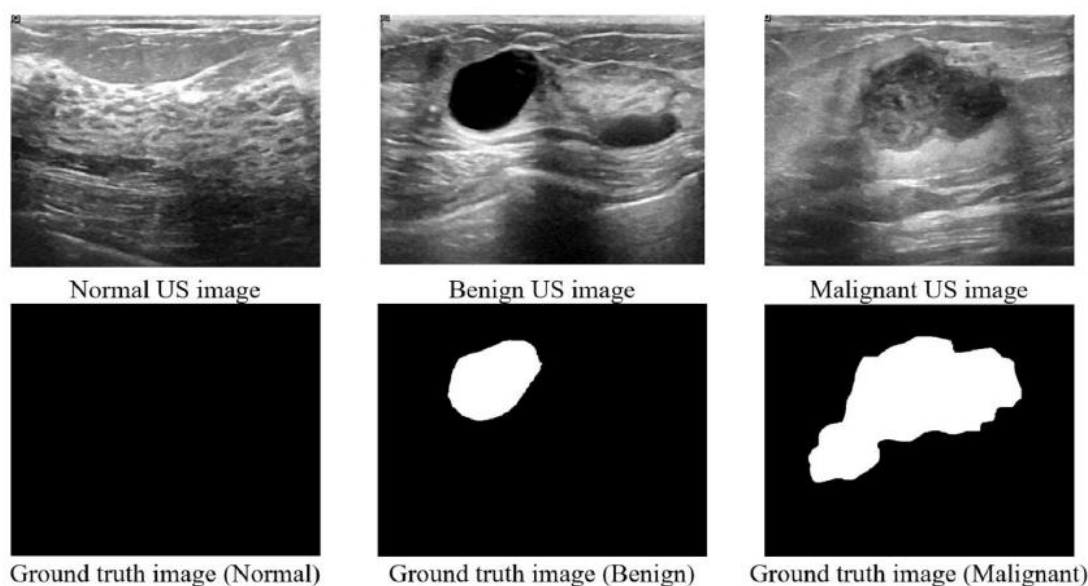
Joshi et al. [24] ανέδειξαν τη συμβολή της επαύξησης του διαθέσιμου συνόλου δεδομένων στην αντιμετώπιση του προβλήματος των ψευδώς θετικών προβλέψεων, μειώνοντάς τις και οδηγώντας σε εμφανή αύξηση της ακρίβειας των μοντέλων. Μια ακόμη πρόσφατη εργασία πάνω στο σύνολο δεδομένων BUSI είναι αυτή των S. Afrifa et al. [25], οι οποίοι έκαναν σύγκριση μη επιβλεπόμενων αλγορίθμων μάθησης που εξήγαγαν χαρακτηριστικά μετατρέποντας τις ασπρόμαυρες εικόνες σε RGB, και επιβλεπόμενων αλγορίθμων. Κατέληξαν στο συμπέρασμα ότι οι επιβλεπόμενες μέθοδοι μάθησης, και κυρίως τα ΣΝΔ, υπερίσχυσαν των πιο παραδοσιακών αλγορίθμων μηχανικής μάθησης όπως είναι το K-Nearest Neighbor.

Οι εφαρμογές που έχουν υλοποιηθεί με την πάροδο των ετών επικεντρώνονται κυρίως σε εικόνες μαστογραφίας ή ιστοπαθολογίας, λόγω των μεγάλων ανοιχτών και εύκολα προσβάσιμων από όλους διαθέσιμων συνόλων δεδομένων. Ένα τέτοιο παράδειγμα αποτελεί η ΔΕ της M. Bouhadda [26], η οποία χρησιμοποιεί έναν διακομιστή Flask και ένα απλό περιβάλλον διεπαφής με HTML και CSS για να δημιουργήσει μια διαδικτυακή εφαρμογή που μέσω ενός μοντέλου κατηγοριοποιεί όγκους από ιστοπαθολογικές εικόνες. Άλλη μια εφαρμογή που χρησιμοποιεί ιστοπαθολογικές εικόνες για να προβλέψει αν ένας όγκος είναι καλοήθης ή κακοήθης είναι το ABCanDroid που αναπτύχθηκε από τους D Chowdhury et al. [27], μια εφαρμογή για κινητά που είναι γραμμένη στη γλώσσα Flutter και ο διακομιστής της φιλοξενείται στην γνωστή πλατφόρμα Heroku. Αναφέρεται επιπλέον πως υπάρχει μια πληθώρα εφαρμογών που βασίζονται αποκλειστικά σε αριθμητικά ή δυαδικά δεδομένα, όπως η εφαρμογή που δημιούργησε η Shital Pawar [28], ωστόσο παραμένει αμφίβολο το πόσο φιλικές προς τον χρήστη είναι τέτοιου είδους εφαρμογές.

2.3 Προεπεξεργασία δεδομένων

2.3.1 Διαθέσιμο σύνολο δεδομένων

Το σύνολο δεδομένων που θα χρησιμοποιηθεί είναι το Dataset BUSI το οποίο περιέχει εικόνες υπερήχων μαστού, και σκοπός του είναι να χρησιμοποιηθεί από ερευνητές για να αναπτυχθούν μοντέλα που θα πραγματοποιούν ανίχνευση και ταξινόμηση όγκων του μαστού. Το σύνολο δεδομένων περιλαμβάνει 780 εικόνες, οι οποίες κατατάσσονται σε τρεις κατηγορίες και εν συνεχεία σε τρεις φακέλους: φυσιολογικές (normal), καλοήθειες (benign) και κακοήθειες (malignant). Οι εικόνες συλλέχθηκαν από το νοσοκομείο Baheya στο Κάιρο στην Αίγυπτο για την πρώτη ανίχνευση και θεραπεία του καρκίνου του μαστού, με τη χρήση των συστημάτων υπερήχων LOGIQ E9 και LOGIQ E9 Agile. Η συλλογή των δεδομένων έγινε το 2018, από 600 γυναίκες ηλικίας 25 έως 75 ετών και το μέσο μέγεθος εικόνας είναι 500x500 pixels. Αρχικά συλλέχθηκαν 1100 εικόνες, αλλά μετά από επεξεργασία του συνόλου δεδομένων και διαγραφή διπλότυπων φωτογραφιών, ο αριθμός μειώθηκε σε 133 φυσιολογικές εικόνες μαστού, 437 και 210 εικόνες καλοήθων και κακοήθων όγκων αντίστοιχα. Οι εικόνες υποβλήθηκαν επιπλέον σε περεταίρω επεξεργασία για την αφαίρεση περιττών πληροφοριών που ενδέχεται να επηρέαζαν τα αποτελέσματα σε μια διαδικασία εκπαίδευσης και μετατράπηκαν από μορφή DICOM σε μορφή PNG. Κάθε εικόνα μέσα στον εκάστοτε φάκελο συνοδεύεται από την αντίστοιχη ή τις αντίστοιχες μάσκες της (ground truth images), η οποίες δημιουργήθηκαν με την χρήση ενός εργαλείου στο Matlab [17].



Εικόνα 24: Δείγματα εικόνων υπερήχων του μαστού από το σύνολο δεδομένων και της αντίστοιχης μάσκα τους [17]

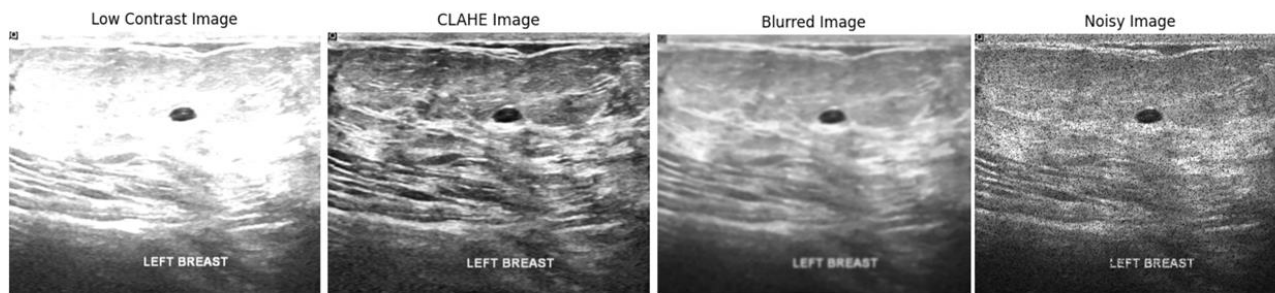
Για το πρόβλημα μας, όπου θα πραγματοποιηθεί δυαδική ταξινόμηση με στόχο τη διάκριση των όγκων σε καλοήθειες και κακοήθειες, είναι απαραίτητο να αφαιρεθεί ο φάκελος των φυσιολογικών υπερήχων μαστού. Μετά την αφαίρεση αυτή, θα ακολουθήσει μια διαδικασία αύξησης και μετασχηματισμού των δεδομένων, προκειμένου να δημιουργηθεί ένα νέο, ανανεωμένο και εκτενές σύνολο δεδομένων, το οποίο θα χρησιμοποιηθεί για την εκπαίδευση μοντέλων βαθιάς μάθησης, συγκεκριμένα των VGG-16, ResNet και NASNet. Κατά την εκπαίδευση, θα συγκριθούν οι επιδόσεις και η ακρίβεια των μοντέλων αυτών, ώστε να επιλεγεί το καταλληλότερο για την τελική εφαρμογή.

2.3.2 Επαύξηση συνόλου δεδομένων

Η έλλειψη δεδομένων, ιδίως στον τομέα της οπτικής αναγνώρισης, έχει ωθήσει τους ερευνητές και τους επιστήμονες να επιδιώξουν τρόπους διεύρυνσης των διαθέσιμων συνόλων δεδομένων. Οι μέθοδοι που χρησιμοποιούνται για να επιτευχθεί αυτό διαφέρουν, αλλά όλες έχουν εμφανή επίδραση στη βελτίωση της ακρίβειας ενός μοντέλου. Η περιστροφή των εικόνων σε διάφορες γωνίες αποτελεί μια από τις πιο απλές μεθόδους, όμως παρόμοια αποτελέσματα παρέχει και ο καθρεπτισμός (οριζόντια ή κάθετα) που αποτελεί και τον πιο δημοφιλή τρόπο επαύξησης δεδομένων. Επιπλέον, η τυχαία περικοπή των εικόνων είναι μια μέθοδος που χρησιμοποιείται συχνά στην ανίχνευση αντικειμένων (Object Detection), ενώ η προσθήκη θορύβου μπορεί επίσης να συνεισφέρει στην βελτίωση της απόδοσης ενός μοντέλου στην αναγνώριση εικόνων. Για εικόνες RGB, συνηθίζεται επίσης η μέθοδος Color Shifting, όπου προστίθεται ή αφαιρείται ένας συγκεκριμένος αριθμός στα τα τρία κανάλια χρωμάτων, προκειμένου να δημιουργηθεί μια εικόνα με ελαφρώς διαφορετικά χρώματα, η οποία μπορεί να χρησιμοποιηθεί στην εκπαίδευση ενός μοντέλου. Αν και απλές στην εφαρμογή τους, οι παραπάνω μέθοδοι, μαζί με κάποιες λιγότερο διαδεδομένες όπως η τοπική παραμόρφωση και το shearing εικόνας, συμβάλλουν ουσιαστικά στην αύξηση της ποικιλίας στα δεδομένα και εν συνεχεία στη βελτίωση της απόδοσης ενός μοντέλου στον τομέα της οπτικής αναγνώρισης.

Προτού ξεκινήσει η διαδικασία αύξησης των δεδομένων, είναι απαραίτητο να διαγραφούν όλες οι μάσκες που περιέχονται σε κάθε φάκελο, είτε πρόκειται για καλοήθεις είτε για κακοήθεις περιπτώσεις. Η χρήση της μάσκας για την επικάλυψη στις εικόνες των υπερήχων είναι μια διαδικασία που δεν θα εξεταστεί στην διπλωματική γιατί δεν έχει ακόμα διαπιστωθεί αν μπορεί να προσφέρει μεγαλύτερη ακρίβεια ή αν μπερδεύει περισσότερο τους αλγορίθμους στην αναγνώριση των όγκων. Αφού ολοκληρωθεί και αυτό, το επόμενο βήμα είναι να πραγματοποιηθεί καθρεπτισμός των εικόνων ως προς τον κάθετο άξονα και να αποθηκευτούν οι νέες εικόνες. Μόλις ολοκληρωθεί ο καθρεπτισμός ως προς τον άξονα y, θα υπάρχουν πλέον 874 εικόνες καλοήθων και 420 εικόνες κακοήθων όγκων. Παρατηρείται ότι το σύνολο δεδομένων από την αρχή είχε ανισορροπία της τάξης του 70-30, οπότε τονίζεται ότι είναι σημαντικό να δοθεί προσοχή στην εξισορρόπησή του πριν την εκπαίδευση των αλγορίθμων, καθώς αυτό μπορεί να επηρεάσει σε μεγάλο βαθμό την ακρίβεια των προβλέψεων του μοντέλου για κάθε κλάση.

Για την περαιτέρω διεύρυνση του συνόλου δεδομένων, θα χρησιμοποιηθούν διάφορες μέθοδοι αλλοίωσης της εικόνας που παρατηρούνται και στον χώρο της ακτινοδιαγνωστικής. Τα πιο κοινά, αν και σχετικά σπάνια, είδη σφαλμάτων που μπορεί να παρατηρηθούν σε μια διαγνωστική εικόνα είναι οι θολές εικόνες, οι εικόνες με μεγάλη ή μικρή αντίθεση και οι παραμορφωμένες από θόρυβο εικόνες. Αυτά τα είδη αλλοιώσεων συχνά μπορεί να επηρεάσουν την κρίση των γιατρών, οδηγώντας σε λανθασμένες διαγνώσεις ή άλλες ανακρίβειες στη διάγνωση [29]. Για να βελτιώσουμε την ακρίβεια του μοντέλου μας, είναι σημαντικό να εκπαιδεύσουμε τους αλγορίθμους χρησιμοποιώντας αυτές τις αλλοιωμένες εικόνες. Με τον τρόπο αυτό διασφαλίζουμε ότι το μοντέλο θα μάθει να αναγνωρίζει και να επεξεργάζεται εικόνες που περιέχουν σφάλματα, μειώνοντας έτσι την πιθανότητα λανθασμένων προβλέψεων όταν αυτό έρθει αντιμέτωπο με παρόμοια προβλήματα σε πραγματικές συνθήκες.



Εικόνα 25: Οι διάφοροι τύποι παραμόρφωσης για την διεύρυνση του συνόλου δεδομένων των υπερήχων

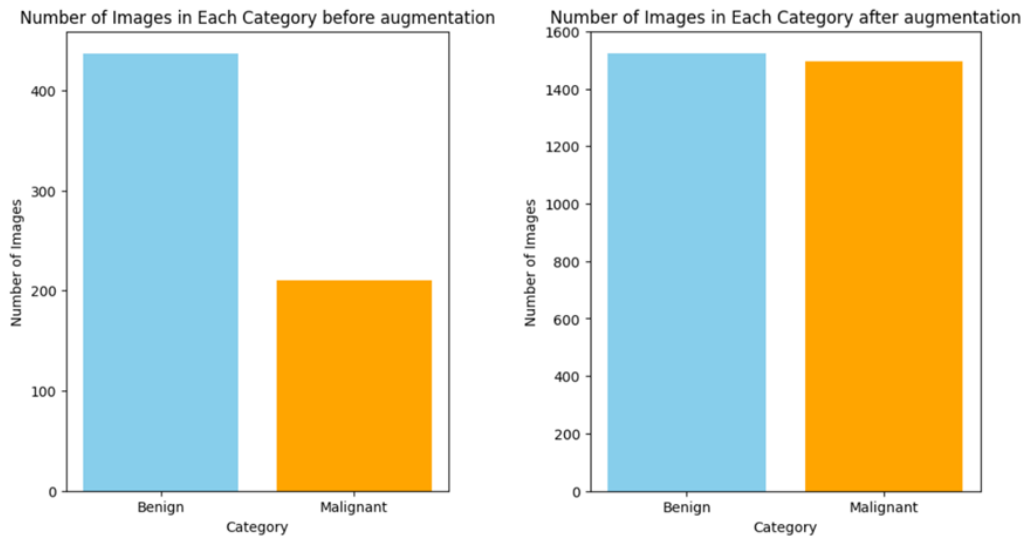
Το πρώτο βήμα για τη διεύρυνση του συνόλου δεδομένων με τις τεχνικές που περιγράφηκαν παραπάνω είναι να οριστεί ένας στόχος όσον αφορά τον αριθμό των διαθέσιμων εικόνων. Οι συνολικές εικόνες μετά τον καθρεπτισμό ανέρχονται σε 1.294 εικόνες υπερήχων μαστού οπότε ο στόχος είναι να διευρυνθεί αυτό το σύνολο περίπου έως τον αριθμό των 3.000 εικόνων. Είναι γνωστό ότι η ανισορροπία ενός συνόλου δεδομένων μπορεί να είναι επιζήμια για την απόδοση ενός μοντέλου που πραγματοποιεί αναγνώριση προτύπων και ταξινόμηση [30]. Συνεπώς, είναι σημαντικό να υπάρχει ένα όσο το δυνατόν πιο ισορροπημένο σύνολο δεδομένων. Δεδομένου ότι χρειαζόμαστε 1.500 εικόνες για κάθε κατηγορία και οι διαθέσιμες εικόνες είναι 874 για την κατηγορία των καλοήθων όγκων και 420 για την κατηγορία των κακοήθων όγκων, θα χρειαστεί να διευρύνουμε το σύνολο των καλοήθων όγκων κατά περίπου 650 φωτογραφίες και το σύνολο των κακοήθων όγκων κατά περίπου 1.100 φωτογραφίες. Κατά την εκτέλεση της διεύρυνσης, θα εφαρμοστούν τυχαίες αλλαγές στις φωτογραφίες του συνόλου δεδομένων με πρώτη την αναγκαία μεγέθυνση κάποιων τυχαίων σημείων σε ένα ποσοστό των εικόνων. Επιπλέον θα εφαρμοστεί αλλαγή της αντίθεσης, βελτίωση του φωτισμού των σκοτεινών σημείων, θόλωση της φωτογραφίας ή προσθήκη θορύβου, με πιθανότητα 25% για κάθε μία από αυτές τις τροποποιήσεις.

Για την κατηγορία των καλοήθων όγκων, αρχικά θα επιλεγούν τυχαία 150 εικόνες για μεγέθυνση με zoom factor ίσο με 1.2. Στη συνέχεια, προκειμένου να φτάσουμε τον αριθμό των περίπου 2000 εικόνων, προβλέπεται η εφαρμογή διαφόρων τεχνικών. Συγκεκριμένα, θα πραγματοποιηθεί μείωση του κόντραστ σε 125 τυχαίες εικόνες, βελτίωση του φωτισμού των σκοτεινών σημείων (CLAHE) σε 125 άλλες, εφαρμογή Gaussian Blur σε 125 εικόνες και προσθήκη θορύβου (Salt and Pepper) σε άλλες 125 φωτογραφίες. Αντίστοιχα, για τις εικόνες κακοήθων όγκων, λόγω του μικρού αρχικού συνόλου, θα επιλεγούν 315 εικόνες για μεγέθυνση με zoom factor 1.2 και αυτή την φορά από τις 420 ήδη υπάρχουσες. Στη συνέχεια, αυτές οι εικόνες θα επεξεργαστούν χρησιμοποιώντας τις ίδιες τεχνικές, αυξάνοντας τις φωτογραφίες κατά 190 σε κάθε βήμα. Ο Πίνακας 1 περιγράφει λεπτομερώς τη διαδικασία και παρέχει τον συνολικό αριθμό δεδομένων σε κάθε βήμα επαύξησης:

Πίνακας 1: Διαδικασία επαύξησης και συνολικός αριθμός δεδομένων μετά από κάθε βήμα

Κατηγορία	Αρχικός αριθμός εικόνων (πριν την επαύξηση)	Μετά τον Καθρεπτισμό	Μετά την Μεγέθυνση	Μετά την παραμόρφωση εικόνων (μείωση contrast, CLAHE, Gaussian Blur, Salt and Pepper Noise)
Benign	437	874	1024	1524
Malignant	210	420	735	1495
Σύνολο εικόνων	647	1294	1759	3019

Τελικά, το σύνολο δεδομένων θα περιέχει όλες τις εικόνες από το αρχικό σύνολο δεδομένων που βρέθηκε στο διαδίκτυο, αλλά θα περιλαμβάνει και φωτογραφίες που έχουν υποστεί τις τεχνικές προεπεξεργασίας που αναφέρθηκαν. Σκοπός ήταν το σύνολο δεδομένων από συνολικά 647 φωτογραφίες να επεκταθεί περίπου στις 3.000 εικόνες, διασφαλίζοντας την ισορροπία των κατηγοριών και την επίτευξη όσο το δυνατόν καλύτερης ακρίβειας. Στην Εικόνα 26 παρατηρείται αυτή η ανακατανομή των κλάσεων του συνόλου δεδομένων μετά την επαύξηση των εικόνων.



Εικόνα 26: Αριθμός των εικόνων πριν και μετά την αύξηση του συνόλου δεδομένων χρησιμοποιώντας τεχνικές παραδοσιακής επαύξησης δεδομένων για την εξισορρόπηση αυτού.

2.3.3 Μετασχηματισμός δεδομένων

Το τελευταίο βήμα που πρέπει να γίνει για έτσι ώστε τα δεδομένα να τροφοδοτηθούν στους αλγορίθμους εξόρυξης δεδομένων είναι ο μετασχηματισμός. Η διαδικασία μετασχηματισμού είναι απαραίτητη για τη βελτίωση της ταχύτητας, της ακρίβειας, καθώς και της αποτελεσματικότερης εξαγωγής χαρακτηριστικών από έναν αλγόριθμο, όπως είναι το ΝΔ στην δική μας περίπτωση. Παρά την πληθώρα δεδομένων και τις διαδικασίες επαύξησης και καθαρισμού που μπορεί να έχουν προηγηθεί, τα δεδομένα ενδέχεται να μην είναι επαρκή για την εξαγωγή χρήσιμων πληροφοριών ή, ακόμη χειρότερα, να δυσχεραίνουν την εκπαίδευση του αλγορίθμου σε κάθε επανάληψη αντί να την διευκολύνουν [31].

Στην παρούσα περίπτωση, όπου πραγματοποιείται αναγνώριση εικόνων και δυαδική κατηγοριοποίηση, ο μετασχηματισμός των δεδομένων είναι σχετικά απλός. Το πρώτο στάδιο περιλαμβάνει τον καθορισμό των διαστάσεων όλων των εικόνων που θα εισαχθούν στο δίκτυο (resize). Αυτό το στάδιο διασφαλίζει ότι όλες οι εικόνες έχουν το ίδιο μέγεθος όταν τροφοδοτούνται στο ΣΝΔ που θα αναλάβει την αναγνώριση. Τα μοντέλα που θα δοκιμαστούν απαιτούν διαφορετικά μεγέθη εικόνων για τη σωστή λειτουργία τους, οπότε οι κατάλληλες διαστάσεις θα καθορίζονται κάθε φορά ανάλογα με τον αλγόριθμο που χρησιμοποιείται. Το δεύτερο στάδιο στη διαδικασία προετοιμασίας των δεδομένων είναι η αλλαγή κλίμακας (scaling) των τιμών των pixels όλων των εικόνων. Αυτό το βήμα περιλαμβάνει τη μετατροπή των τιμών των pixels στο εύρος $[0, 1]$, διαιρώντας κάθε τιμή με τη μέγιστη δυνατή τιμή pixel (255). Η κανονικοποίηση με αυτόν τον τρόπο εξασφαλίζει ότι όλες οι τιμές των pixels βρίσκονται εντός του ίδιου εύρους, αποτρέποντας έτσι τη δημιουργία ανισορροπιών που μπορεί να επηρεάσουν αρνητικά την εκπαίδευση του μοντέλου. Τέλος, τα δεδομένα πρέπει να χωριστούν σε σύνολα εκπαίδευσης (training set), σύνολα επαλήθευσης (validation set) και σύνολα δοκιμής (test set). Ο διαχωρισμός αυτός είναι απαραίτητος ώστε το μοντέλο να εκπαιδευτεί με ένα σύνολο δεδομένων και να αξιολογηθεί η απόδοσή του με ένα διαφορετικό, ανεξάρτητο σύνολο, διασφαλίζοντας ότι το μοντέλο μπορεί να γενικεύσει και να αποδώσει καλά σε νέα, άγνωστα δεδομένα που θα εισάγει ο χρήστης.

Αξίζει να σημειωθεί ότι οι μέθοδοι κανονικοποίησης, όπως περιγράφηκαν παραπάνω, παίζουν καθοριστικό ρόλο στην αποφυγή προβλημάτων όπως το φαινόμενο *vanishing/exploding gradients* [13], που αναφέρθηκε σε προηγούμενο υποκεφάλαιο, και μπορεί να παρουσιαστεί κατά τη διάρκεια της εκπαίδευσης.

2.4 Εκπαίδευση Νευρωνικών Δικτύων

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, θα εκπαιδευτούν πάνω στο ίδιο σύνολο δεδομένων τρία διαφορετικά ΣΝΔ, των οποίων τα επίπεδα θα προσαρμοστούν με τέτοιο τρόπο ώστε να πραγματοποιούν δυαδική κατηγοριοποίηση. Αφού συγκριθούν με τα ίδια κριτήρια, θα επιλεγεί αυτό με την καλύτερη απόδοση. Τα μοντέλα αυτά θα είναι το VGG16, το ResNet και το NASNet, τρεις αρχιτεκτονικές που έχουν ήδη αναλυθεί.

Το πρώτο βήμα πριν την εκπαίδευση των δικτύων είναι να χωριστούν τα δεδομένα στα σύνολα εκπαίδευσης (2112 εικόνες), επαλήθευσης (452 εικόνες) και δοκιμής (455 εικόνες). Στη συνέχεια, θα γίνει η κανονικοποίηση τους, όπως περιγράφηκε προηγουμένως, και ο ορισμός του μεγέθους της εικόνας χρησιμοποιώντας τη βιβλιοθήκη TensorFlow. Αφού ολοκληρωθεί η κανονικοποίηση των δεδομένων και οι εικόνες μετατραπούν στις κατάλληλες διαστάσεις που δέχεται το κάθε μοντέλο, το εκάστοτε μοντέλο φορτώνεται ως μοντέλο βάσης από τα προεκπαιδευμένα μοντέλα που παρέχει η TensorFlow. Με αυτόν τον τρόπο, εισάγονται σαν μοντέλα βάσης το VGG16, το ResNet50 και το NASNetLarge. Η επιλογή των βαρών που έχουν εκπαιδευτεί στο σύνολο δεδομένων ImageNet είναι η καλύτερη επιλογή [12][14][16]. Έχουμε επίσης τη δυνατότητα να αποφασίσουμε αν θα συμπεριλάβουμε το Top Layer στο μοντέλο μας ή αν θα το προσαρμόσουμε ανάλογα με την περίπτωση. Στην συγκεκριμένη περίπτωση, το Top Layer δεν θα συμπεριληφθεί, καθώς χρειάζεται να πραγματοποιηθεί δυαδική κατηγοριοποίηση.

Στη συνέχεια, ορίζονται τα τελευταία επίπεδα πριν την έξοδο του ΝΔ. Αρχικά, για το VGG16, ο πολυδιάστατος πίνακας που έχει παραχθεί από τα πρώτα επίπεδα του συνελικτικού δικτύου μετατρέπεται σε μονοδιάστατο, προετοιμάζοντάς τον για το πλήρως συνδεδεμένο επίπεδο, το οποίο αποτελείται από 256 νευρώνες με ΣΕ ReLU. Για τα δίκτυα ResNet και NASNet, αρκεί να προστεθεί ένα επίπεδο υποδειγματοληψίας καθώς και ένα πλήρως συνδεδεμένο επίπεδο, το οποίο αποτελείται από 1024 νευρώνες με ΣΕ ReLU. Τέλος, σε όλα τα μοντέλα προσθέτουμε το επίπεδο εξόδου, το οποίο αποτελείται από έναν νευρώνα με σιγμοειδή ΣΕ. Αυτό μετατρέπει την έξοδο σε μια τιμή μεταξύ 0 και 1, που μπορεί να ερμηνευτεί ως η πιθανότητα η εικόνα να ανήκει στη μία ή στην άλλη κατηγορία. Τα επίπεδα καθώς και οι υπερπαράμετροι των δικτύων παρουσιάζονται αναλυτικότερα στον Πίνακα 2.

Πίνακας 2: Επίπεδα και υπερπαράμετροι δικτύων

Model	Image size	Output Layers	Optimizer	Learning Rate	Epochs
VGG16	224x224	Flatten + Dense(256 with Relu) + Sigmoid	Adam	10^{-3}	10
ResNet50	224x224	GlobalAveragePooling2D + Dense(1024 with Relu) + Sigmoid	Adam	10^{-3}	10
NASNet	331x331	GlobalAveragePooling2D + Dense(1024 with Relu) + Sigmoid	Adam	10^{-3}	10

Οι υπερπαραμέτροι που επιλέχθηκαν σε όλα τα δίκτυα βασίστηκαν στα ευρήματα των W. Al-Dhabyani et al., οι οποίοι εκπαίδευσαν όλα τα προεκπαιδευμένα δίκτυα που χρησιμοποιήθηκαν, με την μόνη διάφορα ότι χρησιμοποίησαν το ίδιο σύνολο δεδομένων για να πραγματοποιήσουν κατηγοριοποίηση τριών κλάσεων (κακοήθεις, καλοήθεις και φυσιολογικές). Με τα αρχικά επίπεδα των μοντέλων και τις παραμέτρους αυτών να έχουν παγώσει, χρησιμοποιείται ο Adam optimizer με ρυθμό μάθησης 0.001 και ορίζονται 10 εποχές για την εκπαίδευση όλων των μοντέλων, επιτυγχάνοντας αξιοπρεπή ακρίβεια σε κάθε περίπτωση, είτε με επαύξηση δεδομένων είτε χωρίς [18]. Στον Πίνακα 3: Αποτελέσματα εκπαίδευσης των δικτύων από τους W. Al-Dhabyani et al. για την αναγνώριση τριών κατηγοριών σε εικόνες υπερήχων μαστού [18] διακρίνονται τα αποτελέσματα των W. Al-Dhabyani et al, με ή χωρίς επαύξηση δεδομένων.

Πίνακας 3: Αποτελέσματα εκπαίδευσης των δικτύων από τους W. Al-Dhabyani et al. για την αναγνώριση τριών κατηγοριών σε εικόνες υπερήχων μαστού [18].

Dataset	Method	Sub-Method	Without Augmentation	Traditional Augmentation
Dataset BUSI	CNN-AlexNet		58%	62%
	TL	VGG16	70%	74%
		Inception	68%	73%
		ResNet	79%	82%
		NASNet	83%	85%

Μέσα στον optimizer ορίζεται και η συνάρτηση κόστους, η οποία στην προκειμένη περίπτωση, που αφορά δυαδική κατηγοριοποίηση, είναι η `binary_crossentropy`. Η δομή του δικτύου, καθώς και οι παράμετροί του, μπορούν να προβληθούν κάθε στιγμή χρησιμοποιώντας την εντολή `model.summary()`. Για να προκύψουν χρήσιμα συμπεράσματα κατά τη διάρκεια της εκπαίδευσης και μετά το πέρας αυτής, είναι σημαντικό να προσθέσουμε πριν ξεκινήσουμε τη διαδικασία τόσο το `metrics=['accuracy']` όσο και το αντικείμενο `history`. Με αυτόν τον τρόπο, μπορούμε να παρακολουθούμε και να καταγράφουμε την ακρίβεια του μοντέλου μας, επιτρέποντας τη λεπτομερή ανάλυση της απόδοσής του τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα επαλήθευσης.

2.5 Παρουσίαση και σύγκριση αποτελεσμάτων

Κατά την διάρκεια των ετών, έχουν εξεταστεί πολλοί δείκτες για την αξιολόγηση της απόδοσης των αλγορίθμων βαθιάς μάθησης, ιδιαίτερα όταν αυτοί σχετίζονται με έναν τομέα όπως η βιοϊατρική τεχνολογία. Ο πιο γνωστός τρόπος αξιολόγησης και οπτικοποίησης της απόδοσης ενός αλγορίθμου που πραγματοποιεί κατηγοριοποίηση, είναι ο πίνακας σύγχυσης (confusion matrix). Αυτός ο πίνακας αποτελεί ένα εργαλείο που χρησιμοποιείται για την οπτικοποίηση της απόδοσης ενός αλγορίθμου και για τον υπολογισμό διαφόρων δεικτών αξιολόγησης. Είναι ένας τετράγωνος πίνακας όπου οι γραμμές αντιπροσωπεύουν τις πραγματικές τιμές και οι στήλες τις προβλεπόμενες τιμές του αλγορίθμου, παρέχοντας πληροφορίες για τους τύπους σφαλμάτων που πραγματοποιεί ένα ΝΔ [33]. Συγκεκριμένα, μέσα στον πίνακα σύγχυσης ορίζονται τα True Positives (TP) τα οποία είναι οι σωστές προβλέψεις της κλάσης 1, τα True Negatives (TN) που είναι οι σωστές προβλέψεις της κλάσης 0, τα False Positives (FP) τα οποία αντιπροσωπεύουν τις λανθασμένες προβλέψεις της κλάσης 1, γνωστές και ως ψευδώς θετικές, και τα False Negatives (FN) που είναι οι λανθασμένες προβλέψεις της κλάσης 0, γνωστές και ως ψευδώς αρνητικές.

Οι δείκτες αξιολόγησης που προκύπτουν από τους πίνακες σύγχυσης είναι:

1. Accuracy (Ακρίβεια): Ορίζεται ως το ποσοστό των σωστών προβλέψεων σε σχέση με τον συνολικό αριθμό των προβλέψεων.

$$Accuracy = \frac{TP + TN}{T}$$

2. Sensitivity (Ευαισθησία): Είναι ο αριθμός των πραγματικά θετικών (κλάση 1) που εντοπίστηκαν σωστά από τον αλγόριθμο ως προς το άθροισμα των πραγματικών θετικών και των ψευδώς αρνητικών.

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

3. Specificity (Ειδικότητα): Ο αριθμός των πραγματικά αρνητικών (κλάση 0) που εντοπίστηκαν σωστά από τον αλγόριθμο ως προς το άθροισμα των πραγματικών αρνητικών και των ψευδώς θετικών.

$$Specificity = \frac{TN}{TN + FP}$$

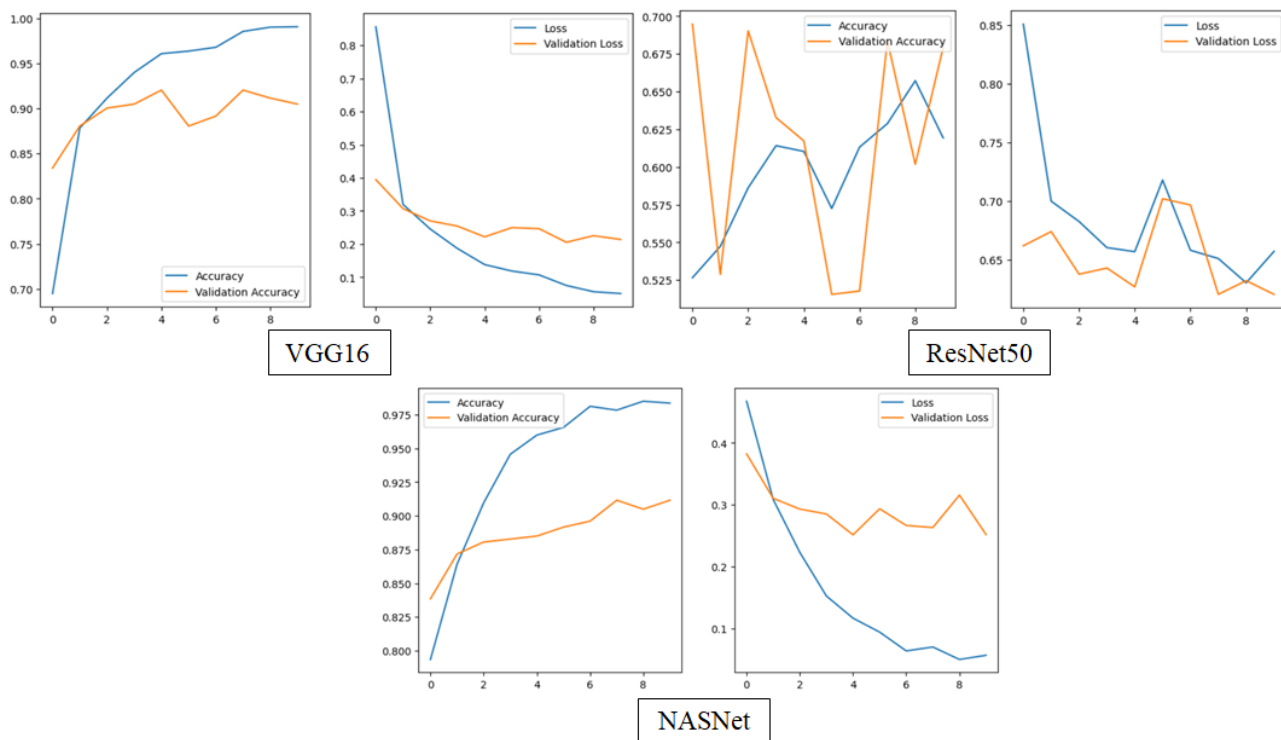
4. Precision (Ακρίβεια Προβλέψεων): Το ποσοστό των σωστών θετικών προβλέψεων ως προς το σύνολο των θετικών προβλέψεων.

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

Ένας ακόμη σημαντικός δείκτης που προκύπτει από τον πίνακα σύγκρισης, αφού υπολογιστούν οι παραπάνω δείκτες, είναι το F1-score. Η χρήση του F1-score προσφέρει μια πιο ολοκληρωμένη εικόνα για την απόδοση ενός αλγορίθμου, καθώς συνδυάζει την ακρίβεια προβλέψεων (precision) και την ευαισθησία (sensitivity) σε έναν ενιαίο δείκτη. Ο τύπος υπολογισμού του F1-score είναι ο εξής:

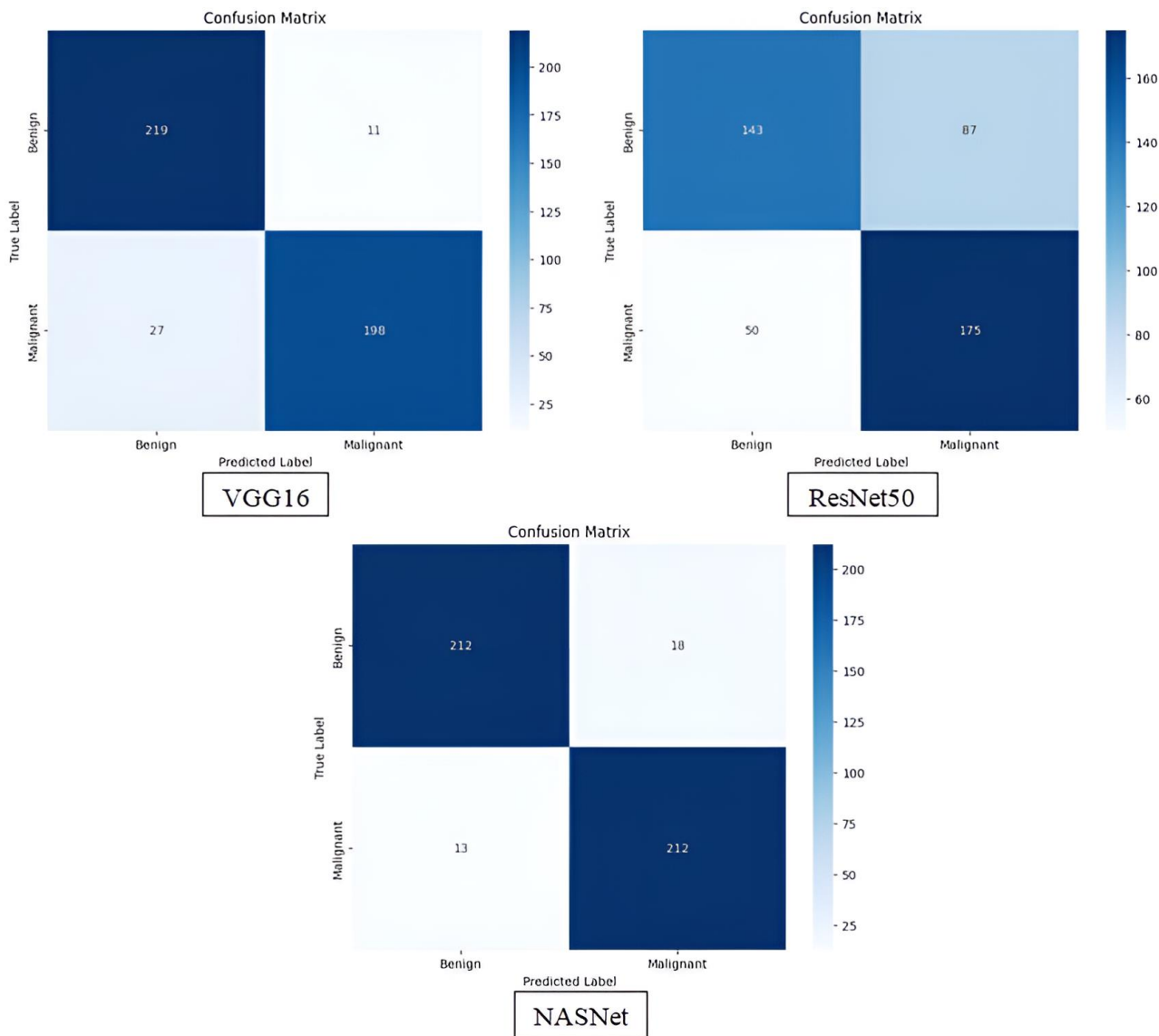
$$F1 - score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity}$$

Οι τρεις διαφορετικές αρχιτεκτονικές VGG16, ResNet και NASNet εκπαιδεύονται με τα δεδομένα εκπαίδευσης, και η απόδοσή τους κρίνεται με βάση τα δεδομένα επαλήθευσης που έχουν οριστεί. Με τη βοήθεια του αντικειμένου history και της βιβλιοθήκης matplotlib, είναι εξαιρετικά εύκολο να οπτικοποιηθεί η μεταβολή των τιμών της ακρίβειας (accuracy) καθώς και της απώλειας (loss) για τα δεδομένα εκπαίδευσης και τα δεδομένα επαλήθευσης σε κάθε εποχή του ΝΔ. Η οπτικοποίηση με αυτόν τον τρόπο είναι ιδιαίτερα χρήσιμη για την παρακολούθηση της διαδικασίας εκπαίδευσης και τη διάγνωση πιθανών προβλημάτων όπως η υπερπροσαρμογή (overfitting) ή η υποπροσαρμογή (underfitting) στα δεδομένα.



Εικόνα 27: Οπτικοποίηση της μεταβολής των τιμών accuracy και loss για τα δεδομένα εκπαίδευσης και τα δεδομένα επαλήθευσης για τα δίκτυα VGG16, ResNet και NASNet (10 epochs)

Η γενική ακρίβεια (accuracy) κάθε μοντέλου προκύπτει από την πρόβλεψη του κάθε μοντέλου πάνω στα δεδομένα δοκιμής. Από τα ίδια δεδομένα, σε συνδυασμό όμως αυτή τη φορά με τις πραγματικές ετικέτες των εικόνων, προκύπτει ο πίνακας σύγκρισης για κάθε αλγόριθμο. Στη συνέχεια, βάσει των τύπων που παρουσιάστηκαν παραπάνω, προκύπτει ο Πίνακας 4, στον οποίο καταγράφονται οι δείκτες αξιολόγησης για κάθε μοντέλο.



Εικόνα 28: Πίνακες Σύγκρισης των Δικτύων VGG16, ResNet και NASNet στα δεδομένα δοκιμής

Πίνακας 4: Δείκτες αξιολόγησης κάθε μοντέλου στα δεδομένα δοκιμής

Model	Accuracy	Sensitivity	Specificity	Precision	F1-score
VGG16	91,6%	88%	89%	94,7%	91,2%
ResNet50	70%	77,7%	74,1%	66,7%	71.8%
NASNet	93.2%	94,2%	92,2%	92,2%	93,1%

Εκτός από τους δείκτες που αξιολογούν την απόδοση ταξινόμησης ενός μοντέλου, υπάρχουν και άλλοι σημαντικοί παράγοντες που αντικατοπτρίζουν την πολυπλοκότητα ενός μοντέλου. Σε ένα μοντέλο βαθιάς μάθησης, εκτός από τους δείκτες απόδοσης, είναι σημαντικό να αναφέρεται και ο χρόνος που απαιτείται για την εκπαίδευσή του. Ιδιαίτερα για μοντέλα online εκπαίδευσης, είναι απαραίτητο αυτός ο χρόνος να είναι ο ελάχιστος δυνατός. Αντιθέτως, στην περίπτωση της offline εκπαίδευσης, μπορεί να είναι αποδεκτός και ένας μεγαλύτερος χρόνος εκπαίδευσης. Επιπλέον, σημαντικό ρόλο σε πολλές εφαρμογές παίζει και ο χρόνος που απαιτείται για να γίνει μια πρόβλεψη σε ένα σταθερό μέγεθος παρτίδας εισόδων, καθώς σε πολλές περιπτώσεις μπορεί να είναι κρίσιμο η πρόβλεψη και η εξαγωγή αποτελέσματος να πραγματοποιούνται με όσο το δυνατόν χαμηλότερες τιμές καθυστέρησης [34]. Στον Πίνακα 4 παρουσιάζονται ο χρόνος εκπαίδευσης (Training time) και η καθυστέρηση στην πρόβλεψη (Latency) κάθε δικτύου.

Πίνακας 5: Χρόνος εκπαίδευσης και χρόνος πρόβλεψης κάθε μοντέλου

Model	Training time	Latency
VGG16	8 mins	260ms
ResNet50	6 mins	125ms
NASNet	12 mins	535ms

Γίνεται ξεκάθαρο από τα γραφήματα μεταβολής των τιμών ακρίβειας και απώλειας, καθώς και από τους πίνακες σύγχυσης, ότι ένα από τα μοντέλα δεν τα καταφέρνει καθόλου καλά στην κατηγοριοποίηση των όγκων. Παρότι οι W. Al-Dhabyani et al., στις δοκιμές τους για την κατηγοριοποίηση τριών κατηγοριών, ανέδειξαν το ResNet για την υψηλή του ακρίβεια, με ή χωρίς τη χρήση παραδοσιακής επαύξησης δεδομένων, το συμπέρασμα από την εκπαίδευση που πραγματοποιήθηκε στη ΔΕ με τα διαθέσιμα δεδομένα είναι ότι επιτυγχάνει τη χαμηλότερη ακρίβεια από τα τρία μοντέλα (70%). Ο λόγος πίσω από αυτή την χαμηλή ακρίβεια, όπως φαίνεται από τα γραφήματα, πιθανότατα έγκειται στην υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης και στην αδυναμία του να γενικεύσει στα δεδομένα επαλήθευσης και δοκιμής.

Τα δίκτυα που ξεχώρισαν τόσο για τη συνολική ακρίβεια τους όσο και για τους υψηλούς δείκτες αξιολόγησης είναι το VGG16 και το NASNet. Το NASNet, μετά από 10 εποχές, επιτυγχάνει την καλύτερη συνολική ακρίβεια με 93.2%, έναντι του VGG16 που έχει λίγο χειρότερη συνολική ακρίβεια με 91.6%. Επιπλέον, το NASNet διακρίνεται για την απόδοσή του στο να εντοπίζει και να κατηγοριοποιεί σωστά σχεδόν όλους τους κακοήθεις όγκους, κάτι που φαίνεται από τον δείκτη Sensitivity. Επιτυγχάνει επίσης ένα πολύ κάλο Specificity και το καλύτερο F1-score. Παρόλα αυτά, το VGG16, αν και έχει χαμηλότερη απόδοση στους δείκτες Specificity, Sensitivity και F1-score, με συνολική ακρίβεια πολύ κοντά σε αυτή του NASNet, διακρίνεται για την εξαιρετική ικανότητά του να αναγνωρίζει καλοήθεις όγκους, όπως φαίνεται από τον δείκτη Precision που αγγίζει το 94,7%. Αξίζει επίσης να σημειωθεί ότι ο χρόνος εκπαίδευσης του VGG16 είναι μικρότερος από αυτόν του NASNet (8 λεπτά έναντι 12 λεπτών), καθώς και ότι ο χρόνος που χρειάζεται το μοντέλο για την πρόβλεψη είναι σημαντικά μικρότερος από αυτόν του NASNet (260ms έναντι 535ms). Σε κάθε περίπτωση, οι ακρίβειες και οι αποδόσεις που επιτεύχθηκαν τόσο στο VGG16 όσο και στο NASNet ξεπερνούν κατά πολύ τις ακρίβειες που παρουσιάστηκαν στο paper “Deep Learning Approaches for Data Augmentation and Classification of Breast Masses using Ultrasound Images” [18].

2.6 Συμπεράσματα Κεφαλαίου

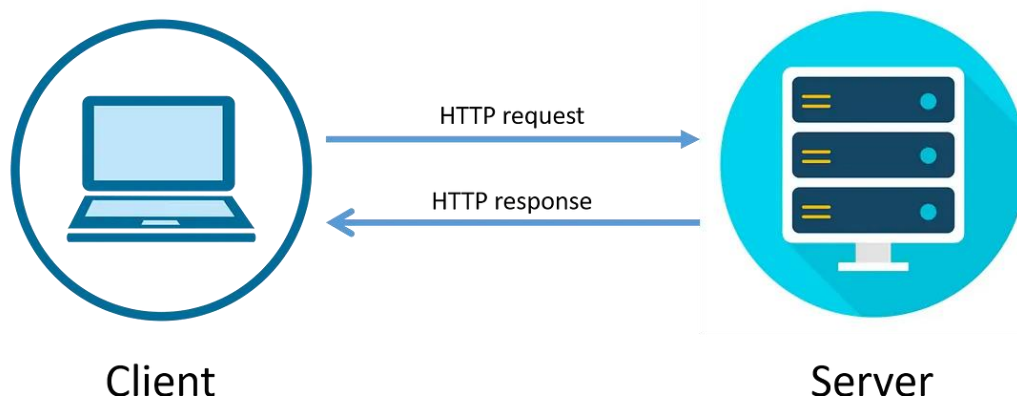
Σε αυτό το κεφάλαιο, αρχικά παρουσιάστηκαν τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη των μοντέλων βαθιάς μάθησης καθώς και το υπολογιστικό περιβάλλον στο οποίο εκπαιδεύτηκαν. Στη συνέχεια, έγινε μια αναλυτική επισκόπηση όλων των προηγούμενων ερευνών που αφορούν είτε την ανάπτυξη και εκπαίδευση διαφόρων μοντέλων για την εξαγωγή των πιο ακριβών προβλέψεων, είτε την ανάπτυξη εφαρμογών παρομοίων με αυτή της διπλωματικής. Το σύνολο δεδομένων καθώς και τα δεδομένα τα ίδια αν και όπως αναλύθηκε είναι περιορισμένα στον χώρο της βιοϊατρικής τεχνολογίας γενικότερα, επαυξήθηκε με ορισμένες μεθόδους ώστε να γίνει μεγαλύτερο. Αποτέλεσμα αυτής της επαύξησης ήταν και η εξισορρόπηση των δύο κατηγοριών, καθώς, όπως ήταν γνωστό, οι εικόνες καλοήθων όγκων πριν την προεπεξεργασία ξεπερνούσαν κατά πολύ αυτές των κακοήθων. Σε περίπτωση που αυτές οι εικόνες τροφοδοτούνταν σε ένα μοντέλο, υπήρχε ο κίνδυνος το μοντέλο να μην μπορεί να διακρίνει τις εικόνες κακοήθων όγκων με την ίδια ευκολία όπως αυτές των καλοήθων. Τα δεδομένα, μετά από κάποιους τελικούς μετασχηματισμούς, ήταν έτοιμα να τροφοδοτηθούν στους αλγόριθμους που επιλέχθηκαν για εκπαίδευση με τεχνικές μεταφοράς μάθησης. Οι αλγόριθμοι αυτοί ήταν οι VGG16, ResNet και NASNet, οι οποίοι αναλύθηκαν εκτενώς στο προηγούμενο κεφάλαιο και προσαρμόστηκαν με την προσθήκη τελικών επιπέδων για να πραγματοποιούν δυαδική κατηγοριοποίηση. Το μοντέλο NASNet ξεχώρισε για τις καλύτερες τιμές στους δείκτες Specificity και F1-score, ενώ επέδειξε συνολική ακρίβεια καλύτερη από αυτή του VGG16 που διακρίθηκε για το υψηλό του Precision. Επιπλέον το NASNet είχε την καλύτερη τιμή στον δείκτη Sensitivity και αυτό σημαίνει ότι ήταν σε θέση να διακρίνει καλύτερα το μεγαλύτερο ποσοστό των κακοήθων περιπτώσεων. Ο δείκτης Sensitivity είναι καθοριστικός, ειδικά όταν πρόκειται για μια ιατρική εφαρμογή όπως αυτή που αναπτύσσεται. Είναι πολύ πιο σημαντικό και κρίσιμο για την υγεία των ασθενών ένα ιατρικό σύστημα να αναγνωρίζει τους κακοήθεις όγκους και να προτεραιοποιεί αυτά τα περιστατικά, παρά να επιτυγχάνει καλύτερη ακρίβεια στην αναγνώριση των καλοήθων περιπτώσεων. Για αυτόν τον λόγο και δευτερευόντως για τις υψηλές τιμές σε συνολική ακρίβεια και F1-score, το μοντέλο που τελικά επιλέγεται για να πραγματοποιεί τη δυαδική κατηγοριοποίηση καλοήθων και κακοήθων όγκων από εικόνες υπερήχων μαστού για την τελική εφαρμογή, είναι το NASNet.

ΚΕΦΑΛΑΙΟ 3^ο : Ανάπτυξη διαδικτυακής εφαρμογής

Από τη στιγμή που η ανάπτυξη του μοντέλου που θα αποτελέσει τη βάση της εφαρμογής ολοκληρώθηκε, το επόμενο βήμα είναι η δημιουργία ενός περιβάλλοντος όπου οι χρήστες θα μπορούν εύκολα να ανεβάζουν εικόνες και να λαμβάνουν προβλέψεις από το μοντέλο. Αυτή η διαδικασία θα υλοποιηθεί μέσω ενός γραφικού περιβάλλοντος φιλικού προς το χρήστη, με ευκρινή και καλαίσθητα στοιχεία που καθοδηγούν τον χρήστη στην διαδικασία μεταφόρτωσης των εικόνων και λήψης των προβλέψεων. Για να είναι το τελικό αποτέλεσμα έτοιμο προς χρήση, απαιτείται η τήρηση μιας συγκεκριμένης διαδικασίας, από τη δημιουργία του διακομιστή μέχρι την ανάπτυξη των σελίδων της εφαρμογής και την φιλοξενία. Αυτή η διαδικασία διασφαλίζει ότι η εφαρμογή θα είναι φιλική προς τους προγραμματιστές και εύκολη στον εντοπισμό οποιουδήποτε σφάλματος στον κώδικα ή στην περαιτέρω ανάπτυξη. Η οπτική και λειτουργική αριστεία που θα διαθέτει επιπλέον θα συμβάλλει σημαντικά στη συνολική εμπειρία του χρήστη και στην προσφορά μιας ομαλής και αποδοτικής εμπειρίας.

3.1 Διακομιστής

Οι διακομιστές (servers) αδιαμφισβήτητα αποτελούν τον ακρογωνιαίο λίθο της σύγχρονης ψηφιακής εποχής. Είτε πρόκειται για φυσικούς υπολογιστές είτε για εικονικούς διακομιστές, λειτουργούν συνήθως σε περιβάλλοντα που χαρακτηρίζονται από υψηλή ασφάλεια και αποδοτικότητα. Οι σύγχρονοι διακομιστές είναι συνήθως εξοπλισμένοι με γρήγορους επεξεργαστές, μεγάλης χωρητικότητας μνήμη και αποθηκευτικά μέσα υψηλής ταχύτητας για να μπορούν να διαχειρίζονται τις αυξημένες απαιτήσεις των σύγχρονων εφαρμογών και υπηρεσιών. Σχεδιασμένοι να εξυπηρετούν πολλαπλά αιτήματα ταυτόχρονα, οι διακομιστές παρέχουν αξιόπιστες και γρήγορες απαντήσεις στα αιτήματα των πελατών (clients). Αυτή η αλληλεπίδραση μεταξύ διακομιστή και πελάτη μπορεί να παίρνει διάφορες μορφές, όπως η αποστολή δεδομένων, η εκτέλεση υπολογισμών, η αποθήκευση και ανάκτηση πληροφοριών, ή η επεξεργασία δεδομένων, όπως εικόνες και βίντεο. Οι διακομιστές είναι υπεύθυνοι για τη διασφάλιση ότι οι πελάτες σε κάθε περίπτωση θα μπορούν να λάβουν γρήγορες απαντήσεις σε όλα τους τα αιτήματά.



Εικόνα 29: Κύκλος αιτήματος και απάντησης μεταξύ μιας διαδικτυακής εφαρμογής και ενός διακομιστή⁷

⁷ <https://medium.com/@sujoy.swe/request-response-model-usages-anatomy-and-drawbacks-42464e475cf5>
[Accessed: 12 July 2024]

Ο διακομιστής της εφαρμογής αναπτύχθηκε σε Python χρησιμοποιώντας το framework Flask, ένα μικρό και ευέλικτο εργαλείο ιδανικό για τη δημιουργία διαδικτυακών εφαρμογών και API. Η επιλογή της Python έγινε για να διευκολυνθεί η διαδικασία προεπεξεργασίας εικόνων από πλευράς διακομιστή και για την επαναχρησιμοποίηση συναρτήσεων που χρησιμοποιήθηκαν κατά τη δημιουργία του μοντέλου. Αρχικά, εισάγονται οι απαραίτητες βιβλιοθήκες, είτε για τη διαχείριση του συστήματος είτε για την προεπεξεργασία εικόνας. Η πρώτη ενέργεια είναι η δημιουργία ενός instance της εφαρμογής Flask και η ενεργοποίηση του CORS, το οποίο επιτρέπει την επικοινωνία με πολλούς πελάτες από διάφορα domains. Στη συνέχεια, κατά την εκκίνηση του διακομιστή, φορτώνεται μόνο μία φορά το μοντέλο μηχανικής μάθησης που επιλέχθηκε (NASNet), εξασφαλίζοντας ότι θα είναι διαθέσιμο για όλα τα αιτήματα των χρηστών.

Ένα πολύ σημαντικό κομμάτι του διακομιστή είναι η κατασκευή ενός συγκεκριμένου route το οποίο δέχεται αιτήματα POST. Όταν ο πελάτης στέλνει ένα αίτημα με μορφή εικόνας που έχει κωδικοποιηθεί σε Base64, ο διακομιστής πρώτα αποκωδικοποιεί την εικόνα και στη συνέχεια την προετοιμάζει για πρόβλεψη. Η εικόνα μετατρέπεται μέσω μιας συνάρτησης σε RGB, αλλάζει το μέγεθος της στις συγκεκριμένες διαστάσεις που απαιτεί το μοντέλο (331x331) και κανονικοποιείται όπως κατά τη διάρκεια της εκπαίδευσης και των δοκιμών στο προηγούμενο κεφάλαιο. Το μοντέλο επιστρέφει ένα αποτέλεσμα το οποίο μετατρέπεται σε μορφή JSON μετά από κάποια δευτερόλεπτα. Το αποτέλεσμα αυτό είναι ένας αριθμός από το 0 έως το 1, όπου το 0 υποδηλώνει εικόνα καλοήθους όγκου ενώ το 1 υποδηλώνει εικόνα κακοήθους όγκου.

Επιπλέον, ο διακομιστής είναι σχεδιασμένος για να διαχειρίζεται σφάλματα με αποτελεσματικό τρόπο. Αν προκύψει οποιοδήποτε πρόβλημα κατά την επεξεργασία του αιτήματος, ο διακομιστής καταγράφει το σφάλμα και επιστρέφει ένα γενικό μήνυμα σφάλματος στην κονσόλα της διαδικτυακής εφαρμογής (κυρίως για σκοπούς αποσφαλμάτωσης), ενώ σε ορισμένες περιπτώσεις επιστρέφει ένα μήνυμα σφάλματος και στον χρήστη, διασφαλίζοντας ότι ενημερώνεται για το πρόβλημα χωρίς να εκθέτονται ευαίσθητες πληροφορίες του συστήματος.

3.2 Γραφικό περιβάλλον χρήστη

3.2.1 Σελίδα παρουσίασης εφαρμογής

Ένα από τα σημαντικότερα θέματα κατά τη ανάπτυξη μιας ιστοσελίδας, ή στη συγκεκριμένη περίπτωση μιας διαδικτυακής εφαρμογής, είναι η δημιουργία ενός εντυπωσιακού Home Page που θα κεντρίσει το ενδιαφέρον του χρήστη και θα τον ενθαρρύνει να εξερευνήσει περισσότερο την εφαρμογή. Σε πολλές περιπτώσεις διπλωματικών εργασιών και άλλων διαδικτυακών εφαρμογών, παρατηρείται έλλειψη προσοχής στον σχεδιασμό και την παρουσίαση, με τους δημιουργούς να περιορίζονται σε ένα απλό σκελετό ιστοσελίδας χωρίς έμφαση στην εμπειρία του χρήστη. Το TumorNet έρχεται να ανατρέψει αυτά τα στερεότυπα, συνδυάζοντας ένα όμορφο γραφικό περιβάλλον με την παροχή ενός μοντέλου μεγάλης ακρίβειας που μπορεί να κατηγοριοποιήσει όγκους ως καλοήθεις ή κακοήθεις.

Η επιλογή της React ως το κύριο εργαλείο για τη δημιουργία της διαδικτυακής εφαρμογής δεν είναι τυχαία. Ένας λόγος που επιλέχτηκε έναντι άλλων frameworks είναι η αποτελεσματικότητα και η απόδοση του Virtual DOM, το οποίο επιτρέπει γρήγορη ανανέωση του γραφικού περιβάλλοντος. Η React διαθέτει επίσης μία από τις μεγαλύτερες και πιο ενεργές κοινότητες προγραμματιστών, προσφέροντας συνεχή υποστήριξη και συντήρηση, νέων ή μη, εργαλείων και βιβλιοθηκών [35]. Η ευελιξία που έχει επιτρέποντας την επαναχρησιμοποίηση components σε μια εφαρμογή και η σύνταξη JSX που παρέχει έναν συνδυασμό HTML και JavaScript, καθιστά τον κώδικα πιο ευανάγνωστο και εύκολο στη συντήρηση σε σχέση με άλλα Web frameworks. Με αυτά τα χαρακτηριστικά, η React αναδεικνύεται ως η ιδανική επιλογή για τη δημιουργία σύγχρονων διαδικτυακών εφαρμογών και ιστοσελίδων⁸.

Στη σελίδα παρουσίασης της εφαρμογής, δίνεται έμφαση στην ανάδειξη των σημαντικών λεπτομερειών του συστήματος και της εφαρμογής, ενώ ταυτόχρονα παρέχονται οι απαραίτητες πληροφορίες που πρέπει να κατανοήσει ο χρήστης πριν κάνει χρήση του εργαλείου, χωρίς να υπερφορτώνεται με περιττές λεπτομέρειες. Περιγράφονται συνοπτικά τα κύρια χαρακτηριστικά και τα οφέλη της εφαρμογής και αναφέρεται η προσφορά της προηγμένης TN για τη διάγνωση όγκων, η ακρίβεια του μοντέλου και κάποια άλλα στατιστικά στοιχεία που αναδεικνύουν την απόδοση της εφαρμογής. Τέλος, παρουσιάζεται μια σύντομη ανασκόπηση της αξιολόγησης της εφαρμογής και παρέχονται σύνδεσμοι για περισσότερες πληροφορίες σχετικά με το αποθετήριο του κώδικα αλλά και της διπλωματικής της ίδιας. Με αυτόν τον τρόπο, δημιουργείται ένα ενδιαφέρον σημείο εκκίνησης για την καθοδήγηση των χρηστών στην περαιτέρω εξερεύνηση της εφαρμογής και της βιβλιογραφίας.

⁸ <https://react.dev/learn> [Accessed: 12 July 2024]

3.2.2 Σελίδα εφαρμογής

Το TumorNet, εκτός από τη σελίδα παρουσίασης που εμφανίζεται σε κάθε χρήστη κατά την είσοδό του στην ιστοσελίδα, περιλαμβάνει και την ίδια την εφαρμογή. Αυτή η εφαρμογή έχει σχεδιαστεί με γνώμονα την ευκολία χρήσης, προσφέροντας μια ομαλή και φιλική προς τον χρήστη εμπειρία. ενσωματώνοντας εργαλεία που καθοδηγούν τον χρήστη βήμα-βήμα, εξασφαλίζοντας την επιτυχή μεταφόρτωση εικόνων και τη λήψη ακριβών προβλέψεων.

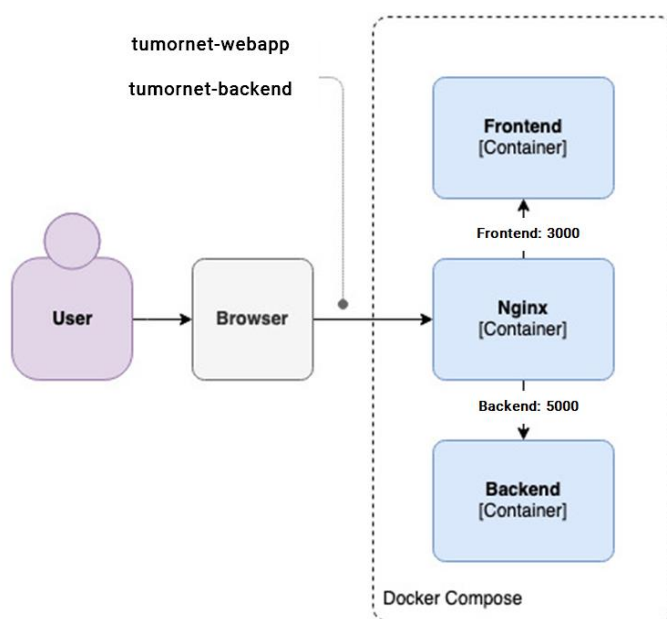
Στην αρχική σελίδα της εφαρμογής υπάρχει ένα μοναδικό στοιχείο input που επιτρέπει μόνο αρχεία τύπου PNG και μέσω του οποίου οι χρήστες μπορούν να ανεβάσουν μια εικόνα στην ιστοσελίδα. Μόλις ο χρήστης επιλέξει μια εικόνα, η εφαρμογή διαβάζει το αρχείο και το κωδικοποιεί σε μορφή base64 χρησιμοποιώντας το API FileReader της React. Κατά τη διάρκεια φόρτωσης της εικόνας, εμφανίζεται ένα στοιχείο loader για να ενημερώσει τον χρήστη ότι η διαδικασία είναι σε εξέλιξη. Η επιλεγμένη εικόνα θα εμφανιστεί στην οθόνη μόλις φορτωθεί πλήρως. Στη συνέχεια, γίνεται διαθέσιμο το κουμπί "Predict", το οποίο όταν ο χρήστης πατήσει, η εφαρμογή στέλνει ένα αίτημα POST στον διακομιστή, περιέχοντας στο σώμα την εικόνα κωδικοποιημένη σε base64. Κατά την εκτέλεση αυτής της διαδικασίας, εμφανίζεται ένας δεύτερος loader για να ενημερώσει τον χρήστη ότι η διαδικασία της πρόβλεψης είναι σε εξέλιξη. Μόλις ληφθεί η απάντηση από τον διακομιστή, η εφαρμογή εμφανίζει το αποτέλεσμα της πρόβλεψης με δύο τρόπους. Ο πρώτος τρόπος είναι η συνοπτική αναφορά, στην οποία εμφανίζεται στην οθόνη εάν η εικόνα ανήκει σε έναν καλοήγητο όγκο ή σε έναν κακοήγητο όγκο, με βάση εάν η πρόβλεψη είναι κάτω ή πάνω από το 0.5, το οποίο αποτελεί και το όριο απόφασης. Ο δεύτερος τρόπος παρουσίασης του αποτελέσματος είναι μέσω ενός γραφήματος που παρουσιάζει την πρόβλεψη του μοντέλου για κάθε κλάση, δείχνοντας με αυτόν τον τρόπο πιο αναλυτικά την κατηγοριοποίηση του συστήματος για την εικόνα.

Η σελίδα παρουσίασης της εφαρμογής, καθώς και η ίδια η εφαρμογή TumorNet, έχουν σχεδιαστεί με γνώμονα τη λειτουργία όχι μόνο σε υπολογιστές, αλλά και σε κινητά τηλέφωνα και tablets. Με τη χρήση των κατάλληλων styles μέσω της CSS και της ισχυρής πλατφόρμας της React, η εφαρμογή προσαρμόζεται εύκολα τόσο σε μεγάλες όσο και σε μικρές οθόνες. Αν και η χρήση ιατρικών εφαρμογών σε κινητά τηλέφωνα δεν είναι ακόμη ιδιαίτερα διαδεδομένη μεταξύ του ιατρικού προσωπικού, αυτή η προσαρμοστικότητα ενισχύει τη χρηστικότητα και την προσβασιμότητα της εφαρμογής, επιτρέποντας στους επαγγελματίες υγείας να έχουν πρόσβαση στην εφαρμογή οπουδήποτε και οποτεδήποτε.

3.3 Φιλοξενία εφαρμογής

Για την εφαρμογή που αναπτύχθηκε στην διπλωματική, πέρα από τις σελίδες που εμφανίζονται στον χρήστη καθώς και του διακομιστή που παραμένει αθέατος για τους χρήστες, απαιτείται ένα επιπλέον βήμα για την ολοκλήρωσή της. Η φιλοξενία της εφαρμογής είναι καθοριστικής σημασίας, καθώς μετατρέπει την τοπική εφαρμογή σε μια δημόσια προσβάσιμη υπηρεσία για το ευρύ κοινό. Στη σύγχρονη εποχή, υπάρχει μια συνεχής τάση προς την αποκέντρωση πολλών εφαρμογών και διαδικτυακών ιστοσελίδων, αποφεύγοντας την εξάρτηση από εφαρμογές τρίτων. Μία από τις τεχνολογίες που επιτρέπει αυτό το επίπεδο αυτονομίας και ευελιξίας είναι η τεχνολογία των *microservices* όπως είναι το Docker, το Podman και άλλα.

Στην συγκεκριμένη ΔΕ επιλέχθηκε η τεχνολογία των *microservices*, καθώς μέσω αυτών επιτυγχάνεται μια αποδοτική, αξιόπιστη και ασφαλής λύση για τη φιλοξενία εφαρμογών ανεξαρτήτως μεγέθους. Η εφαρμογή φιλοξενείται σε δύο ξεχωριστά *containers*: το πρώτο φιλοξενεί το Backend και πιο συγκεκριμένα τον διακομιστή, ο οποίος διαχειρίζεται τα εισερχόμενα αιτήματα POST και εκτελεί προεπεξεργασία στις εικόνες, ενώ το δεύτερο φιλοξενεί το Frontend, το οποίο παρέχει το γραφικό περιβάλλον για την αλληλεπίδραση του χρήστη με την εφαρμογή. Η επικοινωνία μεταξύ αυτών των δύο *containers* διασφαλίζεται μέσω ενός proxy, που δρομολογεί τα αιτήματα και εγγυάται την αποδοτική συνεργασία των δυο *containers*. Για εφαρμογές που διαχειρίζονται ευαίσθητες πληροφορίες, και ειδικότερα στην περίπτωσή μας όπου έχουμε αναπτύξει μια εφαρμογή που επεξεργάζεται ιατρικά δεδομένα, η ασφάλεια αποτελεί ύψιστη προτεραιότητα. Η προστασία των δεδομένων της εφαρμογής καθώς και η ασφαλής επικοινωνία χωρίς παρεμβολές τρίτων, επιτυγχάνεται με την χρήση HTTPS μέσω SSL certificates. Αυτή η μέθοδος κρυπτογραφεί όλες τις ανταλλαγές δεδομένων, διασφαλίζοντας την ιδιωτικότητα τόσο των πληροφοριών των χρηστών όσο και της ιστοσελίδας. Επιπλέον, η χρήση του Nginx ως reverse proxy ενισχύει περαιτέρω την ασφάλεια και την απόδοση της εφαρμογής, εξασφαλίζοντας την σταθερότητα αυτής. Στο παρακάτω διάγραμμα παρουσιάζεται η δομή της εφαρμογής και η επικοινωνία μεταξύ των *containers* με τον Nginx ως reverse proxy, όπως περιγράφηκε προηγουμένως.

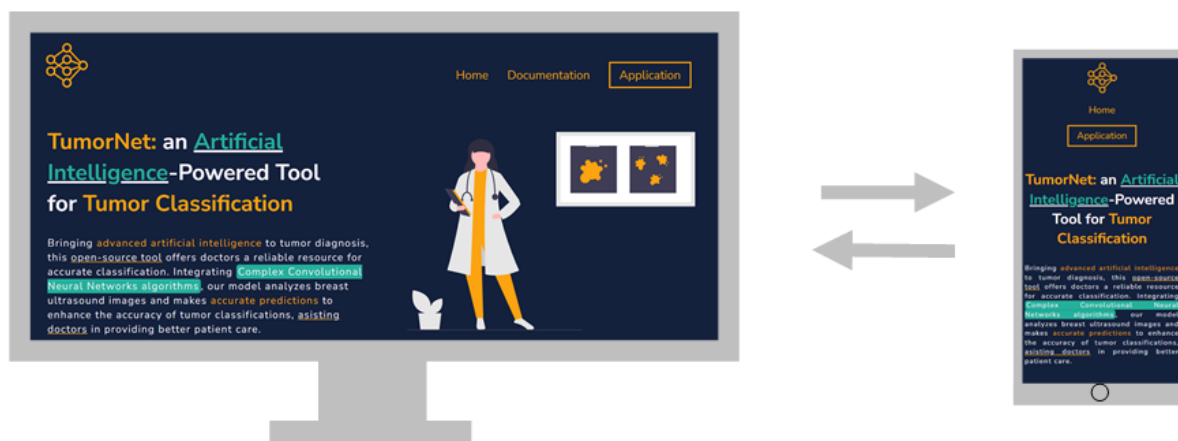


Εικόνα 30: Μπλοκ διάγραμμα των *containers* της εφαρμογής

3.4 Τελικό αποτέλεσμα

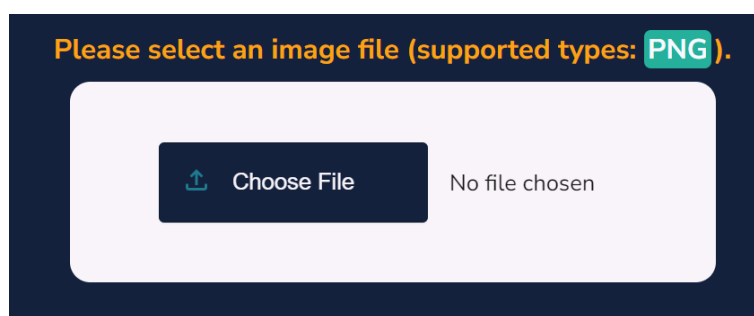
Για να παρέχουμε μια ολοκληρωμένη κατανόηση του τρόπου λειτουργίας της διαδικτυακής εφαρμογής που αναπτύχθηκε, παραθέτουμε μια αναλυτική επίδειξη των δυνατοτήτων και των λειτουργιών της. Η παρουσίαση αυτή περιλαμβάνει μια σειρά από στιγμιότυπα (screenshots) της εφαρμογής, τα οποία απεικονίζουν βήμα προς βήμα τη διαδικασία που ακολουθεί ένας χρήστης, από την είσοδό του στην αρχική σελίδα της εφαρμογής έως τη λήψη απαντήσεων και προβλέψεων από τον διακομιστή.

1. Παρουσίαση της Κεντρικής Σελίδας της Εφαρμογής: Η κεντρική σελίδα της εφαρμογής εισάγει τον χρήστη σε ένα όμορφο γραφικό περιβάλλον, στο οποίο παρουσιάζονται οι πληροφορίες, οι δυνατότητες και ο σκοπός της εφαρμογής. Υπενθυμίζεται ότι όλες οι σελίδες της εφαρμογής, συμπεριλαμβανομένης και της αρχικής σελίδας, είναι σχεδιασμένες με γνώμονα τη λειτουργία σε υπολογιστές, tablet και κινητά τηλέφωνα.



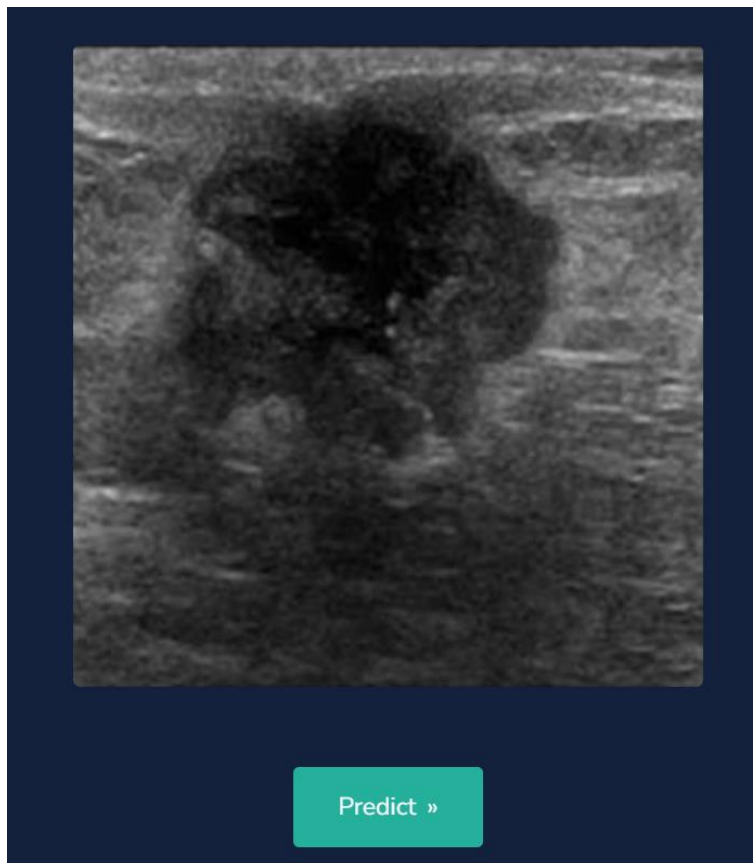
Εικόνα 31: Σελίδα παρουσίασης της εφαρμογής σε υπολογιστή και κινητό

2. Κύρια Διεπαφή Χρήστη: Σελίδα Εφαρμογής: Ο χρήστης, μόλις πατήσει το κουμπί "Application" που εμφανίζεται στην αρχική σελίδα καθώς και στη μπάρα πλοήγησης, μεταφέρεται στη σελίδα της εφαρμογής. Εκεί, εμφανίζεται ένα input box για την εισαγωγή της εικόνας, που θα χρησιμοποιηθεί για την παραγωγή της πρόβλεψης. Από την εφαρμογή γίνονται αποδεκτά μόνο αρχεία σε μορφή PNG. Κατά τη διάρκεια φόρτωσης της εικόνας, εμφανίζεται ένα στοιχείο φόρτωσης (loader) για να ενημερώσει τον χρήστη ότι η διαδικασία είναι σε εξέλιξη.



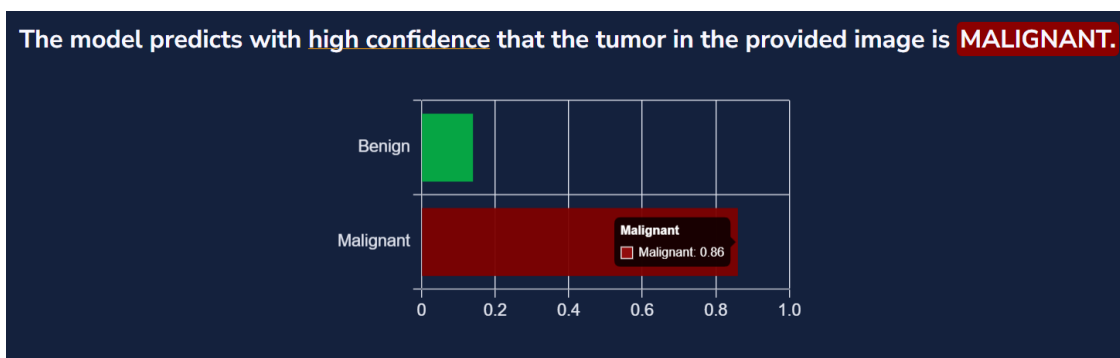
Εικόνα 32: Input Box με το οποίο ο χρήστης επιλέγει μια εικόνα στην σελίδα εφαρμογής

3. Αποστολή Δεδομένων στον Διακομιστή: Μόλις η διαδικασία φόρτωσης της εικόνας ολοκληρωθεί, η επιλεγμένη εικόνα εμφανίζεται στον χρήστη και γίνεται διαθέσιμο το κουμπί "Predict". Με την επιλογή του κουμπιού αυτού, αποστέλλεται αίτημα στον server και εμφανίζεται άλλο ένα στοιχείο φόρτωσης έως ότου υπάρξει απάντηση από τον διακομιστή.



Εικόνα 33: Εμφάνιση εικόνας στον χρήστη και ενεργοποίηση κουμπιού "Predict"

4. Απάντηση Διακομιστή και Παραλαβή Αποτελεσμάτων: Ο διακομιστής αποστέλλει την απάντηση στον χρήστη, κατηγοριοποιώντας την εικόνα ως καλοήγη ή κακοήγη όγκο. Εκτός από τη συνοπτική αναφορά, εμφανίζεται και ένα γράφημα (bar chart) το οποίο παρουσιάζει την πρόβλεψη του μοντέλου για κάθε κατηγορία αριθμητικά.



Εικόνα 34: Εμφάνιση πρόβλεψης μοντέλου με συνοπτική αναφορά και αναλυτικό γράφημα

ΚΕΦΑΛΑΙΟ 4^ο : Συμπεράσματα - Βελτιώσεις

Στην παρούσα ΔΕ, αναπτύχθηκε μια διαδικτυακή εφαρμογή η οποία, μέσω ενός μοντέλου βαθιάς μάθησης, πραγματοποιεί δυαδική κατηγοριοποίηση σε εικόνες υπερήχων καλοήθων και κακοήθων όγκων του μαστού. Ο κύριος στόχος ήταν να δημιουργηθεί ένα αξιόπιστο εργαλείο που θα βοηθήσει τους ιατρούς στη διάγνωση και την προτεραιοποίηση περιστατικών καρκίνου του μαστού στις γυναίκες, αλλά και να ενσωματώσει περαιτέρω την TN στον τομέα της ιατρικής περίθαλψης, βελτιώνοντας την ακρίβεια και την ταχύτητα διάγνωσης. Μετά από εκτενή ανάλυση και προεπεξεργασία των διαθέσιμων δεδομένων και συγκρίσεις διαφόρων μοντέλων, προέκυψε ότι το NASNet επιτυγχάνει την καλύτερη ακρίβεια, της τάξεως του 93,2%, στην κατηγοριοποίηση καλοήθων και κακοήθων όγκων.

Οι πρακτικές εφαρμογές των ευρημάτων της παρούσας διπλωματικής περιλαμβάνουν τη χρήση της διαδικτυακής εφαρμογής σε κλινικές και νοσοκομεία, όπου μπορεί να χρησιμοποιηθεί ως υποστηρικτικό εργαλείο από τους ιατρούς για την ταχύτερη και ακριβέστερη διάγνωση των όγκων του μαστού, συμβάλλοντας στην αποσυμπύεση του φόρτου εργασίας του ήδη επιβαρυσμένου ιατρικού προσωπικού. Ήδη η χρήση εργαλείων TN στην ιατρική έχει δείξει ότι πολλοί ακτινολόγοι έχουν βελτιώσει την απόδοσή τους στη διάγνωση καρκίνου του μαστού χωρίς την ανάγκη επιπρόσθετου χρόνου μελέτης [36]. Η εφαρμογή, όμως, εκτός από τους βασικούς λόγους για τους οποίους δημιουργήθηκε, όπως αναφέρθηκαν παραπάνω, εξυπηρετεί και έναν ακόμη σκοπό. Μέσω της ελεύθερης πρόσβασης στον κώδικα, είτε του μοντέλου είτε της εφαρμογής, είναι πολύ εύκολο κάποιος να κάνει αλλαγές στην εφαρμογή τοπικά στον υπολογιστή του και να οπτικοποιήσει τα εκάστοτε αποτελέσματα της έρευνάς του ή του μοντέλου του, καθώς και να κατασκευάσει μια εντελώς διαφορετική εφαρμογή κατηγοριοποίησης εικόνων.

Όσον αφορά τις περαιτέρω βελτιώσεις που μπορούν να γίνουν στην εφαρμογή συνολικά, υπάρχουν αρκετές σημαντικές προτάσεις. Πρωτίστως, η ενσωμάτωση ενός αλγορίθμου βαθιάς μάθησης υψηλής ακρίβειας, ο οποίος θα μπορεί να αξιολογεί αν μια εικόνα είναι εικόνα υπερήχου μαστού πριν αποστείλει αίτημα στο μοντέλο που πραγματοποιεί την κατηγοριοποίηση, κρίνεται απαραίτητη σε μια τέτοια εφαρμογή, παρόλο που προορίζεται για χρήση αποκλειστικά από ιατρικό προσωπικό. Επιπλέον, από την πλευρά του γραφικού περιβάλλοντος, θα ήταν χρήσιμο να προστεθεί η δυνατότητα περικοπής της εικόνας στο σημείο όπου υπάρχει ένας όγκος, καθώς η ελαχιστοποίηση της περιττής πληροφορίας σε μια εικόνα μπορεί να βελτιώσει σημαντικά την ακρίβεια των προβλέψεων. Επίσης, η προσθήκη υποστήριξης περισσότερων τύπων εικόνας, τα οποία χρησιμοποιούνται συχνότερα από το ιατρικό προσωπικό, όπως είναι τα αρχεία με κατάληξη .dcm , .jif ή ακόμα και εικόνες τύπου .jpeg, αποτελεί μια βελτίωση που θα ενισχύσει την εμπειρία χρήσης της εφαρμογής.

Βελτιώσεις μπορούν να υπάρξουν επίσης και από πλευράς αλγορίθμων και μοντέλου. Η προσθήκη περισσότερων δεδομένων για την εκπαίδευση του ΣΝΔ αναμένεται να βελτιώσει σημαντικά την ικανότητα του μοντέλου να αναγνωρίζει μεγαλύτερο πλήθος εικόνων και όγκων. Επιπλέον, υπάρχει πρόταση για την αναθεώρηση του αλγορίθμου με τον οποίο εξάγεται η τελική πρόβλεψη. Αντί για πρόβλεψη μόνο για την εικόνα που εισάγεται από τον χρήστη, με τη χρήση κατάλληλων μεθόδων προεπεξεργασίας, όπως αυτές που αναφέρθηκαν στα προηγούμενα κεφάλαια, θα μπορούσε να πραγματοποιηθεί πρόβλεψη και στην καθρεπτισμένη εικόνα καθώς και στην εικόνα με προσθήκη θορύβου. Με αυτόν τον τρόπο, η τελική πρόβλεψη θα μπορούσε να εξάγεται με βάση την κλάση

που συγκεντρώνει την πλειοψηφία των προβλέψεων και να παρουσιάζεται στον χρήστη. Ασφαλώς, αυτό συνεπάγεται με μεγαλύτερο χρόνο απόκρισης του διακομιστή στο αίτημα του χρήστη, ωστόσο μπορεί να οδηγήσει σε βελτιωμένη ακρίβεια και αξιοπιστία των αποτελεσμάτων.

Συνοψίζοντας, είναι σημαντικό να τονιστεί η ανάγκη για ενθάρρυνση περισσότερων μελλοντικών μελετών στον τομέα της TN και της YO, καθώς και της συσχέτισής αυτών με τη βιοϊατρική τεχνολογία. Η παρούσα ΔΕ μπορεί να θεωρηθεί ως ένα σημείο εκκίνησης για την περαιτέρω εξερεύνηση σε αυτούς τους τομείς. Οι προτεινόμενες βελτιώσεις στο σύστημα, τόσο από πλευράς αλγορίθμων όσο και από πλευράς γραφικού περιβάλλοντος, στοχεύουν στην ενίσχυση της ακρίβειας, της αποδοτικότητας και της χρηστικότητας της εφαρμογής. Με τη συνέχιση της έρευνας και την ανάπτυξη παρόμοιων εφαρμογών, αναμένεται να επιτευχθεί σημαντική πρόοδος σε όλους τους τομείς που προαναφέρθηκαν και ιδιαίτερα στον τομέα της αποτελεσματικότερης και ταχύτερης παροχής υγειονομικής περίθαλψης.

Βιβλιογραφία – Αναφορές

- [1] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.
- [2] A. Barr, E. A. Feigenbaum, and P. R. Cohen, “The handbook of artificial intelligence.” HeurisTech Press, 1981.
- [3] Ρ.-Ε. Κινγκ, “Υπολογιστική Νοημοσύνη στον Έλεγχο Συστημάτων” *Τραυλό ΙΙ*, 1998.
- [4] S. Haykin, “Neural networks and learning machines, 3/E.” Pearson Education India, 2009.
- [5] K. Suzuki, *Artificial neural networks: methodological advances and biomedical applications*. BoD–Books on Demand, 2011.
- [6] E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, “Hyperparameter tuning for machine and deep learning with R: A practical guide.” Springer Nature, 2023.
- [7] K. O’shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.
- [8] H. Yingge, I. Ali, and K.-Y. Lee, “Deep neural networks on chip—a survey,” in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020, pp. 589–592.
- [9] A. A. Mukhlif, B. Al-Khateeb, and M. A. Mohammed, “Incorporating a novel dual transfer learning approach for medical images,” *Sensors*, vol. 23, no. 2, p. 570, 2023.
- [10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [13] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*, 2013, pp. 1310–1318.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [15] P. Mhasakar, P. Trivedi, S. Mandal, and S. K. Mitra, “Handwritten digit recognition using Bayesian ResNet,” *SN Computer Science*, vol. 2, no. 5, p. 399, 2021.
- [16] B. Zoph, V. Vasudevan, J. Shlens, and Q. v Le, “Learning transferable architectures for scalable image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [17] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, “Dataset of breast ultrasound images,” *Data in brief*, vol. 28, p. 104863, 2020.
- [18] W. Al-Dhabyani, M. Gomaa, H. Khaled, and F. Aly, “Deep learning approaches for data augmentation and classification of breast masses using ultrasound images,” *Int. J. Adv. Comput. Sci. Appl*, vol. 10, no. 5, pp. 1–11, 2019.
- [19] [19] E. P. v Le, Y. Wang, Y. Huang, S. Hickman, and F. J. Gilbert, “Artificial intelligence in breast imaging,” *Clinical radiology*, vol. 74, no. 5, pp. 357–366, 2019.
- [20] A. T. Stavros, D. Thickman, C. L. Rapp, M. A. Dennis, S. H. Parker, and G. A. Sisney, “Solid breast nodules: use of sonography to distinguish between benign and malignant lesions.” *Radiology*, vol. 196, no. 1, pp. 123–134, 1995.
- [21] M. H. Yap *et al.*, “Automated breast ultrasound lesions detection using convolutional neural networks,” *IEEE journal of biomedical and health informatics*, vol. 22, no. 4, pp. 1218–1226, 2017.
- [22] Z. Fang, W. Zhang, and H. Ma, “Breast cancer classification with ultrasound images based on SLIC,” in *Frontier Computing: Theory, Technologies and Applications (FC 2019) 8*, 2020, pp. 235–248.

- [23] L. Alzubaidi *et al.*, “Towards a better understanding of transfer learning for medical imaging: a case study,” *Applied Sciences*, vol. 10, no. 13, p. 4523, 2020.
- [24] R. C. Joshi, D. Singh, V. Tiwari, and M. K. Dutta, “An efficient deep neural network based abnormality detection and multi-class breast tumor classification,” *Multimedia Tools and Applications*, vol. 81, no. 10, pp. 13691–13711, 2022.
- [25] S. Afrifa, V. Varadarajan, P. Appiahene, and T. Zhang, “A Novel Artificial Intelligence Techniques for Women Breast Cancer Classification Using Ultrasound Images,” *Clinical and Experimental Obstetrics & Gynecology*, vol. 50, no. 12, p. 271, 2023.
- [26] M. Bouhadda, “Breast Cancer Image Classification”.
- [27] D. Chowdhury *et al.*, “ABCandroid: a cloud integrated android app for noninvasive early breast cancer detection using transfer learning,” *Sensors*, vol. 22, no. 3, p. 832, 2022.
- [28] S. Pawar, P. Shukla, P. Bagal, A. Dawkhar, and R. Jadhav, “Development of Web Application for Breast Cancer Detection using Machine Learning Classifier,” in *13th International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT*, 2022, pp. 552–556.
- [29] Y. Sun and G. Mogos, “Impact of visual distortion on medical images,” *IAENG International Journal of Computer Science*, vol. 49, no. 1, pp. 36–45, 2022.
- [30] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural networks*, vol. 106, pp. 249–259, 2018.
- [31] [31] P. Domingos, “A few useful things to know about machine learning” *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.
- [32] A. Hijab, M. A. Rushdi, M. M. Gomaa, and A. Eldeib, “Breast cancer classification in ultrasound images using transfer learning” in *2019 Fifth international conference on advances in biomedical engineering (ICABME)*, 2019, pp. 1–4.
- [33] D. R. Sarvamangala and R. v Kulkarni, “Convolutional neural networks in medical image understanding: a survey,” *Evolutionary intelligence*, vol. 15, no. 1, pp. 1–22, 2022.
- [34] S. Gamage and J. Samarabandu, “Deep learning methods in network intrusion detection: A survey and an objective comparison,” *Journal of Network and Computer Applications*, vol. 169, p. 102767, 2020.
- [35] E. Saks, “JavaScript Frameworks: Angular vs React vs Vue.” 2019.
- [36] A. Rodriguez-Ruiz *et al.*, “Detection of breast cancer with mammography: effect of an artificial intelligence support system,” *Radiology*, vol. 290, no. 2, pp. 305–314, 2019.

Παράρτημα Α – Κώδικας για την υλοποίηση μοντέλου και διακομιστή

Μέσα σε έναν φάκελο με το όνομα data περιέχεται αρχικά ο φάκελος 'Dataset_BUSI_with_GT' ο οποίος περιέχει όλα τα δεδομένα. Με το notebook data_preprocessing.ipynb πραγματοποιείται ο καθαρισμός και η επαύξηση των δεδομένων πριν αυτά τροφοδοτηθούν στο τελικό μοντέλο.

data_preprocessing.ipynb

```
import os
import cv2
import shutil
import random
import numpy as np
import glob

shutil.rmtree("./Dataset_BUSI_with_GT/normal/") #Delete folder normal

# Delete files with "mask" in their name from the benign and malignant directories
for file in glob.glob("./Dataset_BUSI_with_GT/benign/*mask*"):
    os.remove(file)
for file in glob.glob("./Dataset_BUSI_with_GT/malignant/*mask*"):
    os.remove(file)

#Mirror Horizontally all images
folder_path_ben = "./Dataset_BUSI_with_GT/benign/"
folder_path_mal = "./Dataset_BUSI_with_GT/malignant/"

# List all files in the folder
image_files_ben = os.listdir(folder_path_ben)
image_files_mal = os.listdir(folder_path_mal)

# Function to mirror image horizontally
def mirror_image(image_path):
    image = cv2.imread(image_path)
    mirrored_image = cv2.flip(image, 1) # 1 for horizontal flip
    return mirrored_image

# Apply mirror transformation to each image and save
for image_file in image_files_ben:
    image_path = os.path.join(folder_path_ben, image_file)
    mirrored_image = mirror_image(image_path)
    mirrored_image_path = os.path.join(folder_path_ben, "mirrored_" + image_file)
    cv2.imwrite(mirrored_image_path, mirrored_image)

for image_file in image_files_mal:
    image_path = os.path.join(folder_path_mal, image_file)
    mirrored_image = mirror_image(image_path)
    mirrored_image_path = os.path.join(folder_path_mal, "mirrored_" + image_file)
    cv2.imwrite(mirrored_image_path, mirrored_image)
```

```
#Zoom, Lower Contrast, CLAHE, Gaussian Blur and Salt and pepper noise addition functions
#Zoom image on the center function
def zoom_image_center(image, zoom_factor=1.2):
    height, width = image.shape[:2]
    new_height, new_width = int(height / zoom_factor), int(width / zoom_factor)
    top = (height - new_height) // 2
    left = (width - new_width) // 2
    bottom, right = top + new_height, left + new_width
    cropped_image = image[top:bottom, left:right]
    zoomed_image = cv2.resize(cropped_image, (width, height),
interpolation=cv2.INTER_LINEAR)
    return zoomed_image

#Lower Contrast function
def lower_contrast(image_path):
    image = cv2.imread(image_path)
    img_float = image.astype(float)
    b, g, r = cv2.split(img_float)
    factor = 1.6 #Contrast factor (factor < 1 for darker picture, factor > 1 for brighter picture and lower contrast)
    b_low_contrast = b * factor
    g_low_contrast = g * factor
    r_low_contrast = r * factor
    low_contrast_img = cv2.merge((b_low_contrast, g_low_contrast, r_low_contrast))
#Convert image back to [0,255]
    low_contrast_img = cv2.convertScaleAbs(low_contrast_img)
    return low_contrast_img

#Contrast Limited Adaptive Histogram Equalization Function
def clahe_image(image_path):
    image = cv2.imread(image_path)
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    clahe = cv2.createCLAHE(clipLimit=1.5, tileGridSize=(8, 8))
    cl_img = clahe.apply(gray_img)
    return cl_img

#Gaussian Blur Addition function
def add_blur(image_path):
    image = cv2.imread(image_path)
    blurred_img = cv2.GaussianBlur(image, (13, 13), 0)
    return blurred_img

#Salt and Pepper addition function
def add_salt_and_pepper_noise(image_path, salt_prob=0.03, pepper_prob=0.03):
    image = cv2.imread(image_path)
    noisy_img = np.copy(image)
    num_salt = np.ceil(salt_prob * image.size)
    num_pepper = np.ceil(pepper_prob * image.size)
```

```
    coords = [np.random.randint(0, i - 1, int(num_salt)) for i in image.shape]#Salt
noise (white pixels)
    noisy_img[coords[0], coords[1], :] = 1
    coords = [np.random.randint(0, i - 1, int(num_pepper)) for i in
image.shape]#Pepper noise (black pixels)
    noisy_img[coords[0], coords[1], :] = 0
    return noisy_img

#Functions applied on benign images
folder_path = "./Dataset_BUSI_with_GT/benign/"

image_files = os.listdir(folder_path)

selected_images_zoom = random.sample(image_files, 150)
selected_images_contrast = random.sample(image_files, 125)
selected_images_clahe = random.sample(image_files, 125)
selected_images_blur = random.sample(image_files, 125)
selected_images_snp = random.sample(image_files, 125)

for image_file in selected_images_zoom:
    image_path = os.path.join(folder_path, image_file)
    image = cv2.imread(image_path)
    zoomed_image = zoom_image_center(image)
    output_path = os.path.join(folder_path, "zoomed_" + image_file)
    cv2.imwrite(output_path, zoomed_image)

for image_file in selected_images_contrast:
    image_path = os.path.join(folder_path, image_file)
    low_contrast_image = lower_contrast(image_path)
    lower_contrast_image_path = os.path.join(folder_path, "lower_contrast_" +
image_file)
    cv2.imwrite(lower_contrast_image_path, low_contrast_image)

for image_file in selected_images_clahe:
    image_path = os.path.join(folder_path, image_file)
    cl_image = clahe_image(image_path)
    cl_image_path = os.path.join(folder_path, "clahe_" + image_file)
    cv2.imwrite(cl_image_path, cl_image)

for image_file in selected_images_blur:
    image_path = os.path.join(folder_path, image_file)
    blurred_image = add_blur(image_path)
    blurred_image_path = os.path.join(folder_path, "blurred_" + image_file)
    cv2.imwrite(blurred_image_path, blurred_image)

for image_file in selected_images_snp:
    image_path = os.path.join(folder_path, image_file)
    snp_image = add_salt_and_pepper_noise(image_path)
    snp_image_path = os.path.join(folder_path, "snp_" + image_file)
    cv2.imwrite(snp_image_path, snp_image)
```

```
#Functions applied on malignant images
folder_path = "./Dataset_BUSI_with_GT/malignant/"

image_files = os.listdir(folder_path)

selected_images_zoom = random.sample(image_files, 315)
selected_images_contrast = random.sample(image_files, 190)
selected_images_clahe = random.sample(image_files, 190)
selected_images_blur = random.sample(image_files, 190)
selected_images_snp = random.sample(image_files, 190)

for image_file in selected_images_zoom:
    image_path = os.path.join(folder_path, image_file)
    image = cv2.imread(image_path)
    zoomed_image = zoom_image_center(image)
    output_path = os.path.join(folder_path, "zoomed_" + image_file)
    cv2.imwrite(output_path, zoomed_image)

for image_file in selected_images_contrast:
    image_path = os.path.join(folder_path, image_file)
    low_contrast_image = lower_contrast(image_path)
    lower_contrast_image_path = os.path.join(folder_path, "lower_contrast_" +
image_file)
    cv2.imwrite(lower_contrast_image_path, low_contrast_image)

for image_file in selected_images_clahe:
    image_path = os.path.join(folder_path, image_file)
    cl_image = clahe_image(image_path)
    cl_image_path = os.path.join(folder_path, "clahe_" + image_file)
    cv2.imwrite(cl_image_path, cl_image)

for image_file in selected_images_blur:
    image_path = os.path.join(folder_path, image_file)
    blurred_image = add_blur(image_path)
    blurred_image_path = os.path.join(folder_path, "blurred_" + image_file)
    cv2.imwrite(blurred_image_path, blurred_image)

for image_file in selected_images_snp:
    image_path = os.path.join(folder_path, image_file)
    snp_image = add_salt_and_pepper_noise(image_path)
    snp_image_path = os.path.join(folder_path, "snp_" + image_file)
    cv2.imwrite(snp_image_path, snp_image)

# Train Validation and Test sets creation
base_dir = os.path.join(os.path.expanduser('~'), 'Desktop', 'data',
'Dataset_BUSI_with_GT')
benign_dir = os.path.join(base_dir, 'benign')
malignant_dir = os.path.join(base_dir, 'malignant')
```

```
train_dir = os.path.join(base_dir, 'train')
val_dir = os.path.join(base_dir, 'val')
test_dir = os.path.join(base_dir, 'test')

# Subdirectories for benign and malignant
for category in ['benign', 'malignant']:
    os.makedirs(os.path.join(train_dir, category), exist_ok=True)
    os.makedirs(os.path.join(val_dir, category), exist_ok=True)
    os.makedirs(os.path.join(test_dir, category), exist_ok=True)

# Function to split images on a 70-15-15 percentage and copy them on the train, val
and test folders
def split_and_copy_images(category, src_dir, train_dir, val_dir, test_dir,
train_split=0.7, val_split=0.15, test_split=0.15):
    images = glob.glob(os.path.join(src_dir, '*')) # * to include all files in the
directory
    random.shuffle(images) #Shuffle images randomly
    total_images = len(images)
    train_count = int(total_images * train_split)
    val_count = int(total_images * val_split)
    train_images = images[:train_count]
    val_images = images[train_count:train_count + val_count]
    test_images = images[train_count + val_count:]

    for img in train_images:
        shutil.copy(img, os.path.join(train_dir, category))

    for img in val_images:
        shutil.copy(img, os.path.join(val_dir, category))

    for img in test_images:
        shutil.copy(img, os.path.join(test_dir, category))

# Split and copy benign images
split_and_copy_images('benign', benign_dir, train_dir, val_dir, test_dir)

# Split and copy malignant images
split_and_copy_images('malignant', malignant_dir, train_dir, val_dir, test_dir)

print("Images have been split and copied on the folders successfully.")
```

Στην συνέχεια δημιουργείται και το notebook nasnet.ipynb στο οποίο υλοποιείται το μοντέλο.

nasnet.ipynb

```
import os
import shutil
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import NASNetLarge
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay,
roc_auc_score, roc_curve, auc
import seaborn as sns

base_dir = 'Dataset_BUSI_with_GT'
train_dir = os.path.join(base_dir, 'train')
val_dir = os.path.join(base_dir, 'val')
test_dir = os.path.join(base_dir, 'test')

#Normalization
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

#Train,Val and Test sets
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(331, 331),
    batch_size=32,
    class_mode='binary'
)

val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=(331, 331),
    batch_size=32,
    class_mode='binary'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(331, 331),
    batch_size=32,
    class_mode='binary',
    shuffle=False
)

base_model = NASNetLarge(weights='imagenet', include_top=False, input_shape=(331,
331, 3))
```



```
#Final layers set
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
output_layer = Dense(1, activation='sigmoid')(x)

model = Model(inputs=base_model.input, outputs=output_layer)

#Freeze early layers
for layer in base_model.layers:
    layer.trainable = False
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy',
metrics=['accuracy'])

#Start training
history = model.fit(
    train_generator,
    epochs=10,
    validation_data=val_generator
)

#Test accuracy
test_loss, test_acc = model.evaluate(test_generator)
print(f'Test accuracy: {test_acc}')

#Accuracy, Val accuracy, Loss and val Loss graphs
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']

epochs_range = range(10)

plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.plot(epochs_range,acc,label='Accuracy')
plt.plot(epochs_range,val_acc,label="Validation Accuracy")
plt.legend()

plt.subplot(1,2,2)
plt.plot(epochs_range,loss,label='Loss')
plt.plot(epochs_range,val_loss,label="Validation Loss")
plt.legend()
plt.show()

#Print Predictions on test data
predictions = model.predict(test_generator)
binary_predictions = (predictions > 0.5).astype(int).flatten()
true_labels = test_generator.classes
```

```
print("Predictions for test data (Predicted, True Label):")
for i in range(len(true_labels)):
    print(f"{binary_predictions[i]}, {true_labels[i]}")

#Confusion matrix plot
cm = confusion_matrix(true_labels, binary_predictions)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Benign',
'Malignant'], yticklabels=['Benign', 'Malignant'])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()

#Save the trained model
model.save('nasnet')
```

Τα βάρη του μοντέλου έχουν κάνει export σε έναν νέο φάκελο με το όνομα “nasnet”.

Ο διακομιστής της εφαρμογής πρέπει να δημιουργηθεί εντός του φακέλου που περιλαμβάνεται ο φάκελος "nasnet". Αυτό επιτρέπει τη φόρτωση των βαρών κατά την εκκίνηση του διακομιστή. Η υλοποίηση του διακομιστή βρίσκεται στο αρχείο "predict_app.py" και εκτελείται στην πόρτα 5000 του υπολογιστή του χρήστη.

predict_app.py

```
from flask import Flask, request, jsonify
from flask_cors import CORS
import base64
import io
from PIL import Image
from keras.models import load_model
from keras.preprocessing.image import img_to_array
import numpy as np

app = Flask(__name__)
CORS(app) # Enable CORS for all routes

def get_model():
    global model
    model = load_model('nasnet')
    print("Model loaded successfully!")
#Necessary image preprocessing function
#Adjust size based on model needs (331x331 for NASNet)
def preprocess_image(image, target_size=(331, 331)):
    if image.mode != "RGB":
        image = image.convert("RGB")
    image = image.resize(target_size)
    img_array = img_to_array(image)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0
```

```
    return img_array

print("Loading Keras model...")
get_model()
#POST Route
@app.route("/predict", methods=["POST"])
def predict():
    try:
        message = request.get_json(force=True)
        print("Received JSON Data:", message)

        encoded = message['image'] #Encoded received image data
        print("Received Image Data:", encoded[:50])

        decoded = base64.b64decode(encoded) #Base64 image decoding
        image = Image.open(io.BytesIO(decoded))
        processed_image = preprocess_image(image, target_size=(331, 331))
        print("Image Preprocessing Successful") # Log successful preprocessing

        prediction = model.predict(processed_image)

        print("Prediction Result:", prediction) # Log prediction result
        # Convert prediction array to a Python List before serializing to JSON
        prediction_list = prediction.tolist()

        response = {
            'prediction': prediction_list
        }
        return jsonify(response)

    except Exception as e:
        print("Error:", str(e))
        return jsonify({'error': 'Internal Server Error'}), 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Η τελική μορφή του φακέλου data θα είναι η εξής:

```
data/
├── Dataset_BUSI_with_GT
│   ├── malignant
│   ├── benign
│   ├── train
│   ├── val
│   └── test
├── nasnet
├── data_preprocessing.ipynb
└── nasnet.ipynb
```

Οι κώδικες για την υλοποίηση των παραπάνω μπορούν να βρεθούν και στο Github Repository: <https://github.com/christopherkormpos/TumorNet-thesis> μέσα στον φάκελο tumor-net-backend.

Παράρτημα Β – Κώδικας διαδικτυακής εφαρμογής

Ο ολοκληρωμένος κώδικας για την υλοποίηση όλων των σελίδων της διαδικτυακής εφαρμογής υπάρχει και στο GitHub repository: <https://github.com/christopherkormpos/TumorNet-thesis> μέσα στον φάκελο tumornet-webapp.

Παρακάτω παρουσιάζεται ο κώδικας μόνο για την λειτουργία της σελίδας της εφαρμογής και χωρίς CSS.

```
tumornet-webapp/  
├─ node_modules  
├─ public  
│   └─ index.html  
├─ src  
│   └─ components  
│       └─ Pages  
│           └─ Application.jsx  
│   └─ App.js  
│   └─ index.js  
└─ package.json
```

index.html

```
<!DOCTYPE html>  
<html Lang="en">  
  <head>  
    <meta charset="utf-8" />  
    <link rel="icon" href="./images/logo_light.png">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <meta  
      name="description"  
      content="Web site created using create-react-app"/>  
    <title>TumorNet</title>  
  </head>  
  <body>  
    <div id="root"></div>  
  </body>  
</html>
```

index.js file

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import App from './App';  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>  
);
```

App.js file

```
import * as React from "react";
import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
import { Application } from "../components/Pages/Application";

export default function App() {
  return (
    <>
      <Router>
        <div className="pages">
          <Routes>
            <Route path="/" element={<Application />} />
          </Routes>
        </div>
      </Router>
    </>
  );
}
```

Application.jsx

```
import React, { useState, useEffect } from 'react';
import '../application.css'
import PredictionChart from '../components/PredictionChart';

export function Application() {
  //UseStateHooks initialization
  const [selectedImage, setSelectedImage] = useState('');
  const [isLoading, setIsLoading] = useState(false);
  const [base64Image, setBase64Image] = useState('');
  const [prediction, setPrediction] = useState(null);
  const [isPredicting, setIsPredicting] = useState(false);

  // Handle image selection and conversion to base64
  const handleImageChange = () => {
    setIsLoading(true);
    setPrediction(null); // Reset prediction
    const reader = new FileReader();
    reader.onload = function (e) {
      const dataURL = reader.result;
      const base64 = dataURL.replace("data:image/png;base64,", "");
      setSelectedImage(dataURL); // Update selectedImage state with dataURL
      setBase64Image(base64); // Update base64Image state
      console.log(base64);
      // Set loading state to false after 3 seconds then scroll to prediction
      // button automatically
      setTimeout(() => {
        setIsLoading(false);
        const imageSelectedElement = document.getElementById('predict-button');
        if (imageSelectedElement) {

```

```
const { top } = imageSelectedElement.getBoundingClientRect();
window.scrollTo({
  top: window.scrollY + top,
  behavior: 'smooth'
});
}
}, 3000);
};

if (document.getElementById('image-selector').files.length > 0) {
  reader.readAsDataURL(document.getElementById('image-selector').files[0]); //
  Read selected file as DataURL
}
};
// Add and remove event listener for image input
useEffect(() => {
  document.getElementById('image-selector').addEventListener('change',
handleImageChange);

  return () => {
    document.getElementById('image-selector').removeEventListener('change',
handleImageChange);
  };
}, []);
// Handle prediction function
const handlePrediction = () => {
  setIsPredicting(true); // Set predicting state to true
  let message = {
    image: base64Image
  };

  console.log(message); //Prints the base64 image (debugging purposes only)
  // Send data to server. Based on the port running the predict_app.py (Here is
  Local host port 5000 /predict)
  fetch("http://127.0.0.1:5000/predict", {
    method: 'POST',
    body: JSON.stringify(message),
  })
  .then(response => response.json())
  .then(data => {
    setTimeout(() => {
      setPrediction(data.prediction); // Update prediction state to true
      console.log(data); //Print the prediction (number between 0 and 1).
      Debugging purposes only
      setIsPredicting(false); // Update prediction state to false
    }, 3000);
  })
  .catch(error => {
    console.error('Error:', error);
    setIsPredicting(false); //If an error occurs set predicting state to false
```

```
});
};
// Component to display prediction result based on the data.prediction. Can also
// be sepereted to the components folder
//but because it is only used in the Application.jsx it is written inside here.
const ValueDisplay = ({ value }) => {
  const isMalignant = value > 0.5;
  return (
    <div>
      {isMalignant ? (
        <div className="malignant">
          <h2 className='malignant'>
            The model predicts with<span className='conf'>high
confidence</span>&nbsp;<span className='highlight-mal'>MALIGNANT.</span>
          </h2>
        </div>
      ) : (
        <div className="benign">
          <h2 className='benign'>
            The model predicts with<span className='conf'>high
confidence</span>&nbsp;<span className='highlight-ben'>BENIGN.</span>
          </h2>
        </div>
      )}
    </div>
  );
};
//Render the Application UI
return (
  <div>
    <h3>Please select an image file (supported types: <span
className='h3highlight'>PNG</span>).</h3>
    <div className='upper'>
      <div className='input'>
        <input
          type="file"
          id="image-selector"
          accept="image/png"
          onChange={handleImageChange}
        />
      </div>
    </div>
    <div id="prediction">
      <div className='image-loader'>
        {isLoading && <div className="loader"></div>}
        {(selectedImage && !isLoading) && <img src={selectedImage} alt="Selected"
/>}
      </div>
    </div>
  </div>
);
```

```
    {(selectedImage && !isLoading) &&
      <div className='button'>
        <button className='predict-button' id="predict-button"
onClick={handlePrediction}>
          <span>
            Predict
          </span>
        </button>
      </div>
    }
    <div className='prediction-loader-div'>
      {isPredicting && <div className="prediction-loader"></div>}
    </div>
    {(!isPredicting && prediction) && (
      <>
        <ValueDisplay value={prediction[0]} />
        <PredictionChart prediction={prediction} />
      </>
    )}
  </div>
</div>
);
}
```

package.json

```
{
  "name": "tumor-net",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@mui/material": "^5.15.19",
    "@mui/x-charts": "^7.6.2",
    "@testing-library/jest-dom": "^5.17.0",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "bootstrap": "^5.3.3",
    "chart.js": "^4.4.3",
    "react": "^18.2.0",
    "react-bootstrap": "^2.10.2",
    "react-chartjs-2": "^5.2.0",
    "react-dom": "^18.2.0",
    "react-google-charts": "^4.0.1",
    "react-router-dom": "^6.22.3",
    "react-scripts": "5.0.1",
    "styled-components": "^6.1.8",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
  }
}
```



```
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "@babel/plugin-proposal-private-property-in-object": "^7.21.11"
  }
}
```

Με τον παραπάνω κώδικα γραμμένο σε React και τα `node_modules` εγκατεστημένα, ο χρήστης μπορεί να εκκινήσει την εφαρμογή χρησιμοποιώντας την εντολή `npm start`, και να τη δει ζωντανά στην πόρτα 3000.