



Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ψηφιακή επεξεργασία και κατάτμηση βιοϊατρικών εικόνων
με χρήση τεχνητών νευρωνικών δικτύων»



Φοιτητής: Κύριλλος Περιστενίδης
Α.Μ.: 47345

Επιβλέπουσα καθηγήτρια:
Μαρία Ραγκούση

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΦΕΒΡΟΥΑΡΙΟΣ 2023

Ψηφιακή επεξεργασία και κατάτμηση βιοϊατρικών εικόνων με χρήση τεχνητών νευρωνικών



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

Diploma Thesis

**«Digital processing and segmentation of biomedical images
using artificial neural networks»**



**Student: Kyrillos Peristenidis
Registration Number: 47345**

**Supervisor
Prof. Maria Rangoussi**

ATHENS-EGALEO, FEBRUARY 2023

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

Μαρία Ραγκούση, Καθηγήτρια (Επιβλέπουσα)	Στυλιανός Ποτηράκης, Καθηγητής	Νικόλαος-Αλέξανδρος Τάτλας, Αν. Καθηγητής

Copyright © Με επιφύλαξη παντός δικαιώματος. All rights reserved.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Περιστενίδης Κύριλλος, Φεβρουάριος 2023

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 Περιστενίδης Κύριλλος του Χαράλαμπου, με αριθμό μητρώου 47345 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας της παρούσας διπλωματικής εργασίας και οποιαδήποτε βοήθεια είχα για την εκπόνηση της είναι αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι πηγές από τις οποίες άντλησα δεδομένα, 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 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



Περιστενίδης Κύριλλος

Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας και παράλληλα του προπτυχιακού μου κύκλου, επιθυμώ να εκφράσω τις ευχαριστίες μου σε όσους με στήριξαν όλα αυτά τα χρόνια. Πρωτίστως επιθυμώ να εκφράσω τις ευχαριστίες μου σε όλο το διδακτικό προσωπικό με το οποίο ήμουν στην ευχάριστη θέση να συναναστραφώ και να απορροφήσω τη γνώση που μου προσέφερε, και να διαβεβαιώσω όλους αυτούς τους ανθρώπους πως επιδιώκω να εκμεταλλευτώ με τον καλύτερο τρόπο όλη αυτή τη γνώση που έλαβα. Ξεχωριστή και ειδικότερη αναφορά όμως επιθυμώ να κάνω στην επιβλέπουσα καθηγήτριά μου στην παρούσα διπλωματική εργασία κ. Μαρία Ραγκούση, την οποία επιθυμώ να ευχαριστήσω, καθώς με εμπιστεύθηκε, μου συμπαραστάθηκε και με καθοδήγησε κατά την εκπόνηση αυτής της διπλωματικής εργασίας, παρά τις δυσμενείς συνθήκες. Τέλος δεν θα μπορούσα να πορευθώ χωρίς την ιδιαίτερη στήριξη της οικογένειάς μου, στην οποία επιθυμώ να αφιερώσω την παρούσα εργασία.

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο μελέτης τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ) και τη χρήση τους με σκοπό την κατάτμηση βιοϊατρικών εικόνων. Θα μπορούσαμε να διακρίνουμε δύο διαφορετικά πεδία ενδιαφέροντος. Στο πρώτο μέρος γίνεται εκτενής αναφορά στα νευρωνικά δίκτυα. Συγκεκριμένα, πραγματοποιείται αρχικά μία σύντομη σύγκριση των βιολογικών νευρωνικών δικτύων και των τεχνητών νευρωνικών δικτύων. Ακολούθως περιγράφονται η μορφή, η λειτουργία και οι εφαρμογές απλών ΤΝΔ, αλλά και βαθιών ΤΝΔ αντίστοιχα, ενώ δίνεται περισσότερη έμφαση στα συνελκτικά ΝΔ, μία κατηγορία βαθιών ΤΝΔ που θα μας απασχολήσει και στην υλοποίησή μας στη συνέχεια. Όσον αφορά στο δεύτερο μέρος, εδώ εμβαθύνουμε στον τομέα της εικόνας και της ψηφιακής της επεξεργασίας της. Αναλυτικότερα, παραθέτουμε τις βασικές έννοιες που συνδέονται με την εικόνα, παρουσιάζουμε ένα σύνολο μετασχηματισμών, τεχνικών βελτίωσης, αποκατάστασης και κατάτμησης αυτής, μέσω αντίστοιχων παραδειγμάτων. Τέλος, στο πειραματικό μέρος της εργασίας, κάνουμε χρήση ενός συνόλου θεωρητικών γνώσεων, που αποκτήθηκαν κατά την εκπόνηση της εργασίας αυτής και υλοποιούμε δύο διαφορετικά συνελκτικά νευρωνικά δίκτυα, με στόχο την κατάτμηση (segmentation) ενός συνόλου βιοϊατρικών εικόνων. Κατά την εκπαίδευση των δικτύων μεταβάλλουμε κάποιες βασικές παραμέτρους και διαπιστώνουμε κατά πόσον η κατάτμηση είναι επιτυχής μέσω του εκάστοτε δικτύου.

Λέξεις-κλειδιά

Τεχνητά νευρωνικά δίκτυα, βαθιά μάθηση, επεξεργασία εικόνας, κατάτμηση εικόνας, συνελκτικά νευρωνικά δίκτυα, νευρώνας, βιοϊατρικές εικόνες

Abstract

The present thesis focuses on the study area of artificial neural networks and their use for the purpose of segmentation of biomedical images. We could distinguish two different areas of interest. The first part makes extensive reference to neural networks. In particular, a brief comparison of biological neural networks and artificial neural networks is made. Subsequently, we describe the form, the function and the applications of simple ANN, but also of deep ANN respectively, while more emphasis is given to the convolutional NN, a category of deep ANN that will concern us in our implementation later. In the second part of this thesis, we focus in the field of image and its digital processing. In more detail, we list the basic concepts associated with the image, we present a variety of transformations, techniques for its improvement, restoration and segmentation, through corresponding examples. Finally, in the experimental part, we use a set of theoretical knowledge, acquired during the elaboration of this thesis and we implement two different types of convolutional neural networks, with the aim of segmenting a set of biomedical images. During the training of the networks we vary certain basic parameters and observe and measure the success of the segmentation process through the respective network.

Keywords

Artificial neural networks, deep learning, image processing, image segmentation, convolutional neural networks, neuron, biomedical images

Περιεχόμενα

Κατάλογος Εικόνων	12
Εισαγωγή	16
1. Νευρωνικά Δίκτυα	17
1.1 Ο Βιολογικός Νευρώνας	17
1.2 Τεχνητά Νευρωνικά δίκτυα	17
1.2.1 Ο τεχνητός Νευρώνας.....	18
1.2.2 Δομή των Τεχνητών Νευρωνικών Δικτύων.....	18
1.2.3 Διαφορές Βιολογικών και Τεχνητών Νευρωνικών Δικτύων	20
1.3 Εφαρμογές και προβλήματα προς επίλυση με ΤΝΔ.	22
1.3.1 Εφαρμογές Τεχνητών Νευρωνικών Δικτύων	22
1.3.2 Προβλήματα προς επίλυση με ΤΝΔ	23
1.4 Μάθηση Τεχνητών Νευρωνικών Δικτύων.....	23
1.5 Συναρτήσεις Ενεργοποίησης	24
1.5.1 Συνάρτηση δυαδικού βήματος.....	25
1.5.2 Γραμμική συνάρτηση.....	25
1.5.3 Σιγμοειδής ή Λογιστική Συνάρτηση	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
1.5.4 Συνάρτηση ανόρθωσης (ReLU).....	26
1.5.5 Διαρρέουσα Συνάρτηση Ανόρθωσης (Leaky RELU).....	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
1.5.6 Συνάρτηση SoftMax	26
1.6 Κυριότερες Αρχιτεκτονικές ΤΝΔ	27
1.6.1 Single Layer Perceptron.....	27
1.6.2 Multilayer Perceptrons (MLP).....	28
1.6.3 Radial Basis Function Νευρωνικό Δίκτυο (RBFNN).....	28
1.6.5 Learning Vector Quantization (LVQ):.....	29
1.7 Παράδειγμα προβλήματος ΤΝΔ και επίλυση στο MATLAB.....	29
1.7.1 Προσέγγιση Συνάρτησης Radial Basis	29
2. Βαθιά Μάθηση (Deep Learning)	31
2.1 Βαθιά Νευρωνικά Δίκτυα	31
2.3 Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)	32
2.3.1 Δημοφιλή συνελκτικά Νευρωνικά Δίκτυα	36
3. Ψηφιακή Επεξεργασία Εικόνας	39
3.1 Ιστορική αναδρομή της ψηφιακής επεξεργασίας εικόνας	39
3.2 θεμελιώδεις έννοιες.....	40

3.2.1 Ψηφιακή Εικόνα	40
3.2.2 Εικονοστοιχείο (Pixel).....	41
3.4 Χαρακτηριστικά Ψηφιακής Εικόνας	41
3.4.1 Βάθος Χρώματος	41
3.4.2 Ανάλυση.....	44
3.4.3 Υφή	45
3.4.4 Τύπος αρχείου εικόνας.....	45
3.5 Χρωματικά Συστήματα	46
3.5.1 RGB	46
3.5.2 CMY	47
3.5.3 HSI.....	47
3.6 Ιστόγραμμα	48
4. Μετασχηματισμοί Εικόνας	49
4.1 Μετασχηματισμός Fourier	49
4.2 Μετασχηματισμός DCT.....	51
4.3 Μετασχηματισμός RADON.....	52
5. Βελτιστοποίηση Εικόνας	54
5.1 Τεχνικές Σημείου	54
5.1.1 Αντιστροφή Φωτεινότητας	54
5.1.2 Επέκταση της Αντίθεσης	55
5.1.3 Κόψιμο Φωτεινότητας	56
5.2 Τεχνικές Χώρου	56
5.2.1 Laplacian filter	57
5.2.2 Φίλτρο Μεσαίας Τιμής	58
6. Αποκατάσταση Εικόνας.....	60
6.1 Μορφές θορύβου	60
6.1.1 Gaussian ή κανονικός θόρυβος.....	60
6.1.2 Ομοιόμορφος θόρυβος (Uniform noise).....	61
6.2 Φίλτρα Εξάλειψης Θορύβου.....	62
6.2.1 Φίλτρο μέσης τιμής (Arithmetic Mean Filter)	62
6.2.2 Φίλτρο μέγιστης τιμής (Maximum Filter)	64
6.2.2 Φίλτρο ελάχιστης τιμής (Minimum Filter)	65
7. Κατάτμηση Εικόνας.....	67
7.1 Κατάτμηση με κατωφλίωση	68
7.1.1 Κατάτμηση με απλή κατωφλίωση	68
7.1.2 Κατάτμηση με κατωφλίωση Otsu.....	69

7.2 Κατάτμηση Βάσει Περιοχής (Region Based Segmentation)	71
7.2.1 Κατάτμηση με ανάπτυξη περιοχής (Region Growing Segmentation).....	71
7.2.2 Κατάτμηση με διαχωρισμό και συγχώνευση(Region Splitting and Merging)	72
7.3 Κατάτμηση με μέθοδο Watershed	73
7.4 Κατάτμηση με Αλγόριθμους Ταξινόμησης (Clustering-Based Segmentation).....	76
8. Κατάτμηση Βιοϊατρικών Εικόνων μέσω Συνελκτικών Νευρωνικών Δικτύων.	79
8.1 Συνοπτική περιγραφή.....	79
8.2 Στόχος	79
8.3 Θεωρητικό Πλαίσιο Πειράματος	79
8.3.1 Matlab και Τύποι Εικόνων στο Matlab.....	79
8.3.2 Ορμή (Momentum)	82
8.3.3 Ρυθμός Μάθησης (Learning Rate).....	82
8.3.4 Epoch και Iteration.....	83
8.3.5 Mini-batch.....	83
8.4 Προ-επεξεργασία Dataset	84
8.4.1 Dataset και resizing.....	84
8.4.1 Labeling του dataset.....	84
8.5 Υλοποίηση	87
8.5.1 Αρχιτεκτονικές.....	87
8.5.2 Εκπαίδευση του Unet.....	92
8.5.3 Έλεγχος του Unet με γνωστά δεδομένα.....	97
8.5.4 Έλεγχος του Unet με άγνωστα δεδομένα	99
8.5.5 Εκπαίδευση του απλού CNN	102
8.5.6 Έλεγχος του απλού CNN με γνωστά δεδομένα.....	107
8.5.7 Έλεγχος του απλού CNN με άγνωστα δεδομένα.....	110
9. Σύγκριση αποτελεσμάτων και συμπεράσματα	114
9.1 Σύγκριση αποτελεσμάτων.....	114
9.2 Συμπεράσματα	114
Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές.....	116
Παραρτήματα	119

Κατάλογος Εικόνων

Εικόνα 1.1: Βαθμίδες νευρικού συστήματος.....	17
Εικόνα 1.2: Ανατομία βιολογικού νευρώνα	Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.
Εικόνα 1.4: Δομή Τεχνητού Νευρωνικού δικτύου	19
Εικόνα 1.5: Συνάρτηση ενεργοποίησης.....	24
Εικόνα 1.6: Συνάρτηση ενεργοποίησης δυαδικού βήματος	25
Εικόνα 1.7: Γραμμική συνάρτηση ενεργοποίησης	25
Εικόνα 1.9: Συνάρτηση ανόρθωσης.....	26
Εικόνα 1.11: Συνάρτηση SoftMax.....	27
Εικόνα 1.12: Single Layer Perceptron	27
Εικόνα 1.13: Multilayer Perceptron.....	28
Εικόνα 1.14: Radial Basis Function Νευρωνικό Δίκτυο	28
Εικόνα 1.15: Ορισμός εισόδων και στόχων	29
Εικόνα 1.16: Συνάρτηση μεταφοράς ακτινικής βάσης.....	30
Εικόνα 1.17: Προσεγγιστική συνάρτηση basis function.	30
Εικόνα 2.1: Νευρωνικό δίκτυο (a) με 1 hidden layer και (b) με 4 hidden layers	31
Εικόνα 2.2: Παράδειγμα ταξινόμησης εικόνας.....	32
Εικόνα 2.3: Γκρι εικόνα υπό την αντίληψη του υπολογιστή.....	33
Εικόνα 2.4: Έγχρωμη εικόνα υπό την αντίληψη του υπολογιστή	33
Εικόνα 2.5: Στρώματα συνελκτικού ΤΝΔ	34
Εικόνα 2.6: Convolutional layer	34
Εικόνα 2.7: Alexnet	36
Εικόνα 2.8: VGG-16.....	36
Εικόνα 2.9: Googlenet	37
Εικόνα 2.10: Unet	38
Εικόνα 3.1: Τυπικό σύστημα παραγωγής ψηφιακής εικόνας	40
Εικόνα 3.2: Παράδειγμα εικονοστοιχείων	41
Εικόνα 3.3: 1 bit βάθος χρώματος / 2 χρώματα.....	42
Εικόνα 3.4: 2 bit βάθος χρώματος / 4 χρώματα.....	42
Εικόνα 3.5: 4 bit βάθος χρώματος / 16 χρώματα.....	43
Εικόνα 3.6: 8 bit βάθος χρώματος / 256 χρώματα.....	43
Εικόνα 3.7: 24 bit βάθος χρώματος / 16.777.216 χρώματα.....	44
Εικόνα 3.8: 24 Μείωση από 300dpi σε 150dpi , 72 dpi και 36 dpi	44
Εικόνα 3.9: Τεχνητά δημιουργημένη υφή (αριστερά) και υφή σε φυσική σκηνή (δεξιά).....	45
Εικόνα 3.10: Μοντέλο RGB	47
Εικόνα 3.11: Μοντέλο CMYK	47
Εικόνα 3.12: Μοντέλο HIS.....	48
Εικόνα 3.3.13: Εικόνα κυττάρου και το αντίστοιχο ιστόγραμμα της.....	48
Εικόνα 4.4.1: Ορθογωνική συνάρτηση $f(m,n)$ (αριστερά) , διακριτή εικόνα (δεξιά).....	50
Εικόνα 4.4.2: DFT της f χωρίς zero padding (αριστερά) , με zero padding (δεξιά).....	50
Εικόνα 4.4.3: DFT της ορθογωνικής $f(m,n)$	50
Εικόνα 4.4.4: Αρχική εικόνα και της ανακατασκευασμένη εικόνα με DCT.....	52
Εικόνα 4.4.5: Γεωμετρία Radon	52
Εικόνα 4.4.6: Εικόνα για Radon μετασχηματισμό	53
Εικόνα 4.4.7: Αποτέλεσμα Radon μετασχηματισμού.....	53
Εικόνα 5.5.1: Αποτέλεσμα Αντιστροφής φωτεινότητας.....	55
Εικόνα 5.5.2: Εικόνα πριν την επέκταση της αντίθεσης και το αντίστοιχο ιστόγραμμα	55

Εικόνα 5.5.3: Εικόνα μετά την επέκταση της αντίθεσης και το αντίστοιχο ιστόγραμμα.....	56
Εικόνα 5.5.4: Εικόνα 'circuit.tif' πριν (αριστερά) και μετά (δεξιά)το κόψιμο φωτεινότητας	56
Εικόνα 5.5.5: Τοποθέτηση μάσκας για βελτίωση εικόνας στο πεδίου του χώρου	57
Εικόνα 5.5.6: Τονισμένες ακμές , αποτέλεσμα εφαρμογής Laplacian filter	58
Εικόνα 5.5.7: Εικόνα 'trees.tif' με πυκνότητα θορύβου 0,2.....	58
Εικόνα 5.5.8: Χρήση median filter μάσκας 3x3 και χρήση median filter μάσκας 7x7	59
Εικόνα 6.6.1: Κατανομή Gaussian noise	61
Εικόνα 6.6.2: Κατανομή uniform noise	61
Εικόνα 6.6.3: Κατανομή impulse noise	62
Εικόνα 6.6.4: (α) Chest MRI, (β) Chest MRI με κανονικό θόρυβο, (γ)Average filter	63
Εικόνα 6.6.5: (α) Chest MRI, (β) Chest MRI με σημειακό θόρυβο, (γ)Average filter	63
Εικόνα 6.6.6: (α) Chest MRI, (β) Chest MRI με κανονικό θόρυβο, (γ)Max filter	64
Εικόνα 6.6.7: (α) Chest MRI, (β) Chest MRI με σημειακό θόρυβο, (γ)Max filter.....	64
Εικόνα 6.6.8: (α) Chest MRI, (β) Chest MRI με κανονικό θόρυβο, (γ)Min filter.....	65
Εικόνα 6.6.9: (α) Chest MRI, (β) Chest MRI με σημειακό θόρυβο, (γ)Min filter	65
Εικόνα 7.7.1: Κατάτμηση εικόνας με απλή κατωφλίωση και κατώφλι $T=0,5$	69
Εικόνα 7.7.2: Ιστόγραμμα εικόνας 'Redbloodcells.jpg'.....	69
Εικόνα 7.7.3: Κατάτμηση εικόνας 'Redbloodcells.jpg' με Otsu κατωφλίωση.....	70
Εικόνα 7.7.4: Κατάτμηση εικόνας 'printedtext.png' με προσαρμοσμένη κατωφλίωση	70
Εικόνα 7.7.5: Κατάτμηση εικόνας 'printedtext.png' με Otsu κατωφλίωση	71
Εικόνα 7.7.6: Κατάτμηση εικόνας 'Redbloodcells.jpg' με 5 seeds.....	72
Εικόνα 7.7.7: Κατάτμηση εικόνας 'Redbloodcells.jpg' με 5 seeds	72
Εικόνα 7.7.8: Κατάτμηση εικόνας με διαχωρισμό και συγχώνευση	73
Εικόνα 7.7.9: Εικόνα ερυθρών κυττάρων σε rgb και grayscale	74
Εικόνα 7.7.10: Gradient magnitude	74
Εικόνα 7.7.11: Εντοπισμός τοπικών μεγίστων	75
Εικόνα 7.7.12: Εφαρμογή εσωτερικών markers	75
Εικόνα 7.7.13: Εντοπισμός αντικειμένων παρασκηίου	75
Εικόνα 7.7.14: Εξωτερικοί markers.....	76
Εικόνα 7.7.15: Αποτέλεσμα Watershed.....	76
Εικόνα 7.7.16: Ιστός με χρώση του με αιμοτοξυλίνη και ηωσίνη.....	77
Εικόνα 7.17: Εφαρμογή των labels των ομάδων	78
Εικόνα 7.18: Ομάδα πρώτη (πάνω αριστερά), Ομάδα δεύτερη (πάνω δεξιά), Ομάδα τρίτη (κάτω αριστερά), Μπλε πυρήνες (κάτω δεξιά).....	78
Εικόνα 8.1: Παράδειγμα ανάγνωσης και απεικόνισης μίας εικόνας	80
Εικόνα 8.2: Παράδειγμα ανάγνωσης και απεικόνισης μίας binary εικόνας.....	81
Εικόνα 8.3: Παράδειγμα ανάγνωσης και απεικόνισης μίας RGB εικόνας	81
Εικόνα 8.4: Παράδειγμα ανάγνωσης και απεικόνισης μίας εικόνας χρωματικού δείκτη.....	81
Εικόνα 8.5: Στοχαστική Κατάβαση Κλίσης με και χωρίς ορμή	82
Εικόνα 8.6: Μεγάλος και μικρός ρυθμός μάθησης.....	83
Εικόνα 8.7: Παραδείγματα ακτινογραφιών θώρακος	84
Εικόνα 8.8: Παράδειγμα μάσκας	85
Εικόνα 8.9: Εισαγωγή εικόνας στο pixel labeler	85
Εικόνα 8.10: Ορισμός των labels.....	86
Εικόνα 8.11: Δημιουργία μάσκας	86
Εικόνα 8.12: Απεικόνιση Unet στο Matlab	87
Εικόνα 8.13: Unet layers 1-19	88
Εικόνα 8.14: Unet layers 20-38	89
Εικόνα 8.15: Unet layers 40-58	89

Εικόνα 8.16: Απεικόνιση απλού CNN στο Matlab.....	91
Εικόνα 8.17: Layers απλού CNN.....	92
Εικόνα 8.18: Accuracy και loss εκπαίδευσης Unet με mini-batch size=1.....	94
Εικόνα 8.19: Αρχικά iterations εκπαίδευσης Unet με mini-batch size=1.....	95
Εικόνα 8.20: Accuracy και loss εκπαίδευσης Unet με mini-batch size=3.....	96
Εικόνα 8.21: Αρχικά iterations εκπαίδευσης Unet με mini-batch size=3.....	96
Εικόνα 8.22: Γνωστό δείγμα ακτινογραφίας πνεύμονα.....	97
Εικόνα 8.23: Αποτέλεσμα κατάτμησης	97
Εικόνα 8.24: Ιδανικό αποτέλεσμα	97
Εικόνα 8.25: Γνωστό δείγμα ακτινογραφίας πνεύμονα.....	98
Εικόνα 8.26: Αποτέλεσμα κατάτμησης	98
Εικόνα 8.27: Ιδανικό αποτέλεσμα	99
Εικόνα 8.28: Άγνωστο δείγμα ακτινογραφίας πνεύμονα	100
Εικόνα 8.29: Αποτέλεσμα κατάτμησης	100
Εικόνα 8.30: Ιδανικό αποτέλεσμα	100
Εικόνα 8.31: Άγνωστο δείγμα ακτινογραφίας πνεύμονα	101
Εικόνα 8.32 Αποτέλεσμα κατάτμησης.....	102
Εικόνα 8.33: Ιδανικό Αποτέλεσμα.....	102
Εικόνα 8.34: Εισαγωγή αρχιτεκτονικής απλού CNN στο Matlab	103
Εικόνα 8.35: Accuracy και loss εκπαίδευσης απλού CNN με mini-batch size=1	104
Εικόνα 8.36: Αρχικά iterations εκπαίδευσης CNN με mini-batch size=1	105
Εικόνα 8.37: Accuracy και loss εκπαίδευσης απλού CNN με mini-batch size=3	106
Εικόνα 8.38: Αρχικά iterations εκπαίδευσης CNN με mini-batch size=3	107
Εικόνα 8.39: Γνωστό δείγμα ακτινογραφίας πνεύμονα.....	107
Εικόνα 8.40: Αποτέλεσμα κατάτμησης	108
Εικόνα 8.41: Ιδανικό Αποτέλεσμα.....	108
Εικόνα 8.42: Γνωστό δείγμα ακτινογραφίας πνεύμονα.....	109
Εικόνα 8.43:Αποτέλεσμα κατάτμησης	109
Εικόνα 8.44: Ιδανικό Αποτέλεσμα.....	109
Εικόνα 8.45: Άγνωστο δείγμα ακτινογραφίας πνεύμονα	110
Εικόνα 8.46: Αποτέλεσμα κατάτμησης	111
Εικόνα 8.47: Ιδανικό Αποτέλεσμα.....	111
Εικόνα 8.48: Άγνωστο δείγμα ακτινογραφίας πνεύμονα	112
Εικόνα 8.49: Αποτέλεσμα κατάτμησης	112
Εικόνα 8.50: Ιδανικό Αποτέλεσμα.....	112
Εικόνα 9.1: Πίνακας σφαλμάτων.....	114

Συντομογραφίες & Ακρωνύμια

ANN - Artificial Neural Networks

SLP – Single layer Perceptron

MLP - Multilayer Perceptron

ΤΝΔ - Τεχνητά Νευρωνικά Δίκτυα

DNN - Deep Neural Network

RBFNN - Radial Basis Function Neural Network

CNN - Convolutional Neural Networks

ΤΝΔ - Τεχνητά Νευρωνικά Δίκτυα

ReLU - Rectified Linear Unit

FC - Fully-connected

ΨΕΕ – Ψηφιακή Επεξεργασία Εικόνας

Εισαγωγή

Λόγω της ταχύτατης προόδου της επιστήμης και της τεχνολογίας, συχνά παρουσιάζονται πρωτοποριακές ανακαλύψεις, οι οποίες μας καθιστούν ικανούς να εκτελούμε άκοπα ένα ευρύ σύνολο δραστηριοτήτων. Η τεχνολογική αυτή πρόοδος αποτελεί καταλυτικό παράγοντα για την πρακτικότερη εφαρμογή της ιατρικής επιστήμης και πιο συγκεκριμένα την αμεσότερη διάγνωση μίας ιατρικής κατάστασης, την εφαρμογή κατάλληλων τεχνικών πρόληψης ή και για τη χορήγηση της ιδανικότερης θεραπείας.

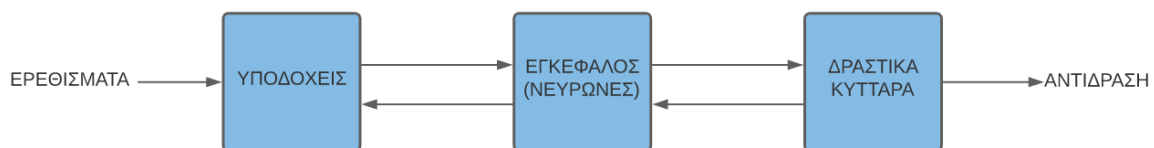
Ειδικότερα στον κλάδο της ιατρικής απεικόνισης, δηλαδή της μη επεμβατικής απεικόνισης του εσωτερικού του σώματος για την κλινική ανάλυση και την ιατρική παρέμβαση, απαιτούνται προηγμένες τεχνικές και μέθοδοι επεξεργασίας των προς ανάλυση βιοϊατρικών εικόνων. Συχνά είναι απαραίτητο, παραδείγματος χάρη, να εντοπιστούν συγκεκριμένοι οργανικοί ιστοί και στη συνέχεια να ανιχνευτούν οι όποιες ανωμαλίες παρουσιάζονται στους ιστούς αυτούς. Όπως είναι προφανές, μία τέτοια διαδικασία είναι χρονοβόρα και η χειροκίνητη εκτέλεσή της καθίσταται δυσμενής ή ακόμα και αδύνατη σε περιπτώσεις απαίτησης επεξεργασίας μεγάλης πληθώρας βιοϊατρικών εικόνων σε περιορισμένο χρόνο. Λύση στην ανίχνευση των συγκεκριμένων αυτών περιοχών του ανθρώπινου ιστού μέσα στις βιοϊατρικές εικόνες, δηλαδή την κατάτμηση (διαχωρισμό) των περιοχών ενδιαφέροντος μέσα σε μία εικόνα, έρχεται να προσφέρει η χρήση νευρωνικών δικτύων. Τα νευρωνικά δίκτυα αυτοματοποιούν την προαναφερθείσα διαδικασία, μιμούμενα τον ανθρώπινο εγκέφαλο, μέσω ποικίλων τεχνικών και αλγορίθμων. Την αυτοματοποιημένη κατάτμηση των βιοϊατρικών εικόνων πραγματεύεται η παρούσα διπλωματική εργασία.

Η εργασία που εκπονήθηκε εκτείνεται σε επτά κεφάλαια. Αρχικά, στο πρώτο κεφάλαιο γίνεται αναφορά στα τεχνητά νευρωνικά δίκτυα (ANN) και τις εφαρμογές τους σε πλήθος ζητημάτων που απασχολούν διάφορους τομείς, σε βασικά είδη ANN και τις αρχιτεκτονικές. Στο δεύτερο κεφάλαιο παρουσιάζονται τα βαθιά νευρωνικά δίκτυα (DNN) και αναλύονται εκτενώς τα συνελκτικά νευρωνικά δίκτυα (convolutional neural networks), τα οποία χρησιμοποιούνται και αργότερα στην πρακτική εφαρμογή που υλοποιήθηκε κατά την εκπόνηση της παρούσας εργασίας. Στο επόμενο κεφάλαιο παρατίθενται κάποιες βασικές έννοιες που συνδέονται με την ψηφιακή επεξεργασία της εικόνας. Ακολουθεί το τέταρτο κεφάλαιο, όπου παρουσιάζονται μερικοί από τους κυριότερους μετασχηματισμούς των ψηφιακών εικόνων, στις οποίες αυτοί μπορούν να εφαρμοστούν, με σκοπό την αποτελεσματικότερη κωδικοποίησή τους. Συνεχίζοντας, στο πέμπτο, έκτο και έβδομο κεφάλαιο παρατίθεται μία πληθώρα τεχνικών για την βελτίωση, κατά κύριο λόγο, του οπτικού αποτελέσματος της αναπαράστασης μίας ψηφιακής εικόνας, για την αποκατάσταση της ποιότητας της εικόνας αυτής από τον θόρυβο και για την κατάτμηση της αντίστοιχα. Ολοκληρώνουμε την εκπονηθείσα διπλωματική εργασία με την πρακτική εφαρμογή κάποιων εκ των προαναφερθέντων θεωρητικών γνώσεων. Συγκεκριμένα, πραγματοποιείται στο περιβάλλον αριθμητικής υπολογιστικής MATLAB, η υλοποίηση δύο διαφορετικών αρχιτεκτονικών CNN, για την κατάτμηση βιοϊατρικών εικόνων, καθώς επίσης εκτιμάται οπτικά και ποσοτικά η αποτελεσματικότητα του εκάστοτε δικτύου εξ αυτών, ενώ παρατίθενται και όποια συμπεράσματα απορρέουν από την ανωτέρω υλοποίηση. Τέλος σημειώνεται ότι τα τμήματα του κώδικα των οποίων έγινε χρήση παρατίθενται στο κλείσιμο της εργασίας το πεδίο των Παραρτημάτων.

1. Νευρωνικά Δίκτυα

1.1 Ο Βιολογικός Νευρώνας

Ο εγκέφαλος μας μπορεί να χαρακτηριστεί και ως ο πυρήνας του νευρικού μας συστήματος, που επανειλημμένα δέχεται διεγέρσεις από ποικίλους παράγοντες, τους οποίους αναλύει και συμβουλεύει το ανθρώπινο σώμα να πράξει αναλόγως. Μία πιο ενδελεχής εξέταση του νευρικού μας συστήματος θα το περιέγραφε ως τρεις βαθμίδες όπως απεικονίζεται στο ακόλουθο γράφημα.



Εικόνα 1.1: Βαθμίδες νευρικού συστήματος

Ξεκινώντας από αριστερά, συναντάμε τους υποδοχείς, οι οποίοι αποτελούν εξειδικευμένα κύτταρα που διεγείρονται από ποικίλους παράγοντες που παρουσιάζονται γύρω μας. Οι υποδοχείς πιο συγκεκριμένα, μετασχηματίζουν αυτές τις διεγέρσεις σε παλμικά σήματα, τα οποία λαμβάνονται από τους νευρώνες. Στη συνέχεια, αφού έχει επέλθει η ανάλυση των παλμικών αυτών σημάτων από τους νευρώνες, αποστέλλεται κάποιο παλμικό σήμα στα δραστικά κύτταρα που με τη σειρά τους δραστηριοποιούνται και ενεργούν σύμφωνα με το σήμα αυτό που έχουν λάβει. Η διαδικασία που περιγράψαμε και παρουσιάζεται στο παραπάνω διάγραμμα μπορεί να πραγματοποιηθεί αμφίδρομα.

Η ιδέα της ανάπτυξης του τεχνητού νευρώνα ήταν απόρροια του τρόπου κατά τον οποίο δρούσε ο βιολογικός νευρώνας. Στο σύνολό τους συναντώνται κατά προσέγγιση από ένα έως δέκα δισεκατομμύρια νευρώνες στον άνθρωπο, εκ των οποίων ο εκάστοτε νευρώνας έχει άμεση επικοινωνία με δέκα χιλιάδες γειτονικούς νευρώνες. Υποστηρίζεται ότι το σύνολο τους φτάνει στον αριθμό που προαναφέραμε σε διάστημα κάποιων ημερών, αφού το έμβρυο έχει γεννηθεί. Επίσης, παρά το γεγονός ότι οι νευρώνες του ανθρώπινου νευρικού συστήματος συνεχώς μειώνονται, δεν υπάρχει αντίκτυπο στον τρόπο με τον οποίο επεξεργάζονται οι διεγέρσεις, αφού ο καταλυτικός παράγοντας για τη διαδικασία αυτή είναι ο διάλυος επικοινωνίας μεταξύ των εκάστοτε γειτονικών νευρώνων. Οι διάλυοι αυτοί ονομάζονται συνάψεις, και είναι απαραίτητες για την επικοινωνία μεταξύ των νευρώνων. Τα μέσα επικοινωνίας αυτά μεταβάλλονται ή αναγεννιούνται σύμφωνα με τα ερεθίσματα που λαμβάνουμε όσο ζούμε.

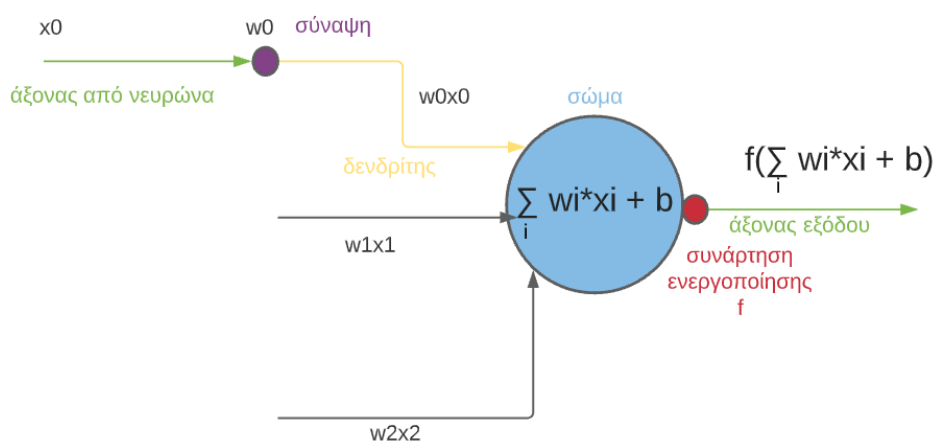
1.2 Τεχνητά Νευρωνικά δίκτυα

Τα τεχνητά νευρωνικά δίκτυα (ANN) αποτελούν υπολογιστικά συστήματα τα οποία έχουν ως πηγή έμπνευσης τον εγκέφαλο (πυρήνα) των ζωντανών οργανισμών. Το τεχνητό νευρωνικό δίκτυο στηρίζεται σε ένα πλήθος αλληλένδετων τεχνητών νευρώνων, οι οποίοι αναπαριστούν σε γενικές γραμμές τα νευρικά κύτταρα του εγκεφάλου των ζωντανών οργανισμών. Κάθε δεσμός μεταξύ των κόμβων, αντίστοιχος της σύναψης στον ανθρώπινο εγκέφαλο, έχει τη δυνατότητα να μεταβιβάσει ένα ερέθισμα σε διαφορετικούς κόμβους. Εφόσον κάποιος κόμβος

έχει λάβει το παλμικό ερέθισμα, είναι σε θέση να αφυπνίσει τους γειτονικούς κόμβους με τους οποίους έχει άμεση επικοινωνία. Το «ερέθισμα» σε ένα δεσμό είναι ουσιαστικά μία αριθμητική τιμή και το αποτέλεσμα που εξέρχεται από τον εκάστοτε κόμβο εκτιμάται με τη βοήθεια μίας μη γραμμικής συνάρτησης. Οι κόμβοι και οι δεσμοί τους διαθέτουν ένα φορτίο το οποίο συμμορφώνεται με την πορεία της εκπαίδευσης. Το φορτίο αυτό καθορίζει το πόσο ισχυρό θα είναι το ερέθισμα μεταξύ ενός δεσμού.

1.2.1 Ο τεχνητός Νευρώνας

Παρακάτω (Εικόνα 1.2) όπου παρουσιάζεται ένας τεχνητός νευρώνας ή κόμβος, παρατηρούμε τα σήματα τα οποία μεταδίδονται κατά την κατεύθυνση των αξόνων όπως για παράδειγμα το σήμα x_0 το οποίο πολλαπλασιάζεται με τους δενδρίτες κάποιου διαφορετικού κόμβου σύμφωνα με το πόσο δυνατή είναι η σύνδεσή τους (σύναψη), σύμφωνα λοιπόν με την τιμή του βάρους w_0 . Ανάλογα με το πόσο ισχυρές είναι αυτές οι συνδέσεις, άρα ανάλογα με τις τιμές των βαρών, καθορίζεται το ποσοστό κατά το οποίο ένας κόμβος δρα σε κάποιον γειτονικό του κόμβο. Το κάθε πολλαπλασιασμένο σήμα προστίθεται με τα υπόλοιπα και εφόσον η τιμή του τελικού αθροίσματος εμφανίζεται μεγαλύτερη από ένα συγκεκριμένο threshold, σε αυτήν την περίπτωση ο κόμβος ενεργοποιείται και δημιουργεί ένα παλμικό σήμα το οποίο διαδίδεται μέσω του άξονα σε διαφορετικούς κόμβους.



Εικόνα 1.2: Δομή τεχνητού νευρώνα

1.2.2 Δομή των Τεχνητών Νευρωνικών Δικτύων

Τα νευρωνικά δίκτυα είναι πολύπλοκες δομές κατασκευασμένες από τεχνητούς νευρώνες που μπορούν να λάβουν πολλαπλές εισόδους για να παράγουν μια ενιαία έξοδο. Αυτή είναι η πρωταρχική δουλειά ενός Νευρωνικού Δικτύου, να μετατρέπει την είσοδο σε μια ουσιαστική έξοδο. Συνήθως, κάθε ΝΔ απαρτίζουν ένα input layer και ένα output layer με ένα ή πολλαπλά hidden layers.

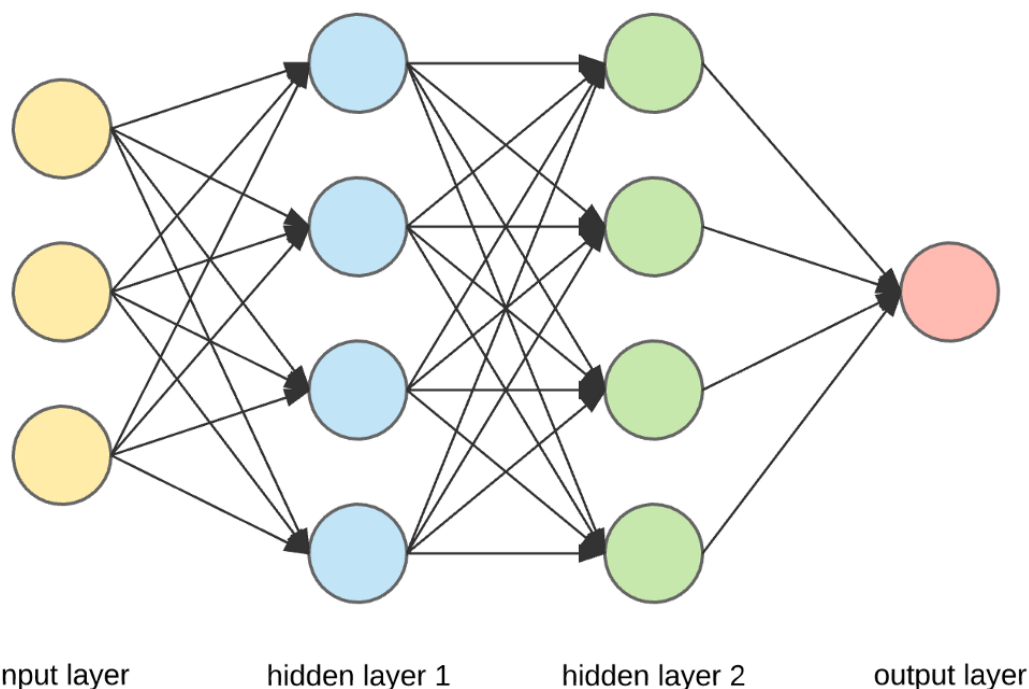
Σε κάθε Νευρωνικό Δίκτυο, όλοι οι νευρώνες επηρεάζουν ο ένας τον άλλον, και ως εκ τούτου, είναι όλοι συνδεδεμένοι. Το δίκτυο μπορεί να αναγνωρίσει και να παρατηρήσει κάθε πτυχή του συνόλου των πληροφοριών και το πώς τα διαφορετικά μέρη των δεδομένων μπορεί ή όχι να σχετίζονται μεταξύ τους. Έτσι τα νευρωνικά δίκτυα είναι ικανά να βρίσκουν εξαιρετικά πολύπλοκα μοτίβα σε τεράστιους όγκους δεδομένων.

Σε ένα νευρωνικό δίκτυο, η ροή των πληροφοριών γίνεται με δύο τρόπους:

- **Δίκτυα ανατροφοδότησης (Feedforward Networks)** : Σε αυτό το μοντέλο, τα σήματα ταξιδεύουν μόνο προς μία κατεύθυνση, προς το επίπεδο εξόδου. Τα Feedforward Networks διαθέτουν ένα input layer και ένα output layer με μηδενικά ή πολλαπλά hidden layers. Χρησιμοποιούνται ευρέως στην αναγνώριση προτύπων.
- **Δίκτυα ανάδρασης (Feedback Networks)** : Σε αυτό το μοντέλο, τα επαναλαμβανόμενα ή διακρατικά δίκτυα «θυμούνται» την εσωτερική τους κατάσταση για να διεργαστούν την ακολουθία των εισερχόμενων δεδομένων. Σε αυτά, τα σήματα μπορούν να ταξιδεύουν και προς τις δύο κατευθύνσεις μέσω των βρόχων (κρυφό στρώμα/ κρυφά στρώματα) στο δίκτυο. Συνήθως χρησιμοποιούνται σε χρονοσειρές και διαδοχικές εργασίες.

Τα στοιχεία που αποτελούν ένα τεχνητό νευρωνικό δίκτυο , είναι :

- Τα επίπεδα εισόδου (input layers)
- Τα κρυφά επίπεδα (hidden layers)
- Τα επίπεδα εξόδου (output layers)
- Τα βάρη (weights)
- Οι κόμβοι-νευρώνες (neurons-nodes)



Εικόνα 1.3: Δομή Τεχνητού Νευρωνικού δικτύου

(Πηγή : https://miro.medium.com/max/1000/1*3fA77_mLNiJTSgZFhYnU0Q.png)

Στην παραπάνω εικόνα, το εξωτερικό κίτρινο στρώμα είναι το στρώμα εισόδου. Ένας νευρώνας είναι η βασική μονάδα ενός νευρωνικού δικτύου. Οι νευρώνες ή κόμβοι λαμβάνουν είσοδο από μια εξωτερική πηγή ή άλλους κόμβους. Κάθε κόμβος συνδέεται με έναν άλλο

κόμβο από το επόμενο επίπεδο και κάθε τέτοια σύνδεση έχει ένα συγκεκριμένο βάρος. Τα βάρη αποδίδονται σε έναν νευρώνα με βάση τη σχετική σημασία του έναντι άλλων εισόδων.

Όταν όλες οι τιμές του κόμβου από το κίτρινο στρώμα πολλαπλασιαστούν (μαζί με το βάρος τους) και συνοψιστούν, δημιουργείται μια τιμή για το πρώτο κρυφό στρώμα. Με βάση τη συνοπτική τιμή, το μπλε στρώμα έχει μια προκαθορισμένη συνάρτηση «ενεργοποίησης» που καθορίζει εάν αυτός ο κόμβος θα «ενεργοποιηθεί» ή όχι και πόσο «ενεργός» θα είναι. Αν θεωρήσουμε του κόμβους ως συστατικά της δημιουργίας του δικτύου, αφού αποτελούν τα σημεία εκκίνησης της διαδικασίας, την ποσότητα κάθε συστατικού αντιπροσωπεύει το «βάρος».

Το στρώμα ή τα στρώματα που κρύβονται μεταξύ του επιπέδου εισόδου και εξόδου είναι γνωστά ως κρυφά στρώματα. Ονομάζονται με αυτόν τον τρόπο καθώς είναι πάντα κρυμμένα από τον εξωτερικό κόσμο. Ο κύριος υπολογισμός ενός νευρωνικού δικτύου πραγματοποιείται στα κρυφά στρώματα. Έτσι, το κρυφό επίπεδο παίρνει όλες τις εισόδους από το επίπεδο εισόδου και εκτελεί τον απαραίτητο υπολογισμό για να δημιουργήσει ένα αποτέλεσμα. Αυτό το αποτέλεσμα στη συνέχεια προωθείται στο επίπεδο εξόδου και παρουσιάζεται στον χρήστη.

Προκύπτει όπως είναι λογικό, ότι η αποδοτικότητα του τεχνητού νευρωνικού μας δικτύου εξαρτάται άμεσα από:

- Το σύνολο των hidden layers
- Το σύνολο των nodes ανά layer
- Τις τιμές των βαρών κατά τις συνδέσεις των κόμβων

Προφανώς, όταν γίνεται χρήση περισσότερων hidden layers και nodes ανά layer, η επεξεργαστική ισχύς του τεχνητού νευρωνικού δικτύου αυξάνεται.

1.2.3 Διαφορές Βιολογικών και Τεχνητών Νευρωνικών Δικτύων

Αν και οι τεχνητοί νευρώνες είχαν ως πηγή έμπνευσης τις βιολογικές διεργασίες που οι επιστήμονες μπόρεσαν να παρατηρήσουν στον εγκέφαλο, διαφέρουν από τους βιολογικούς ομολόγους τους με πολλούς τρόπους. Παρατηρούμε λοιπόν διαφορές που αφορούν:

- Το μέγεθος:

Ο εγκέφαλός μας περιέχει περίπου 86 δισεκατομμύρια νευρώνες και περισσότερες από 100 τρισεκατομμύρια (ή σύμφωνα με ορισμένες εκτιμήσεις 1000 τρισεκατομμύρια) συνάψεις (συνδέσεις). Ο αριθμός των «νευρώνων» στα τεχνητά δίκτυα είναι πολύ μικρότερος από αυτόν (συνήθως στο χώρο 10-1000), αλλά η σύγκριση των αριθμών τους με αυτόν τον τρόπο είναι παραπλανητική. Τα perceptron απλώς λαμβάνουν εισόδους στους «δενδρίτες» τους και παράγουν έξοδο στους «κλάδους των νευραξόνων» τους. Ένα δίκτυο perceptron ενός μόνο στρώματος αποτελείται από πολλά perceptron που δεν είναι διασυνδεδεμένα: όλα εκτελούν ακριβώς την ίδια εργασία ταυτόχρονα. Τα βαθιά νευρωνικά δίκτυα αποτελούνται συνήθως από νευρώνες εισόδου, νευρώνες εξόδου και νευρώνες στα κρυφά επίπεδα, μεταξύ. Όλα τα στρώματα είναι συνήθως (αλλά όχι απαραίτητα) πλήρως συνδεδεμένα με το επόμενο στρώμα, πράγμα που σημαίνει ότι οι τεχνητοί νευρώνες έχουν συνήθως τόσες συνδέσεις όσες και οι τεχνητοί νευρώνες στο προηγούμενο και το επόμενο στρώμα μαζί. Τα συνελκτικά νευρωνικά

δίκτυα χρησιμοποιούν επίσης διαφορετικές τεχνικές για να εξάγουν χαρακτηριστικά από τα δεδομένα που είναι πιο εξελιγμένα από αυτά που μπορούν να κάνουν μόνοι μερικοί διασυνδεδεμένοι νευρώνες. Η χειροκίνητη εξαγωγή χαρακτηριστικών απαιτεί ισχύ του ανθρώπινου εγκεφάλου, η οποία επίσης δεν λαμβάνεται υπόψη κατά την άθροιση του αριθμού των «νευρώνων» που απαιτούνται για τις εργασίες Deep Learning. Ο περιορισμός στο μέγεθος δεν είναι μόνο υπολογιστικός: η απλή αύξηση του αριθμού των στρωμάτων και των τεχνητών νευρώνων δεν αποφέρει πάντα καλύτερα αποτελέσματα στις εργασίες μηχανικής μάθησης.

- Την τοπολογία:

Όλα τα επίπεδα των τεχνητών νευρωνικών δικτύων υπολογίζονται ένα προς ένα, αντί να αποτελούν μέρος ενός δικτύου που έχει κόμβους που υπολογίζουν ασύγχρονα. Τα δίκτυα τροφοδοσίας υπολογίζουν την κατάσταση ενός στρώματος τεχνητών νευρώνων και τα βάρη τους και έπειτα γίνεται αξιοποίηση των αποτελεσμάτων για τον υπολογισμό του επόμενου επιπέδου. Στην αντίστροφη διάδοση, ο αλγόριθμος υπολογίζει κάποια αλλαγή στα βάρη με τον αντίθετο τρόπο, για να μειώσει τη διαφορά των υπολογιστικών αποτελεσμάτων τροφοδοσίας στο επίπεδο εξόδου από τις αναμενόμενες τιμές του επιπέδου εξόδου. Στα βιολογικά δίκτυα, οι νευρώνες μπορούν να ενεργοποιούνται ασύγχρονα.

- Την ταχύτητα:

Ορισμένοι βιολογικοί νευρώνες μπορούν να πυροδοτήσουν περίπου 200 φορές το δευτερόλεπτο κατά μέσο όρο. Τα σήματα ταξιδεύουν με διαφορετικές ταχύτητες ανάλογα με τον τύπο της νευρικής ώθησης και κυμαίνονται από 0,61 m/s έως 119 m/s. Οι ταχύτητες διάδοσης του σήματος ποικίλλουν επίσης για κάθε οργανισμό ανάλογα το ύψος, τη θερμοκρασία, την ιατρική κατάσταση, την έλλειψη ύπνου κ.λπ. Αντίθετα, οι τεχνητοί νευρώνες δεν αντιμετωπίζουν «κόπωση», είναι συναρτήσεις που μπορούν να υπολογιστούν τόσες φορές και όσο γρήγορα θα επέτρεπε η αρχιτεκτονική του υπολογιστή.

- Την ανοχή σε σφάλματα:

Τα δίκτυα βιολογικών νευρώνων λόγω της τοπολογίας τους είναι επίσης ανεκτικά σε σφάλματα. Οι πληροφορίες αποθηκεύονται περιττά, έτσι ώστε οι μικρές αποτυχίες να μην οδηγούν σε απώλεια μνήμης. Ο εγκέφαλος μπορεί επίσης να ανακάμψει και να θεραπευτεί σε ένα βαθμό. Τα τεχνητά νευρωνικά δίκτυα δεν διαμορφώνονται για ανοχή σε σφάλματα ή αυτοαναγέννηση αν και η ανάκτηση είναι δυνατή με την αποθήκευση της τρέχουσας κατάστασης (τιμές βάρους) του μοντέλου και τη συνέχιση της εκπαίδευσης από αυτήν την κατάσταση. Οι εγκαταλείψεις μπορούν να ενεργοποιήσουν και να απενεργοποιήσουν τυχαίους νευρώνες σε ένα στρώμα κατά τη διάρκεια της προπόνησης, μιμούμενοι μη διαθέσιμες διαδρομές για σήματα και αναγκάζοντας κάποιο πλεονασμό. Ακόμα, η εκπαίδευση τεχνητών νευρωνικών δικτύων για μεγαλύτερες χρονικές περιόδους δεν θα επηρεάσει την αποτελεσματικότητα των τεχνητών νευρώνων. Ωστόσο, το υλικό που χρησιμοποιείται για προπόνηση μπορεί να φθαρεί πολύ γρήγορα εάν χρησιμοποιείται τακτικά και θα πρέπει να αντικατασταθεί. Μια άλλη διαφορά είναι ότι όλες οι διεργασίες (καταστάσεις και τιμές) μπορούν να παρακολουθούνται στενά μέσα σε ένα τεχνητό νευρωνικό δίκτυο.

- Την κατανάλωση ισχύος:

Ο εγκέφαλος καταναλώνει περίπου το 20% όλης της ενέργειας του ανθρώπινου σώματος. Ένας ενήλικος εγκέφαλος λειτουργεί με περίπου 20 watt (μόλις αρκετά για να ανάψει αμυδρά μια λάμπα) όντας εξαιρετικά αποδοτικός. Τα μηχανήματά μας είναι πολύ λιγότερο αποδοτικά από τα βιολογικά συστήματα. Οι υπολογιστές παράγουν επίσης πολλή θερμότητα όταν χρησιμοποιούνται.

- Τη μάθηση:

Ακόμα δεν απόλυτα προφανές πώς μαθαίνουν οι εγκέφαλοι ή πώς οι περιττές συνδέσεις αποθηκεύουν και ανακαλούν πληροφορίες. Οι ίνες του εγκεφάλου αναπτύσσονται και απλώνονται για να συνδεθούν με άλλους νευρώνες, η νευρο-πλαστικότητα επιτρέπει τη δημιουργία νέων συνδέσεων ή περιοχών για κίνηση και αλλαγή λειτουργίας και οι συνάψεις μπορεί να ενισχυθούν ή να εξασθενήσουν ανάλογα με τη σημασία τους. Με τη μάθηση, χτίζουμε σε πληροφορίες που είναι ήδη αποθηκευμένες στον εγκέφαλο. Οι γνώσεις μας βαθαίνουν με την επανάληψη και κατά τη διάρκεια του ύπνου και οι εργασίες που κάποτε απαιτούσαν ειδική εξάσκηση μπορούν να εκτελεστούν αυτόματα μόλις κατακτηθούν. Τα τεχνητά νευρωνικά δίκτυα από την άλλη πλευρά, έχουν ένα προκαθορισμένο μοντέλο, όπου δεν μπορούν να προστεθούν ή να αφαιρεθούν άλλοι νευρώνες ή συνδέσεις. Μόνο τα βάρη των συνδέσεων μπορούν να αλλάξουν κατά τη διάρκεια της προπόνησης. Τα δίκτυα ξεκινούν με τυχαίες τιμές βάρους και θα προσπαθήσουν να φτάσουν σε ένα σημείο όπου περαιτέρω αλλαγές στα βάρη δεν θα βελτιώνουν πλέον την απόδοση του δικτύου. Ακριβώς όπως υπάρχουν πολλές λύσεις για τα ίδια προβλήματα στην πραγματική ζωή, δεν υπάρχει καμία εγγύηση ότι τα βάρη του δικτύου θα είναι η καλύτερη δυνατή διάταξη βαρών σε ένα πρόβλημα, ενώ θα αντιπροσωπεύουν μόνο μία από τις άπειρες προσεγγίσεις για τη λύση του προβλήματος. Η μάθηση μπορεί να περιγραφεί ως η διαδικασία εύρεσης βέλτιστων βαρών για την ελαχιστοποίηση των διαφορών μεταξύ της αναμενόμενης και της παραγόμενης εξόδου του δικτύου.

1.3 Εφαρμογές και προβλήματα προς επίλυση με ΤΝΔ.

1.3.1 Εφαρμογές Τεχνητών Νευρωνικών Δικτύων

Μερικοί από τους τομείς , όπου έχει γίνει χρήση τεχνητών νευρωνικών δικτύων με πολύ ικανοποιητικά αποτελέσματα , είναι οι ακόλουθοι:

- Ιατρική:

Στον τομέα της ιατρικής βρήκαν εφαρμογή τα ΤΝΔ διευκολύνοντας την καταγραφή για παράδειγμα των καρδιακών παλμών , των ποσοστών της περιεκτικότητας διαφόρων στοιχείων στο αίμα και της αναπνευστικής συχνότητας. Καταγράφοντας τις αλλαγές των τιμών των παραπάνω, τα ΤΝΔ θέτουν του ιατρούς ικανούς να εφαρμόζουν κατάλληλες τεχνικές πρόληψης ασθενειών ή και να εφαρμόζουν κατάλληλη θεραπεία για μία ιατρική κατάσταση.

- Χρηματιστήριο:

Τα ΤΝΔ συχνά βρίσκουν εφαρμογή στην πολυσυζητημένη αγορά του χρηματιστήριου , όπου γίνεται μέσω αυτών παρακολούθηση των εναλλαγών των τιμών των μετοχών. Με τον τρόπο αυτό αναλύονται πολλοί παράγοντες οι οποίοι καταγράφονται, με σκοπό την ακριβέστερη πρόβλεψη της μελλοντικής αξίας της εκάστοτε μετοχής.

- **Μετεωρολογία:**

Μέσω του τρόπου λειτουργίας τους τα ΤΝΔ χρησιμοποιούνται ευρέως για την πρόβλεψη των καιρικών συνθηκών, αφού γίνεται σε πρώτη φάση καταγραφή των παλαιότερων τιμών διαφόρων καιρικών παραγόντων (π.χ. θερμοκρασίας).

- **Αυτόνομη οδήγηση:**

Τα βαθιά νευρωνικά δίκτυα είναι ένα από τα κύρια συστατικά της τεχνολογίας της αυτόνομης οδήγησης. Τα νευρωνικά δίκτυα αναλύουν τα οπτικά δεδομένα που λαμβάνονται από τις κάμερες του αυτοκινήτου και έτσι προχωράνε στην αποφυγή εμποδίων, στην αυτόματη αλλαγή κατεύθυνσης και γενικότερα στην προσπάθεια αποφυγής ατυχημάτων.

1.3.2 Προβλήματα προς επίλυση με ΤΝΔ

Τα τεχνητά νευρωνικά δίκτυα συχνά προσφέρουν λύσεις σε ποικίλα συγκεκριμένα προβλήματα, τα οποία είναι πιο εξειδικευμένα και κάποια από αυτά είναι τα ακόλουθα:

- **Κατηγοριοποίηση (classification):**

Τα τεχνητά νευρωνικά δίκτυα βρίσκουν εφαρμογή ως εργαλεία με στόχο την κατηγοριοποίηση δεδομένων, τα οποία προσκομίζονται και χωρίζονται σε δύο ή περισσότερες κλάσεις. Ο διαχωρισμός αυτός λαμβάνει χώρα με την προϋπόθεση ότι ο χρήστης του δικτύου θα έχει γνωστοποιήσει στο δίκτυο κάποιο υπόδειγμα κατηγοριοποίησης, βάσει του οποίου θα διαχωριστούν τα δεδομένα μας στις προκαθορισμένες κλάσεις.

- **Ομαδοποίηση (clustering):**

Όσον αφορά τα προβλήματα ομαδοποίησης, γίνεται και εδώ διαχωρισμός ενός συνόλου δεδομένων, αλλά αντίθετα με την προηγούμενη κατηγορία προβλήματος, εδώ ο χρήστης δεν προσφέρει κάποιο υπόδειγμα το οποίο πρέπει να ακολουθηθεί, αλλά το ΤΝΔ προχωράει στο διαχωρισμό λαμβάνοντας υπόψη τα κοινά στοιχεία των δεδομένων μας.

- **Πρόβλεψη (forecasting):**

Τα ΤΝΔ πολύ συχνά βρίσκουν λύση στην ανάγκη πρόβλεψης της μελλοντικής έκβασης κάποιων καταστάσεων, βασιζόμενα στην καταγραφή κάποιων αρχικών δεδομένων. Παράδειγμα αυτής της χρήσης αποτελεί και η πρόγνωση της θερμοκρασίας.

- **Έλεγχος (control):**

Ένα ΤΝΔ συχνά χρησιμοποιείται για την εποπτεία της προόδου μίας διεργασίας. Έτσι το δίκτυο μπορεί να παρεμβάλλεται ή να ενημερώνει το χρήστη σε περίπτωση που η πορεία της διεργασίας αυτής παρεκκλίνει από τις καθορισμένες προδιαγραφές.

1.4 Μάθηση Τεχνητών Νευρωνικών Δικτύων

Ένα χαρακτηριστικό γνώρισμα των Νευρωνικών Δικτύων είναι η δυνατότητα να μαθαίνουν. Η μάθηση αυτή ουσιαστικά πραγματοποιείται με τη βοήθεια της κυκλοφορίας τιμών και βαρών, με στόχο την σταδιακή προσκόμιση της πληροφορίας, που αργότερα μπορεί να χρησιμοποιηθεί όποτε κριθεί απαραίτητο. Ενώ συναντιούνται ποικίλες τεχνικές για τον εντοπισμό των καταλληλότερων τιμών και βαρών των ΤΝΔ, σε γενικές γραμμές

αντιστοιχίζουμε όλες τις τεχνικές μάθησης στην επιβλεπόμενη μάθηση και την μη επιβλεπόμενη μάθηση.

Κατά την επιβλεπόμενη μάθηση :

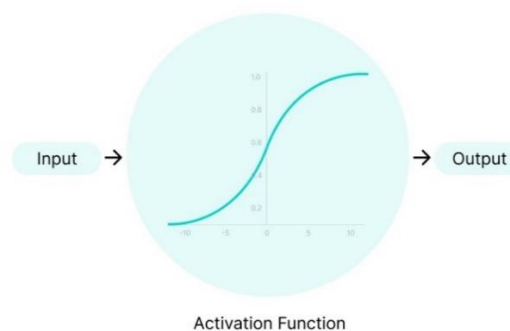
- Τα δεδομένα εκπαίδευσης περιλαμβάνουν τόσο την είσοδο όσο και το επιθυμητά αποτελέσματα.
- Σε πολλά παραδείγματα , τα σωστά αποτελέσματα (στόχοι) είναι γνωστά και προσφέρονται ως είσοδος στο μοντέλο κατά τη διαδικασία της μάθησης.
- Η διάθεση ενός κατάλληλου dataset εκπαίδευσης, επικύρωσης και ελέγχου είναι καταλυτικός παράγοντας της σωστής λειτουργίας του δικτύου.
- Η διαδικασία είναι συνήθως γρήγορη και ακριβής.
- Το ΤΝΔ δίνει τα σωστά αποτελέσματα όταν δίνονται νέα δεδομένα στην είσοδο χωρίς να είναι γνωστός εκ των προτέρων ο στόχος.

Κατά τη μη επιβλεπόμενη μάθηση :

- Τα σωστά αποτελέσματα (στόχοι) δεν είναι γνωστά και δεν προσφέρονται ως είσοδος στο μοντέλο κατά τη διαδικασία της μάθησης, σε αντίθεση με την επιβλεπόμενη μάθηση.
- Μπορεί να χρησιμοποιηθεί για την ομαδοποίηση των δεδομένων εισόδου σε κλάσεις μόνο με βάση τις στατιστικές τους ιδιότητες.
- Η εφαρμογή ετικετών (labels) μπορεί να πραγματοποιηθεί ακόμη και αν οι ετικέτες είναι διαθέσιμες μόνο για μικρό αριθμό αντικειμένων αντιπροσωπευτικών των επιθυμητών κλάσεων.

1.5 Συναρτήσεις Ενεργοποίησης (Activation Functions)

Η συνάρτηση ενεργοποίηση αποτελεί καταλυτικό παράγοντα της λειτουργίας ενός τεχνητού νευρωνικού δικτύου. Μια activation function καθορίζει αν θα γίνει η αφύπνιση ενός κόμβου. Καθορίζει δηλαδή, ποιοι κόμβοι θα ληφθούν υπόψιν για τους υπολογισμούς των εξόδων.



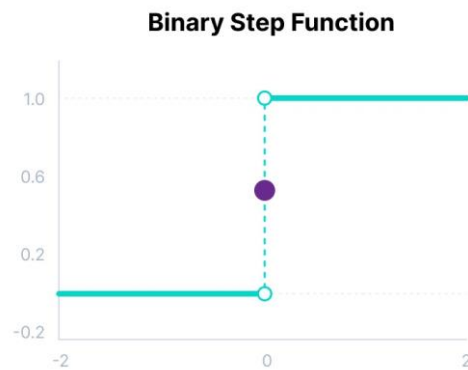
Εικόνα 1.4: Συνάρτηση ενεργοποίησης

(Πηγή : <https://www.v7labs.com/blog/neural-networks-activation-functions>)

Σύμφωνα με το είδος του ζητήματος που θέλουμε να διεκπεραιώσουμε , κάνουμε χρήση της κατάλληλης συνάρτησης ενεργοποίησης . Κάποια από τα πιο διαδεδομένα είδη συναρτήσεων ενεργοποίησης αναφέρονται ακολούθως.

1.5.1 Συνάρτηση δυαδικού βήματος

Η συνάρτηση δυαδικού βήματος εξαρτάται από μια τιμή κατωφλίου που αποφασίζει αν θα γίνει η αφύπνιση ενός κόμβου. Η τιμή εισόδου που λαμβάνεται από την συνάρτηση αφυπνίζει τον εκάστοτε κόμβο, μόνο εφόσον η τιμή αυτή ξεπερνάει ένα συγκεκριμένο όριο, πράγμα που σημαίνει ότι η έξοδος του δεν μεταφέρεται στο επόμενο κρυφό στρώμα.



Εικόνα 1.5: Συνάρτηση ενεργοποίησης δυαδικού βήματος

(Πηγή : <https://www.v7labs.com/blog/neural-networks-activation-functions>)

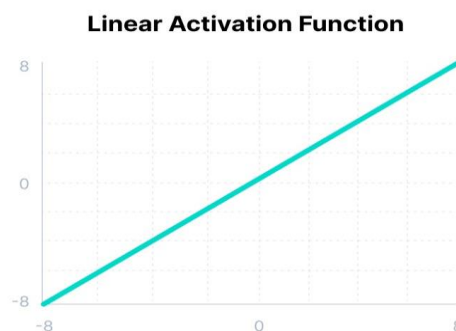
Για την συνάρτηση δυαδικού βήματος ισχύει η σχέση :

$$f(x) = \begin{cases} 0 & \text{για } x < 0 \\ 1 & \text{για μη αρνητικό } x \end{cases} \quad (1.1)$$

Πρέπει να τονιστεί ότι η συνάρτηση αυτή δεν μπορεί να παρέχει εξόδους πολλαπλών τιμών, για παράδειγμα, δεν μπορεί να χρησιμοποιηθεί για προβλήματα ταξινόμησης πολλών κλάσεων.

1.5.2 Γραμμική συνάρτηση

Η συνάρτηση γραμμικής ενεργοποίησης είναι επίσης γνωστή ως Λειτουργία ταυτότητας όπου η ενεργοποίηση είναι ανάλογη της εισόδου.



Εικόνα 1.6: Γραμμική συνάρτηση ενεργοποίησης

(Πηγή : <https://www.v7labs.com/blog/neural-networks-activation-functions>)

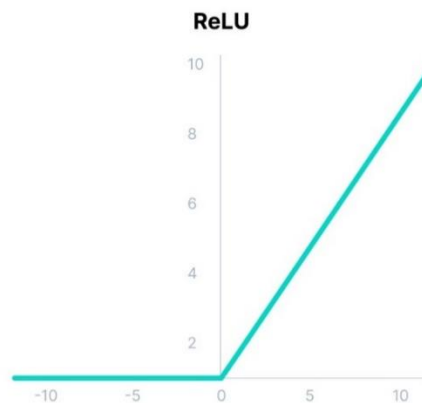
Για την συνάρτηση δυαδικού βήματος ισχύει η σχέση :

$$f(x) = x \quad (1.2)$$

Όλα τα επίπεδα του νευρωνικού δικτύου θα καταρρεύσουν σε ένα εάν χρησιμοποιηθεί μια γραμμική συνάρτηση ενεργοποίησης. Ανεξάρτητα από τον αριθμό των επιπέδων στο νευρωνικό δίκτυο, το τελευταίο στρώμα θα εξακολουθεί να είναι μια γραμμική συνάρτηση του πρώτου στρώματος.

1.5.3 Συνάρτηση ανόρθωσης (ReLU)

Παρά τις όποιες ομοιότητες με την περίπτωση της γραμμικής συνάρτησης, η ReLU (Rectified Linear Unit) έχει μια παράγωγο συνάρτηση και επιτρέπει την αντίστροφη διάδοση. Οι κόμβοι εδώ θα απενεργοποιηθούν μόνο εάν σημειωθεί τιμή εξόδου κατώτερη του μηδέν.

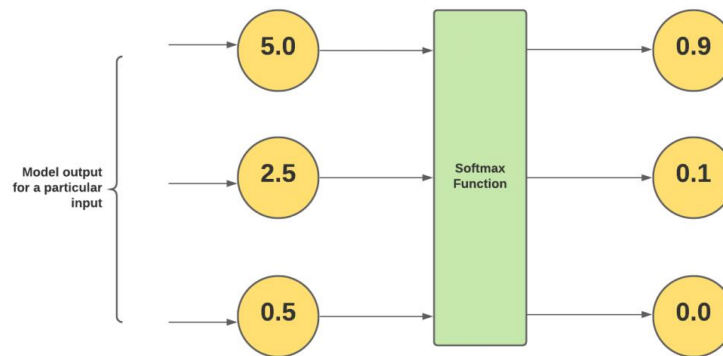


Εικόνα 1.7: Συνάρτηση ανόρθωσης

(Πηγή : <https://www.v7labs.com/blog/neural-networks-activation-functions>)

1.5.4 Συνάρτηση SoftMax

Η SoftMax function συναντάται στο τελευταίο στρώμα ενός ΤΝΔ σε περιπτώσεις κυρίως ζητημάτων πολυταξικής ταξινόμησης. Ουσιαστικά η συνάρτηση αυτή μετατρέπει όλες τις τιμές που προκύπτουν από τους νευρώνες του output layer σε τιμές από 0 έως 1 ,ενώ εάν οι τιμές αυτές προστεθούν προκύπτει η μονάδα , προκύπτουν δηλαδή τιμές πιθανοτήτων , οι οποίες αντιστοιχίζονται στο εκάστοτε class που εκπροσωπείται από το output layer.



Εικόνα 1.8: Συνάρτηση SoftMax

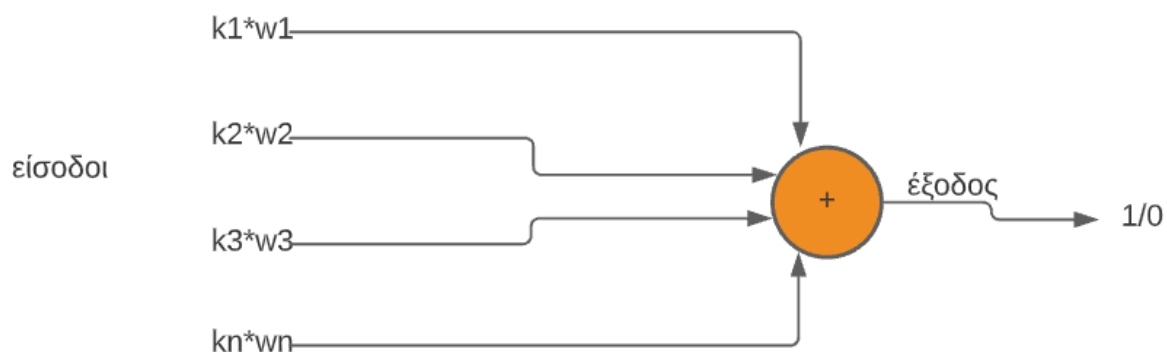
(Πηγή : <https://vitalflux.com/what-softmax-function-why-needed-machine-learning>)

1.6 Κυριότερες Αρχιτεκτονικές ΤΝΔ

1.6.1 Single Layer Perceptron

Το perceptron ακολουθεί αλγόριθμους επιβλεπόμενης μάθησης με σκοπό τη δυαδική κατηγοριοποίηση. Αποτελεί τον πιο λιτό τύπο νευρωνικού δικτύου που μπορούμε να συναντήσουμε. Στην ουσία το perceptron κάνει χρήση μιας συνάρτησης με την οποία μπορεί να κρίνει αν μια είσοδος αντιστοιχείται σε μία εκ των δύο διαθέσιμων κλάσεων. Το perceptron αποτελείται μόνο από το στρώμα εισόδου, όπου προσφέρονται οι πληροφορίες προς υπολογισμό, ο οποίος λαμβάνει χώρα στο μοναδικό άλλο στρώμα, αυτό της εξόδου, το οποίο δίνει την πιο προσιτή τιμή εξόδου μετά από τον κατάλληλο υπολογισμό.

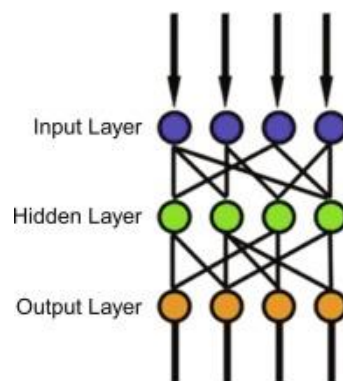
Πρακτικά το perceptron, όπως φαίνεται και στην παρακάτω εικόνα, λαμβάνει κάποιες τιμές εισόδου ($k_1 \dots k_n$), η εκάστοτε εκ των οποίων έχει διαφορετική βαρύτητα για το σύστημα, η οποία αντιπροσωπεύεται από ένα βάρος ($w_1 \dots w_n$). Αφού πολλαπλασιαστούν τα βάρη με τις τιμές εισόδου, προστίθενται τα αποτελέσματα των πολλαπλασιασμών αυτών και ανάλογα εάν το άθροισμα είναι μεγαλύτερο ή μικρότερο από ένα κατώφλι, τότε παίρνουμε τελικά την τιμή 1 ή 0 από την έξοδο.



Εικόνα 1.9: Single Layer Perceptron

1.6.2 Multilayer Perceptrons (MLP)

Το Multilayer Perceptron (MLP) είναι ένα δίκτυο εμπρόσθιας τροφοδότησης. Αποτελείται από τρία είδη layers, το input layer, το output layer και το hidden layer, τα οποία παρουσιάζονται παρακάτω. Το στρώμα εισόδου λαμβάνει το σήμα εισόδου που πρόκειται να επεξεργαστεί. Η επιθυμητή διεργασία (task), όπως η πρόβλεψη ή η κατηγοριοποίηση, λαμβάνουν χώρα στο στρώμα εξόδου. Ένας αυθαίρετος αριθμός κρυφών επιπέδων που τοποθετούνται μεταξύ του επιπέδου εισόδου και εξόδου είναι η πραγματική υπολογιστική μηχανή του MLP. Παρόμοια με ένα δίκτυο εμπρόσθιας τροφοδοσίας σε ένα MLP, τα δεδομένα ρέουν προς τα εμπρός από το επίπεδο εισόδου στο επίπεδο εξόδου. Οι νευρώνες στο MLP εκπαιδεύονται με τον αλγόριθμο εκμάθησης πίσω διάδοσης. Τα MLP έχουν σχεδιαστεί για να προσεγγίζουν κάθε συνεχή λειτουργία και μπορούν να επιλύσουν προβλήματα που δεν είναι γραμμικά διαχωρίσιμα. Κύριες περιπτώσεις χρήσης του MLP αποτελούν η αντιστοίχιση προτύπων σε κλάσεις, η αναγνώριση και η προσέγγιση.

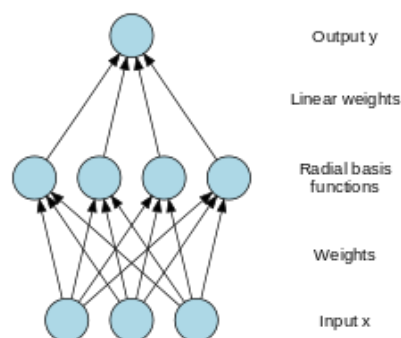


Εικόνα 1.10: Multilayer Perceptron

(Πηγή: <https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron>)

1.6.3 Radial Basis Function Νευρωνικό Δίκτυο (RBFNN)

Το νευρωνικό δίκτυο ακτινικής βάσης στην απλούστερη του μορφή είναι ένα νευρωνικό δίκτυο τριών επιπέδων εμπρόσθιας τροφοδοσίας. Το πρώτο επίπεδο αντιστοιχεί στις εισόδους του δικτύου, το δεύτερο είναι ένα κρυφό στρώμα που αποτελείται από έναν αριθμό μη γραμμικών μονάδων ενεργοποίησης RBF και το τελευταίο αντιστοιχεί στην τελική έξοδο του δικτύου.



Εικόνα 1.11: Radial Basis Function Νευρωνικό Δίκτυο

(Πηγή : https://en.wikipedia.org/wiki/Radial_basis_function_network)

Ένα διάνυσμα εισόδου x είναι η είσοδος σε όλες τις συναρτήσεις ακτινικής βάσης, καθεμία με διαφορετικές παραμέτρους. Η πληροφορία που εξάγεται από το δίκτυο αυτό προκύπτει αφού συνδυαστούν οι εξοδοί από τις συναρτήσεις ακτινικής βάσης. Εδώ, αντίθετα με το MLP, σε ένα πρόβλημα κατηγοριοποίησης, το δίκτυο κάνει το διαχωρισμό ανάλογα με τα υποδείγματα (κλάσεις) που έχει λάβει από το χρήστη, τα οποία αποτελούν τις κατηγορίες στις οποίες μπορεί να ανήκει η κάθε είσοδος. Αν μία είσοδος έχει ομοιότητες με τα στοιχεία που υπάρχουν σε μία συγκεκριμένη κατηγορία τότε εντάσσεται και αυτή σε αυτή την κατηγορία.

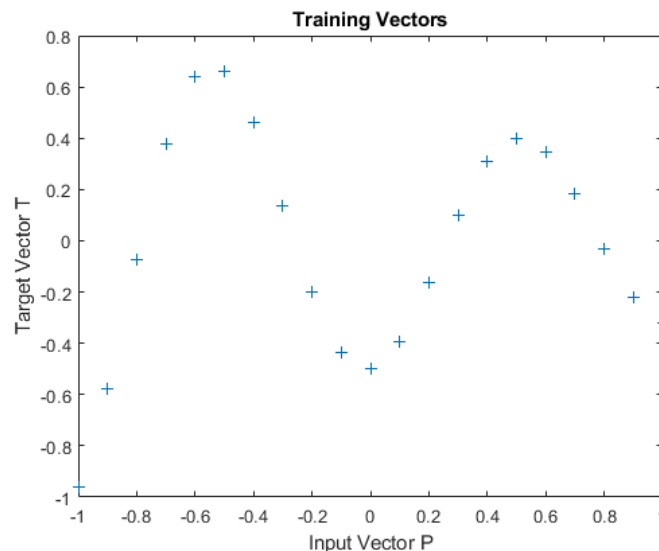
1.6.5 Learning Vector Quantization (LVQ):

Το δίκτυο κβαντοποίησης διανύσματος μάθησης είναι ένας τύπος Τεχνητού Νευρωνικού Δικτύου που εμπνέεται επίσης από βιολογικά μοντέλα νευρωνικών συστημάτων. Βασίζεται σε πρωτότυπο αλγόριθμο ταξινόμησης εποπτευόμενης μάθησης. Το LVQ διαθέτει ένα input layer και ένα output layer.

1.7 Παράδειγμα προβλήματος ΤΝΔ και επίλυση στο MATLAB

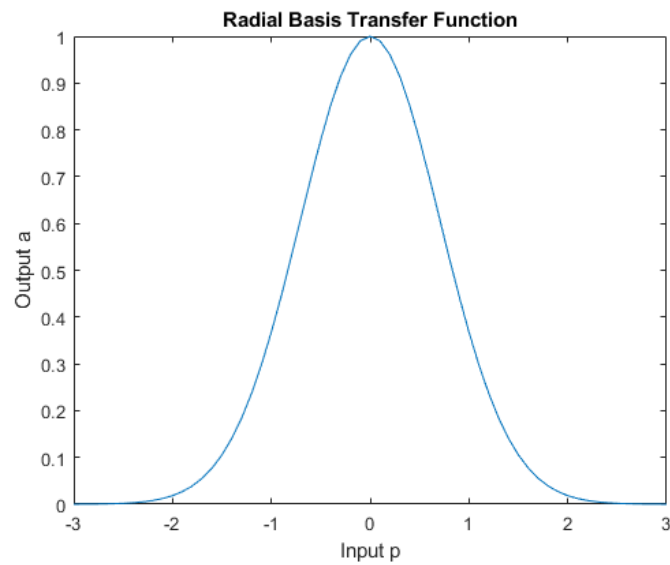
1.7.1 Προσέγγιση Συνάρτησης Radial Basis

Στο παρακάτω παράδειγμα δημιουργούμε ένα δίκτυο ακτινικής βάσης που προσεγγίζει μια συνάρτηση που ορίζεται από ένα σύνολο σημειακών δεδομένων. Ορίζουμε 21 εισόδους P και σχετικούς στόχους T .



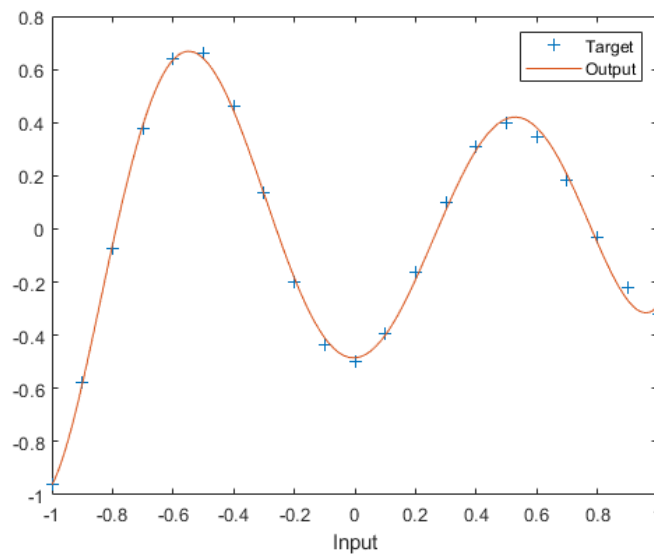
Εικόνα 1.12: Ορισμός εισόδων και στόχων.

Θα θέλαμε να βρούμε μια συνάρτηση που να ταιριάζει στα 21 σημεία δεδομένων. Αυτό μπορεί να επιτευχθεί με τη βοήθεια ενός δικτύου ακτινικής βάσης. Ένα δίκτυο ακτινικής βάσης είναι ένα δίκτυο με δύο επίπεδα. Ένα κρυφό στρώμα νευρώνων ακτινικής βάσης και ένα στρώμα εξόδου γραμμικών νευρώνων. Παρακάτω δίνουμε τη συνάρτηση μεταφοράς ακτινικής βάσης που χρησιμοποιείται από το κρυφό στρώμα.



Εικόνα 1.13: Συνάρτηση μεταφοράς ακτινικής βάσης.

Η συνάρτηση NEWRB δημιουργεί ένα δίκτυο ακτινικής βάσης που προσεγγίζει τη συνάρτηση που ορίζεται από τα P και T. Εκτός από το σύνολο εκπαίδευσης και τους στόχους, η NEWRB παίρνει δύο ορίσματα, το τετραγωνικό σφάλμα αθροίσματος και τη διασπορά. Παρακάτω φαίνεται η συνάρτηση που προσεγγίζει καταλληλότερα τα δεδομένα μας.



Εικόνα 1.14: Προσεγγιστική συνάρτηση basis function.

2. Βαθιά Μάθηση (Deep Learning)

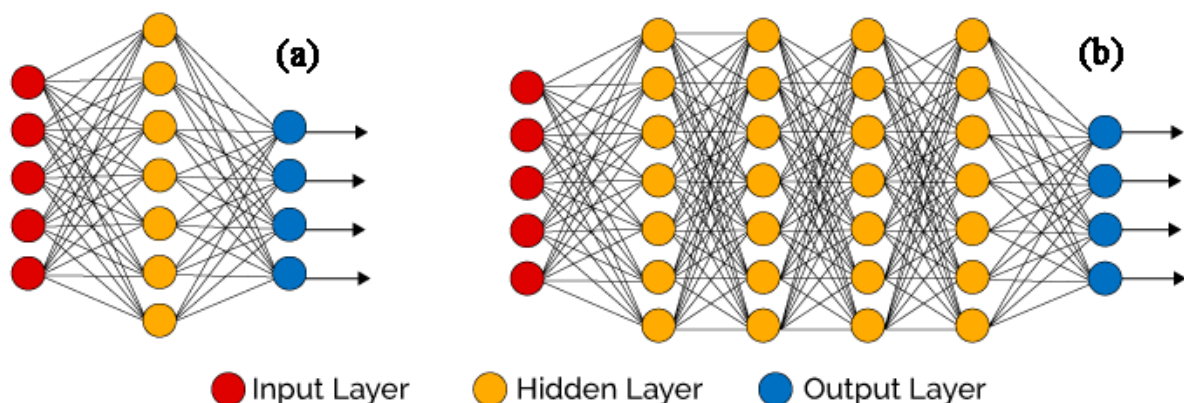
Με τον όρο «deep learning» περιγράφουμε έναν τρόπο μηχανικής μάθησης η οποία καθιστά τα υπολογιστικά συστήματα ικανά να αποκτήσουν γνώση βάσει της εμπειρίας από την εμπειρία και να αντιληφθούν το περιβάλλον με βάση μια ιεραρχία εννοιών. Σύμφωνα με το λεξικό, η βαθιά μάθηση είναι μία μορφή artificial intelligence βασισμένη σε λειτουργίες της ανθρώπινης σκέψης και συμπεριφοράς.

Ως εκ τούτου, η βαθιά μάθηση είναι ένας τομέας μάθησης αναπαράστασης. Επειδή ο υπολογιστής συλλέγει γνώση από την εμπειρία, δεν χρειάζεται άνθρωπος για να χειριστεί τον υπολογιστή και να καθορίσει τη γνώση που χρειάζεται ο υπολογιστής. Η βαθιά μάθηση είναι μια τεχνολογία λογισμικού που χρησιμοποιείται από τους προγραμματιστές για να διδάξουν στους υπολογιστές να πραγματοποιούν διεργασίες που ο άνθρωπος πραγματοποιεί από την αρχή της ύπαρξης του. Τα μοντέλα εκπαιδεύουν πολυεπίπεδους αλγόριθμους προκειμένου να επιτύχουν έναν συγκεκριμένο στόχο.

Τα μοντέλα βαθιάς μάθησης βασίζονται σε στρώματα τεχνητών νευρωνικών δικτύων (και όχι σε δεδομένα εισαγωγής) για να εκπαιδεύονται από προγραμματισμένες περιπτώσεις χαρακτηριστικών. Αυτά τα στρώματα πολλαπλών επιπέδων επιτρέπουν στα μοντέλα να εντοπίζουν λάθη και να εκπαιδεύονται από αυτά.

2.1 Βαθιά Νευρωνικά Δίκτυα

Τα μοντέλα νευρωνικών δικτύων έχουν γίνει ένα ισχυρό εργαλείο μηχανικής μάθησης και τεχνητής νοημοσύνης. Ένα βαθύ ΤΝΔ είναι ουσιαστικά ένα ΤΝΔ, το οποίο διαθέτει περισσότερα από ένα hidden layer ανάμεσα στο input layer και το output layer.



Εικόνα 2.1: Νευρωνικό δίκτυο (a) με 1 hidden layer και (b) με 4 hidden layers

(Πηγή : https://www.researchgate.net/figure/Artificial-Neural-Network-Vs-Deep-Neural-Network-14_fig1_344056345)

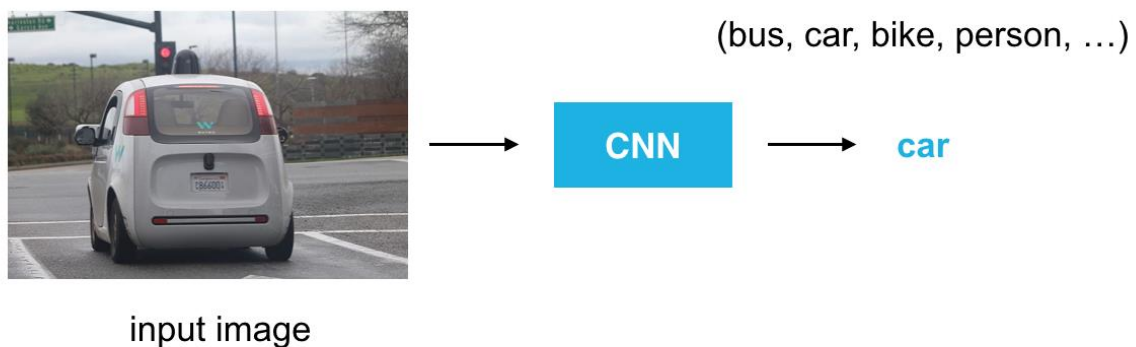
Τα βαθιά νευρωνικά δίκτυα εγκαθιστούν «βαθιές» αρχιτεκτονικές στα νευρωνικά δίκτυα. Ο όρος "βαθιά" αναφέρεται σε συναρτήσεις με μεγαλύτερη πολυπλοκότητα στον αριθμό των επιπέδων και των μονάδων. Τα μεγάλα σύνολα δεδομένων κατέστησαν δυνατή τη δημιουργία πιο ακριβών μοντέλων χρησιμοποιώντας πρόσθετα και μεγαλύτερα επίπεδα για την

καταγραφή πολυπλοκότερων μοτίβων. Τα βαθιά νευρωνικά δίκτυα, όπως και τα ρηχά εκτελούν γενικά εποπτευόμενες εργασίες μάθησης, χτίζοντας γνώση από σύνολα δεδομένων όπου η σωστή απάντηση παρέχεται εκ των προτέρων. Στη συνέχεια, τα δίκτυα μαθαίνουν να βρίσκουν τη σωστή απάντηση μόνα τους, αυξάνοντας την ακρίβεια των προβλέψεών τους.

2.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)

Τα ΣΝΔ χρησιμοποιούνται συχνότερα για ταξινόμηση και εργασίες όρασης υπολογιστή. Πριν από τα CNN, χρησιμοποιήθηκαν χειροκίνητες, χρονοβόρες μέθοδοι εξαγωγής χαρακτηριστικών για την αναγνώριση αντικειμένων στις εικόνες. Ωστόσο, τα συνελικτικά νευρωνικά δίκτυα παρέχουν τώρα μια πιο επεκτάσιμη προσέγγιση για την ταξινόμηση εικόνων και τις εργασίες αναγνώρισης αντικειμένων, αξιοποιώντας αρχές από τη γραμμική άλγεβρα, ειδικά διαδικασίες πολλαπλασιασμού πινάκων, με σκοπό την ανίχνευση συγκεκριμένων πληροφοριών στις εικόνες.

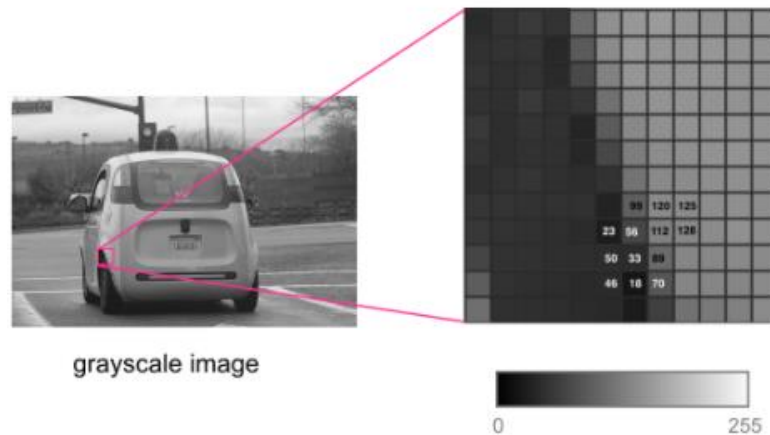
Στην περίπτωση της ταξινόμησης εικόνων, θέλουμε ένα νευρωνικό δίκτυο που κοιτάζει μια εικόνα και βγάζει τη σωστή κλάση για αυτήν την εικόνα. Ας πάρουμε μια εικόνα ενός αυτοκινήτου ως παράδειγμα.



Εικόνα 2.2: Παράδειγμα ταξινόμησης εικόνας

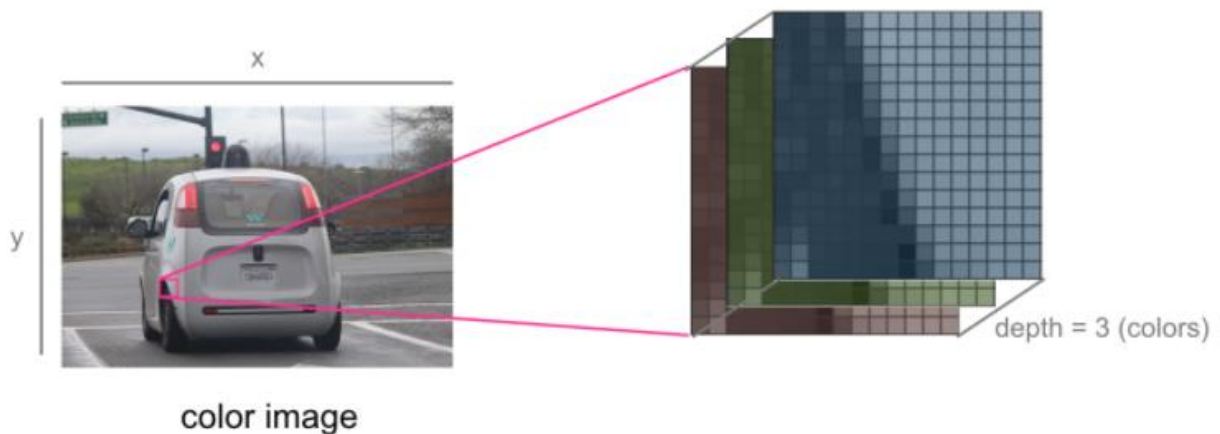
(Πηγή : https://cezannec.github.io/Convolutional_Neural_Networks/)

Μια εικόνα θεωρείται από ένα νευρωνικό δίκτυο (και από τους υπολογιστές) ως ένα πλέγμα αριθμητικών τιμών. Για εικόνες σε κλίμακα του γκρι, η τιμή του κάθε εικονοστοιχείου κυμαίνεται από 0 έως 255, όπου το 0 είναι μαύρο και το 255 είναι λευκό. Για μια τυπική έγχρωμη εικόνα, υπάρχουν τιμές κόκκινου, πράσινου και μπλε pixel για κάθε θέση pixel (x,y).



Εικόνα 2.3: Γκρι εικόνα υπό την αντίληψη του υπολογιστή

(Πηγή : https://cezannec.github.io/Convolutional_Neural_Networks/)



Εικόνα 2.4: Έγχρωμη εικόνα υπό την αντίληψη του υπολογιστή

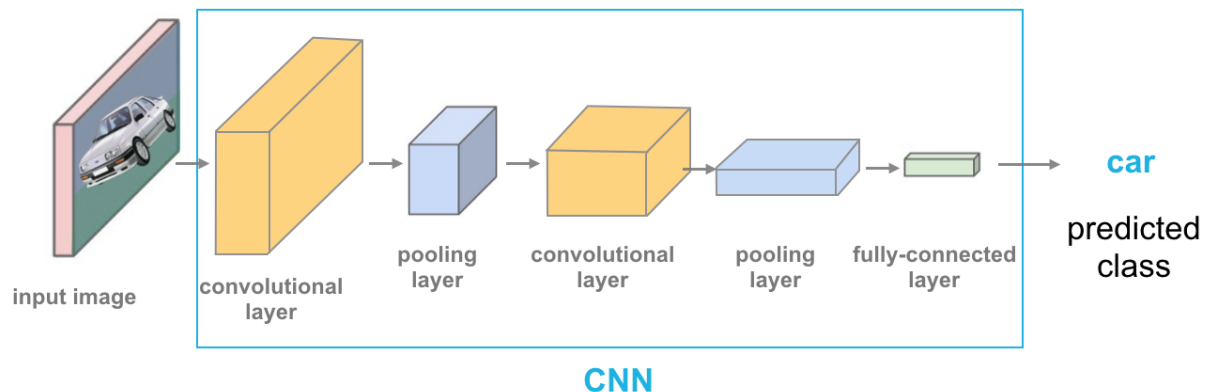
(Πηγή : https://cezannec.github.io/Convolutional_Neural_Networks/)

Για να δημιουργήσουμε έναν ταξινομητή εικόνας, χρειαζόμαστε έναν αλγόριθμο που μπορεί να εξετάσει αυτές τις τιμές pixel και να ταξινομήσει αυτήν την εικόνα ως αυτοκίνητο.

Τα ΣΝΔ ξεχωρίζουν από άλλου τύπου ΤΝΔ λόγω της επιβεβαιωμένης υψηλότερης αποτελεσματικότητας τους σε περιπτώσεις όπου τα δεδομένα που εισάγονται στο δίκτυο είναι εικόνες ή ήχοι. Αποτελούνται από τα παρακάτω βασικά στρώματα:

- Convolutional layer
- Pooling layer
- Fully-connected layer

Το convolutional layer εμφανίζεται στην αφετηρία του δικτύου. Των στρωμάτων αυτών είναι δυνατόν να έπονται επιπλέον όμοια στρώματα ή pooling layers, με το fully-connected layer να αποτελεί το τελευταίο layer.

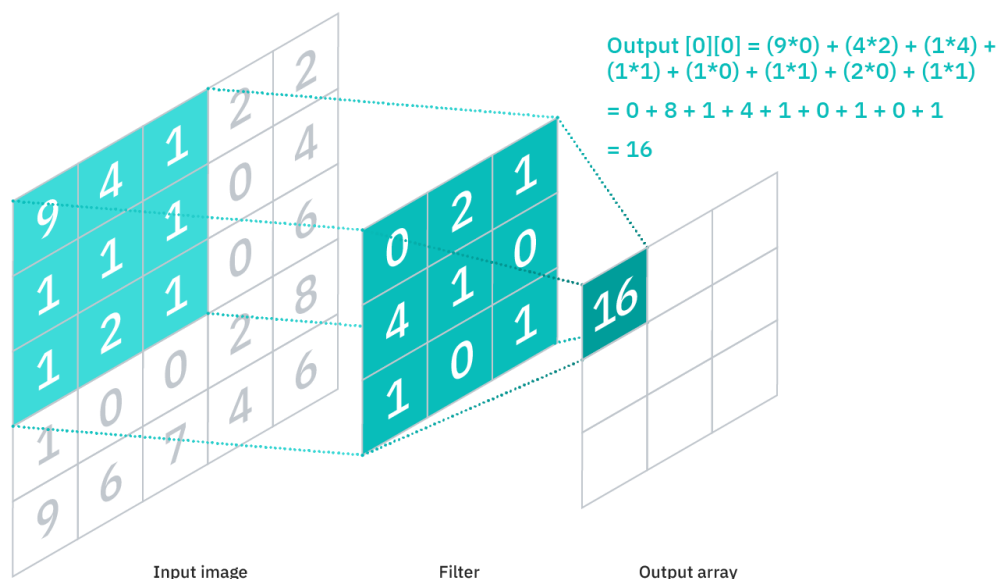


Εικόνα 2.5: Στρώματα συνελκτικού ΤΝΔ

(Πηγή : https://cezannec.github.io/Convolutional_Neural_Networks/)

➤ Convolutional layer

Το στρώμα αυτό αποτελεί θεμέλιο λίθο του κάθε ΣΝΔ και σε αυτό γίνεται η πλειονότητα επεξεργαστικών διαδικασιών. Προαπαιτούμενα για τη λειτουργία του στρώματος αυτού είναι τα δεδομένα εισόδου, τα ένα φίλτρα και οι χάρτες χαρακτηριστικών. Έστω ότι έχουμε μία rgb εικόνα, τότε ως συνέλιξη περιγράφουμε την σάρωση της έκτασης της εικόνας από ένα φίλτρο (χάρτη χαρακτηριστικών), που επιβεβαιώνει την παρουσία ή μη του εκάστοτε χαρακτηριστικού. Το φίλτρο αυτό είναι συνήθως ένας πίνακας 3x3 που σαρώνει μία συγκεκριμένη περιοχή της εικόνας με σκοπό τον υπολογισμό έπειτα του εσωτερικού γινομένου των εικονοστοιχείων της εικόνας και του φίλτρου και το αποτέλεσμα αυτό παρέχεται στον πίνακα εξόδου.



Εικόνα 2.6: Convolutional layer

(Πηγή: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>)

Πρώτου το ΣΝΔ εκπαιδευτεί είναι απαραίτητο να προκαθοριστούν οι παρακάτω παράμετροι του συνελκτικού στρώματος:

- Ο **αριθμός των φίλτρων** (number of filters) καθορίζει το πόσο βαθιά θα είναι η έξοδος ή αλλιώς, πόσους χάρτες χαρακτηριστικών θα προκύψουν.
- Ο **διασκελισμός** (stride) είναι το βήμα του φίλτρου πάνω στην εικόνα , δηλαδή ο αριθμός των εικονοστοιχείων που μεταπηδά. Μεγάλο βήμα ισοδυναμεί σε χαμηλότερη απόδοση.
- Η **επένδυση μηδενικών** (zero padding) εφαρμόζεται όταν το φίλτρο δεν ταιριάζει απόλυτα στην είσοδο. Έτσι τίθενται όλα τα εικονοστοιχεία που πέφτουν έξω από τη μήτρα εισόδου ίσα με μηδέν, παράγοντας μια μεγαλύτερη ή εξίσου μεγέθους έξοδο.

Ακολουθώντας το κάθε συνελκτικό στρώμα, συναντάται η εφαρμογή μίας συνάρτησης ανόρθωσης στους χάρτες χαρακτηριστικών.

➤ Pooling Layer

Στο στρώμα αυτό μειώνει τις διαστάσεις των εισόδων, ελαχιστοποιώντας τις παραμέτρους τους. Με παραπλήσιο τρόπο με το προηγούμενο layer, εδώ ελέγχει το φίλτρο όλο το εύρος της εικόνας, χωρίς όμως την ύπαρξη βαρών. Υπάρχουν δύο είδη pooling:

- Average pooling :

Κατά την εφαρμογή του στην εικόνα εισόδου, το φίλτρο, εντοπίζει τη μέση τιμή εντός του πεδίου υποδοχής και τη αποστέλλει στην έξοδο .

- Max pooling:

Κατά την εφαρμογή του στην εικόνα εισόδου, το φίλτρο, εντοπίζει το εικονοστοιχείο με τη μεγαλύτερη τιμή και αποστέλλει την τιμή αυτή στην έξοδο .

Ενώ πολλές πληροφορίες χάνονται στο pooling layer, μειώνεται η πολυπλοκότητα, βελτιώνεται η αποτελεσματικότητα και επίσης παρατηρούμε περιορισμό του κινδύνου του overfitting.

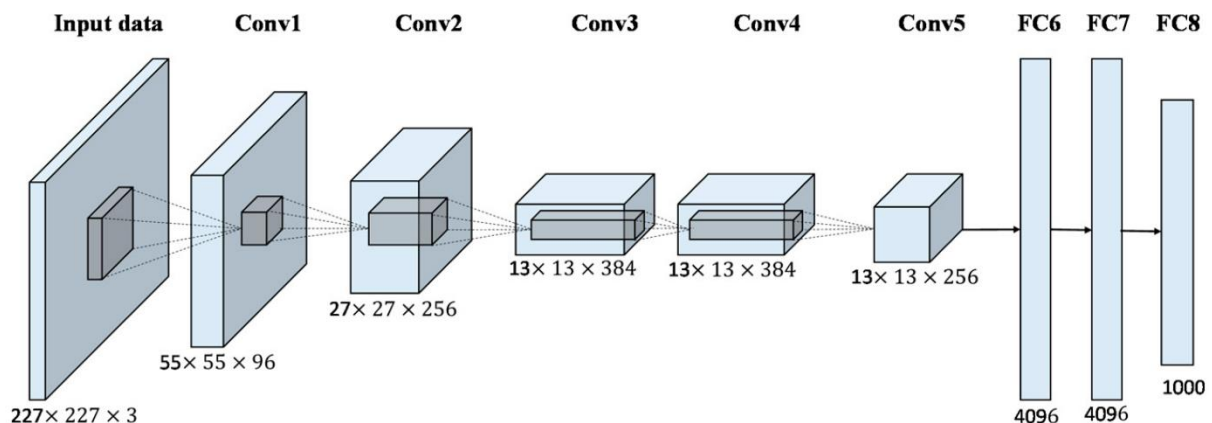
➤ Fully-Connected Layer

Αυτό το επίπεδο εκτελεί την διεργασία της αντιστοίχισης βασιζόμενο στα εξαγόμενα χαρακτηριστικά από τα προηγούμενα επίπεδα. Σε αντίθεση με τα στρώματα που προηγήθηκαν, τα πλήρως συνδεδεμένα στρώματα χρησιμοποιούν συναρτήσεις softmax αντί για συναρτήσεις ανόρθωσης, για να αντιστοιχίσουν με τον κατάλληλο τρόπο την εκάστοτε είσοδο, με την παραγωγή μίας τιμής πιθανότητας σε εύρος από 0 έως 1.

2.3.1 Δημοφιλή συνελκτικά Νευρωνικά Δίκτυα

➤ Alexnet:

Για διεργασίες αντιστοίχισης εικόνων σε κλάσεις, ως το πρώτο συνελκτικό νευρωνικό δίκτυο που στέφθηκε νικητής στο ImageNet Challenge το 2012, το AlexNet περιλαμβάνει στην αρχιτεκτονική του πέντε επίπεδα συνέλιξης και τρία πλήρως συνδεδεμένα επίπεδα. Έτσι, το AlexNet απαιτεί 61 εκατομμύρια βάρη και 724 εκατομμύρια MAC (multiply-add computation) για να ταξινομήσει την εικόνα με μέγεθος 227×227 .

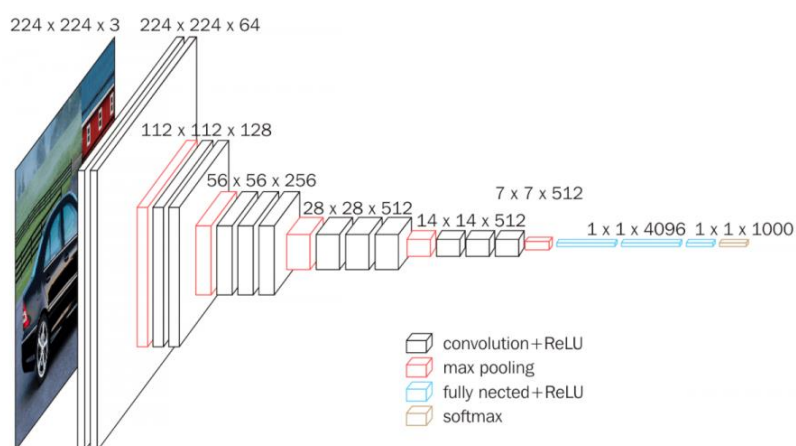


Εικόνα 2.7: Alexnet

(Πηγή : <https://www.mdpi.com/2072-4292/9/8/848>)

➤ VGG-16:

Για την επίτευξη υψηλότερης ακρίβειας, το VGG-16 εκπαιδεύεται σε μια βαθύτερη δομή 16 στρωμάτων που αποτελείται από 13 στρώματα συνέλιξης και τρία πλήρως συνδεδεμένα στρώματα, που απαιτούν 138 εκατομμύρια βάρη και 15.5G MAC για την ταξινόμηση της εικόνας με μέγεθος 224×224 .

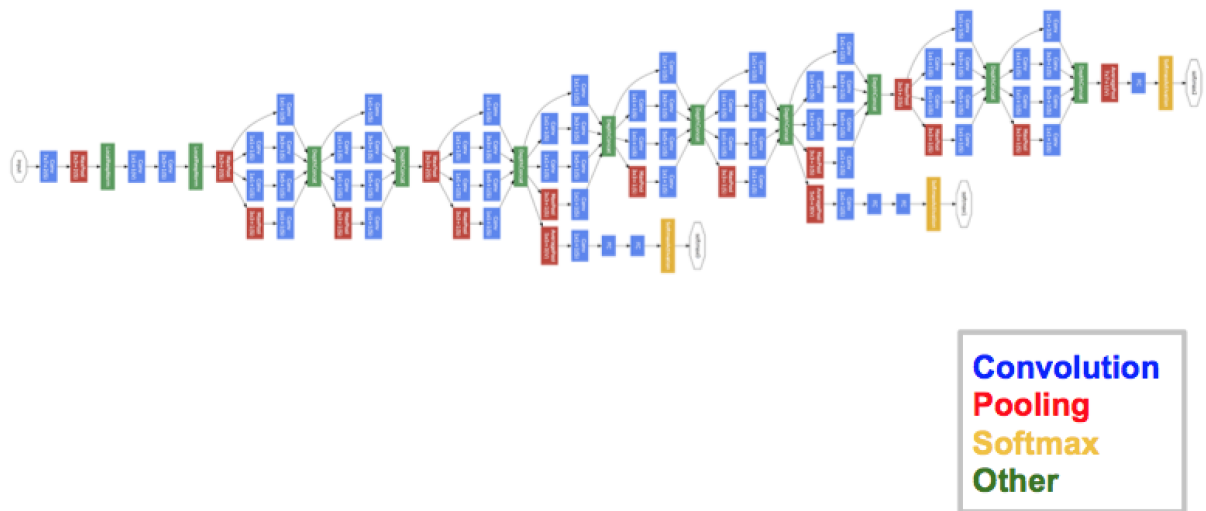


Εικόνα 2.8: VGG-16

(Πηγή : <https://neurohive.io/en/popular-networks/vgg16/>)

➤ GoogleNet:

Για τη βελτίωση της ακρίβειας μειώνοντας παράλληλα την υπολογιστική ισχύ, το GoogleNet εισάγει μια μονάδα έναρξης που αποτελείται από φίλτρα διαφορετικού μεγέθους. Ως αποτέλεσμα, το GoogleNet επιτυγχάνει καλύτερη απόδοση ακρίβειας από το VGG-16, ενώ απαιτούνται μόνο επτά εκατομμύρια βάρη και 1,43 G MAC για την επεξεργασία της εικόνας με το ίδιο μέγεθος.

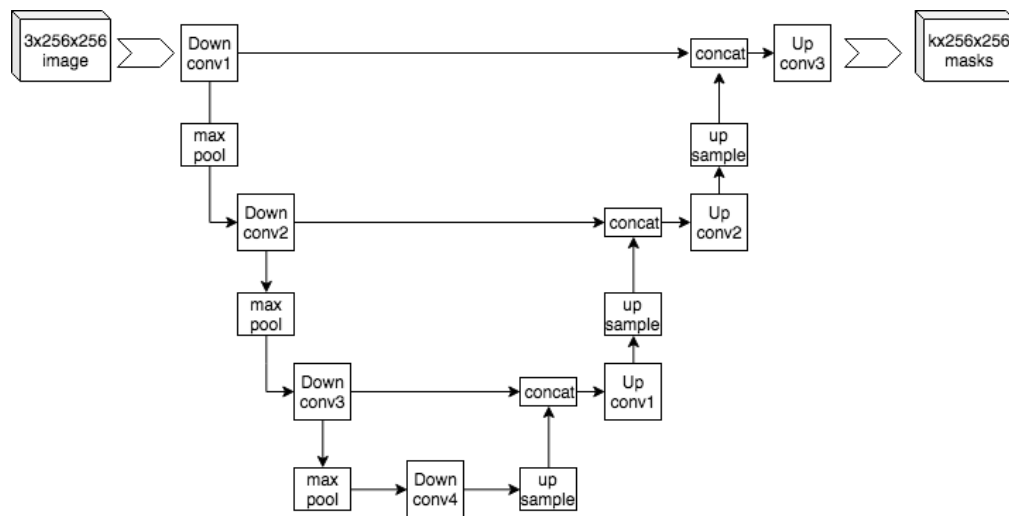


Εικόνα 2.9: Googlenet

(Πηγή: <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>)

➤ U-Net :

Το U-Net είναι ένα συνελκτικό νευρωνικό δίκτυο που αναπτύχθηκε για την κατάτμηση βιοϊατρικής εικόνας. Το δίκτυο βασίζεται στα πλήρως συνελκτικά δίκτυα και η αρχιτεκτονική του τροποποιήθηκε και επεκτάθηκε ώστε να λειτουργεί με λιγότερες εικόνες εκπαίδευσης και να παρέχει εξίσου ακριβείς κατατμήσεις. Οι λειτουργίες συγκέντρωσης (pooling) αντικαθίστανται από λειτουργίες upsampling. Επομένως, τα στρώματα αυτά προσφέρουν έξοδο μεγαλύτερης ανάλυσης. Υπάρχει ένας μεγάλος αριθμός καναλιών χαρακτηριστικών στα τμήματα του upsampling, τα οποία θέτουν το δίκτυο σε θέση να διαδίδει δεδομένα σε στρώματα μεγαλύτερης ανάλυσης. Η αρχιτεκτονική του περιγράφεται από ένα μονοπάτι συστολής και ένα διαστολής και έτσι το δίκτυο παίρνει τη μορφή «U». Το μονοπάτι της συστολής αποτελείται από συνεχείς συνελίξεις ακολουθούμενες από μία συνάρτηση ReLU και μία διαδικασία max pooling. Εδώ η χωρική πληροφορία μειώνεται, ενώ η πληροφορία που αφορά τα χαρακτηριστικά της εικόνας αυξάνεται. Το μονοπάτι διαστολής συνδυάζει τη χωρική πληροφορία με την πληροφορία που αφορά τα χαρακτηριστικά και μέσω upsampling, λαμβάνουμε την επιθυμητή εικόνα.



Εικόνα 2.10: U-net

(Πηγή: https://en.wikipedia.org/wiki/U-Net#/media/File:Example_architecture_of_U-Net_for_producing_k_256-by-256_image_masks_for_a_256-by-256_RGB_image.png)

3. Ψηφιακή Επεξεργασία Εικόνας

Η ταχύτατη εξέλιξη των μεταγενέστερων μορφών τεχνολογίας τα τελευταία χρόνια έχει οδηγήσει στην εξέλιξη κάποιων επιμέρους χώρων τους. Στον χώρο των πολυμέσων παρατηρείται πλέον ιδιαίτερα ογκώδης πολυμεσική πληροφορία, η οποία κατέστησε αναγκαίο να βρεθούν τρόποι, ώστε να αξιοποιηθεί η πληροφορία αυτή. Αναφορικά με τις εικόνες, το ζήτημα αυτό βρίσκει λύση μέσω της Ψηφιακής Επεξεργασίας Εικόνας.

Η **Ψηφιακή Επεξεργασία Εικόνας** (ΨΕΕ) ή **Digital Image Processing** είναι ένας κλάδος της επιστήμης ο οποίος ασχολείται με το χειρισμό μιας εικόνας της οποίας έχει γίνει λήψη, δηλαδή τις τεχνικές ανάκτησης, ψηφιοποίησης, ανάλυσης, αποθήκευσης, μετάδοσης, συμπίεσης, βελτίωσης, αποκατάστασης μιας εικόνας. Σήμερα συναντάμε πληθώρα εφαρμογών της ΨΕΕ, όπως στη ρομποτική όραση, στην ιατρική εικόνα (τομογραφίες, υπερήχους), στις αεροφωτογραφίες, στην αναγνώριση προτύπων, στη μετάδοση (τηλεομοιοτυπία, τηλεδιάσκεψη), στα συστήματα ασφαλείας, στα γραφικά κλπ. Ακόμα στη δημιουργία και μετάδοση στερεοσκοπικής εικόνας για τηλεοπτικές μεταδόσεις υψηλής ανάλυσης (3-DTV), τρισδιάστατων ιατρικών εικόνων κλπ. Σημαντική είναι η διάκριση με την επιστήμη των γραφικών, η οποία ασχολείται κυρίως με τη δημιουργία συνθετικών εικόνων. Στο κεφάλαιο αυτό γίνεται αναφορά σε διάφορες θεμελιώδεις έννοιες.

3.1 Ιστορική αναδρομή της ψηφιακής επεξεργασίας εικόνας

Μεγάλος αριθμός τεχνικών της Ψ.Ε.Ε, είδε την ανάπτυξή του κατά τη δεκαετία του 1960, στα εργαστήρια Bell, στο εργαστήριο Jet Propulsion, στο MIT, στο πανεπιστήμιο του Maryland και σε διάφορες άλλες ερευνητικές εγκαταστάσεις, για χρήση σε δορυφορικές εικόνες, μετατροπή προτύπων καλωδιακής φωτογραφίας, αναπαράσταση βιοϊατρικών εικόνων, διάκριση γραμμάτων και ωραιοποίηση φωτογραφίας.

Στα πρώτα στάδια την ανάπτυξης της επεξεργασίας εικόνας στόχο αποτελούσε η εμπλουτισμός της ποιότητας των εικόνων, με απώτερο σκοπό την ευκολότερη αναγνώριση λεπτομερειών των εικόνων από το ανθρώπινο μάτι. Στην ψηφιακή επεξεργασία εικόνας, η είσοδος ήταν μια εικόνα με αρκετά χαμηλή ποιότητα και η έξοδος είναι μια εικόνα με κάποιου είδους βελτίωση.

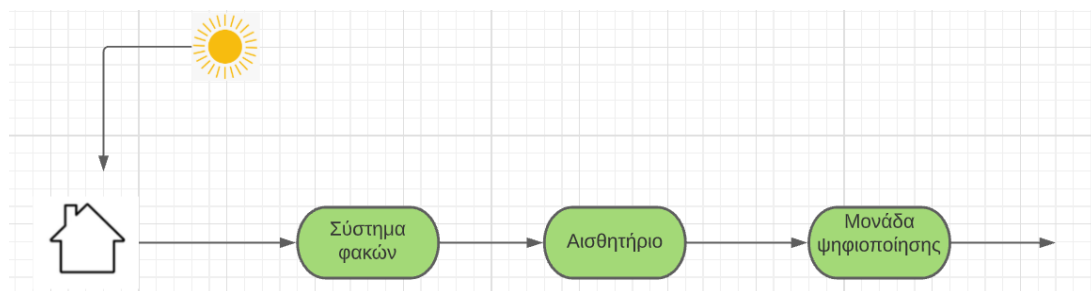
Η πρώτη περίπτωση ικανοποιητικού αποτελέσματος επεξεργασίας της ψηφιακής εικόνας έλαβε χώρα στο εργαστήριο Jet Propulsion (JPL) στην Αμερική. Εκεί έγινε χρήση μεθόδων επεξεργασίας εικόνας όπως γεωμετρική διόρθωση, μεταβολή βαθμίδας, αφαίρεση θορύβου κ.λπ. σε έναν μεγάλο αριθμό φωτογραφιών της σεληνιακής επιφάνειας που μεταφέρθηκαν με επιτυχία πίσω στη Γη από το Ranger 7 (πρώτος διαστημικός ανιχνευτής των Ηνωμένων Πολιτειών) το 1964, λαμβάνοντας υπόψη τη θέση του ήλιου και την επιφάνεια της σελήνης. Η πολύ ικανοποιητική χαρτογράφηση της σεληνιακής επιφάνειας αποτέλεσε ένα σημαντικό επίτευγμα. Μετά από κάποιο χρονικό διάστημα, έγινε μία πιο ενδεδειγμένη ψηφιακή επεξεργασία εικόνας πάνω στις χιλιάδες εικόνες που στάλθηκαν πίσω από το διαστημόπλοιο, με αποτελέσαμε να δημιουργηθεί ο τοπογραφικός χάρτης, ο χρωματικός χάρτης και το πανοραμικό ψηφιδωτό του της σελήνης, ανακαλύψεις που έθεσαν γερά θεμέλια για την επερχόμενη προσγείωση επανδρωμένων σκαφών στην επιφάνεια της σελήνης.

Λόγω του περιορισμένου και δυσεύρετου υπολογιστικού εξοπλισμού της τότε δεκαετίας, η επεξεργασία που πραγματοποιήθηκε απαιτούσε μεγάλο χρηματικό προϋπολογισμό. Στα τέλη της δεκαετίας του 1970, οι εφαρμογές της ψηφιακής επεξεργασίας πληθύναν καθώς υπήρξε διαθεσιμότητα μεγαλύτερου αριθμού εξειδικευμένου για την Ψ.Ε.Ε. εξοπλισμού. Αυτό οδήγησε στην επεξεργασία εικόνων σε πραγματικό χρόνο, για ορισμένα ειδικά προβλήματα, όπως η μετατροπή προτύπων στον τομέα της τηλεόρασης. Αργότερα, οι κλασικοί υπολογιστές έγιναν ταχύτεροι και πιο ισχυροί, με αποτέλεσμα να αποτελούν τον κατά κύριο λόγο εξοπλισμό για εφαρμογές ψηφιακής επεξεργασίας εικόνων. Λόγω της ανάπτυξης της υπολογιστικής δύναμης από το 2000 και μετά, η ψηφιακή επεξεργασία εικόνας βρίσκει εφαρμογές σε πληθώρα τομέων και το κόστος έχει ελαχιστοποιηθεί.

3.2 Θεμελιώδεις έννοιες

3.2.1 Ψηφιακή Εικόνα

Ο βασικός τρόπος κατά τον οποίο δημιουργείται μια ψηφιακή εικόνα απεικονίζεται στο Σχήμα 3.1. Το αισθητήριο πραγματοποιεί τη μετατροπή του φωτός σε ηλεκτρικό ρεύμα, του οποίου γίνεται η δειγματοληψία, ο κβαντισμός και η ψηφιοποίηση με τη βοήθεια ενός μετατροπέα A/D.



Εικόνα 3.1: Τυπικό σύστημα παραγωγής ψηφιακής εικόνας

Οι ψηφιακές εικόνες ανήκουν σε μία από τις παρακάτω κατηγορίες:

- Η **μονοχρωματική** εικόνα αποτελεί ουσιαστικά ένα πίνακα $M \times N$ εικονοστοιχείων δύο διαστάσεων, του οποίου οι τιμές περιγράφουν τη φωτεινότητα ή intensity του φωτιζόμενου αντικειμένου :

$$I(i, j) \text{ με } i = 1, \dots, M \text{ και } j = 1, \dots, N \quad (3.1)$$

όπου $0 \leq I(i, j) \leq L - 1$. Το L είναι συχνά ίσο με μία δύναμη του 2, δηλαδή $L = 2^k$, με το k να παίρνει συνήθως την τιμή $k = 8$, στο οποίο αντιστοιχούν 256 αποχρώσεις του γκρι. Η τιμή $I(i, j)$ εξαρτάται από τη φωτεινότητα στο εικονοστοιχείο (i, j) . Συνήθως συναντάμε ψηφιακές εικόνες διαστάσεων με τιμές δυνάμεις του 2, για παράδειγμα 256×256 ή 512×512 .

- Ο δυαδικός τύπος είναι ο πιο απλοϊκός τύπος των ψηφιακών εικόνων. Σε μια εικόνα του τύπου αυτού συναντάμε φωτεινότητα δύο σταθμών, οι οποίες συχνά είναι το μαύρο και το άσπρο. Το μαύρο παίρνει τιμή ίση με 0 και το άσπρο ίση με 1. Οι εικόνες της κατηγορίας αυτής χρειάζονται λιγότερη μνήμη για την αποθήκευσή της και το κόστος που απαιτείται για την επεξεργασία της είναι αρκετά μικρό. Όταν οι εικόνες λάβουν τη μορφή αυτή, είναι

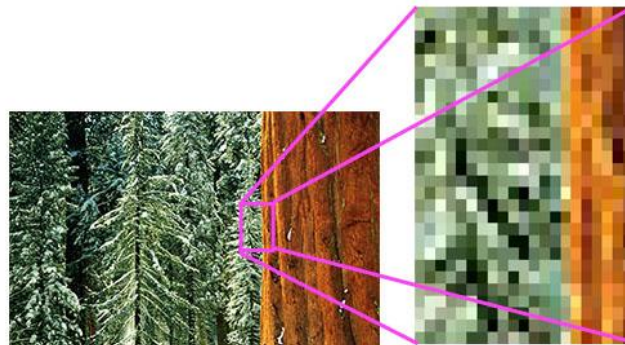
δυνατό να απεικονιστούν ιδιαίτερης σημασίας δεδομένα, με τα βασικότερα να είναι το εμβαδόν και η θέση αντικειμένων, η μορφή αντικειμένων κ.α. Αξίζει να σημειωθεί ότι πληθώρα εφαρμογών της Ψηφιακής επεξεργασίας την εικόνας, όπως η οπτική αναγνώριση χαρακτήρων, η αναγνώριση υπογραφής, η αναγνώριση αποτυπωμάτων, πραγματοποιούνται πολύ συχνά χρησιμοποιώντας εικόνες δυαδικού τύπου.

- Στην κατηγορία των **έγχρωμων** εικόνων ανήκουν οι εικόνες αυτές που απεικονίζουν τον πραγματικό κόσμο. Τρεις μονοχρωματικές εικόνες αποτελούν μια έγχρωμη ψηφιακή εικόνα. Επομένως, το χρώμα του εκάστοτε pixel διαθέτει τρεις συνιστώσες, οι οποίες αντιστοιχούνται στις φωτεινότητες των τριών μονοχρωματικών εικόνων.

Παραδείγματος χάρη, στο σύστημα χρωμάτων RGB, το εκάστοτε χρώμα είναι αποτέλεσμα του συνδυασμού των θεμελιωδών χρωμάτων, του κόκκινου (R), του πράσινου (G) και του μπλε (Blue).

3.2.2 Εικονοστοιχείο (Pixel)

Με τον όρο **Pixel** (pix = picture, el = element) αναφερόμαστε ουσιαστικά στο θεμελιώδες και ταυτόχρονα το μικρότερο πλήρες τμήμα μίας ψηφιακής εικόνας. Όσο πληθαίνουν τα pixel των οποίων γίνεται χρήση για την αναπαράσταση μίας εικόνας, τόσο πιο κοντά είναι το αποτέλεσμα της αναπαράστασης στο πραγματικό. Το σύνολο των pixel σε μια εικόνα αναφέρεται συχνά ως ανάλυση.



Εικόνα 3.2: Παράδειγμα εικονοστοιχείων

(Πηγή: <https://sites.google.com/site/cameraandphotocam/all-about-pixels>)

3.4 Χαρακτηριστικά Ψηφιακής Εικόνας

Τα χαρακτηριστικά μίας εικόνας ή διαφορετικά οι πληροφορίες που αφορούν μία εικόνα, μπορεί να σχετίζονται είτε έμμεσα με το περιεχόμενο της, όπως για παράδειγμα η ανάλυση, ο τύπος αρχείου, οι διαστάσεις κλπ., είτε να σχετίζονται άμεσα, όπως είναι το βάθος χρώματος, η υφή κ.α. Το ανθρώπινο μάτι αντιλαμβάνεται τη δεύτερη κατηγορία χαρακτηριστικών.

3.4.1 Βάθος Χρώματος

Το βάθος χρώματος, που συναντάται και ως βάθος bit, σχετίζεται με το πλήθος των δυαδικών ψηφίων των οποίων γίνεται χρήση με σκοπό να αναπαρασταθούν τα χρώματα των pixel μίας

εικόνας. Όσο αυξάνεται το βάθος χρώματος, δηλαδή ο αριθμός των bit που χρησιμοποιούμε για το εκάστοτε χρώμα, τόσο πιο πολύ πληθαίνουν οι διαθέσιμες χρωματικές αποχρώσεις που μπορούν να περιέχονται σε μία εικόνα. Είναι φανερό ότι μία έγχρωμη εικόνα θα έχει αναγκαστική προϋπόθεση τη χρήση τριπλάσιου αριθμού ψηφίων από μία μονοχρωματική. Αν θεωρήσουμε το βάθος χρώματος x , τότε έχουμε σε μία εικόνα 2^x διαθέσιμες αποχρώσεις προς κωδικοποίηση.

Με color depth ενός bit, οι διαθέσιμες αποχρώσεις θα είναι $2^1 = 2$ αποχρώσεις. Γενικότερα, τα χρώματα αυτά είναι άσπρο και μαύρο, αλλά υπάρχει περίπτωση να είναι δύο άλλα διαφορετικά χρώματα. Σε αυτή την κατηγορία συναντάμε τις προαναφερόμενες δυαδικές εικόνες ή μονοχρωματικές.

Οι εικόνες με αποκλειστικά και μόνο αποχρώσεις του γκρι, είναι αποτέλεσμα αποχρώσεων που συναντώνται μεταξύ του μαύρου και του άσπρου. Το color depth είναι 8, άρα διατίθενται 256 διαφορετικές αποχρώσεις, όπου το μαύρο εμπεριέχει την τιμή 0 και το άσπρο την τιμή 255. Όσο σκουραίνουν τα pixel της εικόνας αντιστοιχίζονται σε τιμές κοντά στο 0, ενώ τα pixel με πιο ανοιχτές αποχρώσεις αντιστοιχίζονται σε τιμές πιο κοντά στο 256. Παρακάτω μπορούμε να δούμε πώς η ίδια εικόνα αλλάζει σύμφωνα με τον αριθμό των bit που χρησιμοποιούνται:



Εικόνα 3.3: 1 bit βάθος χρώματος / 2 χρώματα

(Πηγή:http://repfiles.kallipos.gr/html_books/50/Chapter_5/index.html#5.2.)



Εικόνα 3.4: 2 bit βάθος χρώματος / 4 χρώματα

(Πηγή:http://repfiles.kallipos.gr/html_books/50/Chapter_5/index.html#5.2.)



Εικόνα 3.5: 4 bit βάθος χρώματος / 16 χρώματα

(Πηγή:http://repfiles.kallipos.gr/html_books/50/Chapter_5/index.html#5.2.)



Εικόνα 3.6: 8 bit βάθος χρώματος / 256 χρώματα

(Πηγή:http://repfiles.kallipos.gr/html_books/50/Chapter_5/index.html#5.2.)



Εικόνα 3.7: 24 bit βάθος χρώματος / 16.777.216 χρώματα

(Πηγή:http://repfiles.kallipos.gr/html_books/50/Chapter_5/index.html#5.2.)

Οι εικόνες για τις οποίες χρησιμοποιούνται 24 bit συναντώνται και ως true color εικόνες και έχουν στη διάθεση τους ένα σύνολο 16.777.216 αποχρώσεων . Αυτό οφείλεται στο RGB χρωματικό μοντέλο , όπου για κάθε ένα από τα τρία βασικά χρώματα διατίθενται 8 bit. Έτσι έχουμε 256 αποχρώσεις για το εκάστοτε βασικό χρώμα και επομένως 256^3 συνολικές αποχρώσεις.

3.4.2 Ανάλυση

Με τον όρο ανάλυση εικόνας (image resolution) εκφράζουμε το πλήθος των pixels που στην αντίστοιχη μονάδα της επιφάνειας και έχει συχνά την μονάδα μέτρησης pixels/in² ή dpi (dots per inch). Η έννοια της ανάλυσης είναι συνδεδεμένη στενά με την έννοια της ευκρίνειας της εικόνας , δηλαδή του πόσο ευδιάκριτες είναι η λεπτομέρειες μίας εικόνας.

Παρακάτω μπορούμε να δούμε πώς η ίδια εικόνα αλλάζει σύμφωνα με την μεταβολή της ανάλυσης της (μείωση dpi):



Εικόνα 3.8: 24 Μείωση από 300dpi σε 150dpi , 72 dpi και 36 dpi

(Πηγή <https://malcolmtattersall.com.au/wp/2012/06/greener-computing-newsletters/>)

3.4.3 Υφή

Η υφή αποτελεί ένα σημαντικό χαρακτηριστικό το οποίο μπορούμε να λάβουμε από μία εικόνα και του οποίου γίνεται χρήση συχνά στην κατάτμηση εικόνων και πιο συγκεκριμένα στην αναγνώριση αντικειμένων ή περιοχών μίας εικόνας. Αν και η υφή της εικόνας δεν είναι ένα τέλειο μέτρο για την κατάτμηση, χρησιμοποιείται μαζί με άλλα μέτρα, όπως το χρώμα, για ένα πιο ακριβές αποτέλεσμα κατάτμησης. Η υφή χαρακτηρίζει κάθε επιφάνεια και μας προσφέρει δεδομένα που αφορούν τη δομή των αντικειμένων. Με μία περισσότερο μαθηματική προσέγγιση θα λέγαμε ότι είναι η αποτύπωση της διακύμανσης που έχουν τα εικονοστοιχεία μιας εικόνας που απεικονίζουν ένα συνεχές αντικείμενο. Οι υφές μίας εικόνας μπορούν να δημιουργηθούν τεχνητά ή να βρεθούν σε φυσικές σκηνές που αποτυπώνονται σε μια εικόνα.

Κάποιες παράμετροι που επηρεάζουν την υφή είναι το χρώμα, η λάμψη, η ανακλαστικότητα, η διαφάνεια, η διαύγεια και η ομαλότητα.



Εικόνα 3.9: Τεχνητά δημιουργημένη υφή (αριστερά) και υφή σε φυσική σκηνή (δεξιά).

(Πηγή https://en.wikipedia.org/wiki/Image_texture#Region_Based)

3.4.4 Τύπος αρχείου εικόνας

Οι μορφές αρχείων εικόνας είναι τυποποιημένα μέσα οργάνωσης και αποθήκευσης ψηφιακών εικόνων. Μια μορφή αρχείου εικόνας έχει τη δυνατότητα αποθήκευσης της πληροφορίας σε συμπιεσμένη μορφή, σε συμπίεσμένη μορφή (με πιθανότητα κάποιων απωλειών) ή σε διανυσματική μορφή. Τα αρχεία εικόνας αποτελούνται από ψηφιακά δεδομένα σε μία από αυτές τις μορφές, έτσι ώστε αυτά να μπορούν να τυποποιηθούν για υπολογιστική κυρίως χρήση. Η ραστεροποίηση μετατρέπει τα δεδομένα εικόνας σε ένα σύνολο από pixels. Κάθε εικονοστοιχείο έχει έναν αριθμό bits για τον καθορισμό το χρώματός του. Η προαναφερθείσα διαδικασία εξαρτάται από το βάθος χρώματος που υποστηρίζει η εκάστοτε συσκευή.

Παρακάτω θα αναφέρουμε τις πιο διαδεδομένες μορφές.

- Το **JPEG** (Joint Photographic Experts Group) είναι μια μέθοδος συμπίεσης όπου παρατηρείται απώλεια. Συχνά γίνεται αποθήκευση των συμπιεσμένων εικόνων JPEG σε μορφή αρχείου JFIF (JPEG File Interchange Format). Η επέκταση ονόματος αρχείου JPEG/JFIF είναι JPG ή JPEG. Σήμερα η μεγάλη πλειονότητα των φωτογραφικών μηχανών είναι ικανή να αποθηκεύσει εικόνες σε μορφή JPEG/JFIF, η οποία υποστηρίζει εικόνες των 8 bit σε gray-scale και έγχρωμες εικόνες των 24 bit. Η συμπίεση μέσω της μεθόδου

αυτής έχει ως απόρροια τη σμίκρυνση του μεγέθους του αρχείου της εικόνας και ανάλογα με το βαθμό της συμπίεσης

- Το **GIF** (Graphics Interchange Format) συναντά περιορισμό τη χρήση 8 bit ή 256 χρωμάτων (ενώ το βάθος χρώματος 24 bit είναι τεχνικά εφικτό). Το GIF είναι το πιο κατάλληλο για αποθήκευση γραφικών με λίγα χρώματα, όπως απλά διαγράμματα, σχήματα, λογότυπα και εικόνες στυλ κινουμένων σχεδίων, καθώς χρησιμοποιεί συμπίεση LZW χωρίς απώλειες, η οποία είναι πιο αποτελεσματική όταν οι μεγάλες περιοχές έχουν ένα χρώμα και λιγότερο αποτελεσματική για φωτογραφικές ή απογοητευμένες εικόνες. Λόγω της απλότητας και της ηλικίας του GIF, πέτυχε σχεδόν καθολική υποστήριξη λογισμικού. Λόγω των δυνατοτήτων κίνησης, εξακολουθεί να χρησιμοποιείται ευρέως για την παροχή εφέ κίνησης εικόνας, παρά τον χαμηλό λόγο συμπίεσης σε σύγκριση με τις σύγχρονες μορφές βίντεο.
- Η μορφή αρχείου **PNG** (Portable Network Graphics) φτιάχτηκε ως δωρεάν εναλλακτική λύση ανοιχτού κώδικα αντί του GIF. Η μορφή αρχείου PNG αφορά εικόνες βάθους χρώματος έως 24 bit (16 εκατομμύρια χρώματα). Σε σύγκριση με το JPEG, το PNG υπερέρχει όταν η εικόνα έχει μεγάλες, ομοιόμορφα χρωματισμένες περιοχές.

3.5 Χρωματικά Συστήματα

Η φύση του φωτός που ανακλάται στο μάτι του ανθρώπου καθορίζει το χρώμα οποιουδήποτε αντικείμενου βλέπουμε. Το ορατό φως απαρτίζεται από ένα περιορισμένο εύρος συχνοτήτων ολόκληρου του ηλεκτρομαγνητικού φάσματος. Όταν το σύνολο των μηκών κύματος που ανήκουν στο ορατό φάσμα αντανακλάται ομοίως από ένα αντικείμενο, τότε αυτό φαίνεται λευκό. Όμως όταν ένα αντικείμενο αντανακλά ένα μικρότερο συγκεκριμένο μέρος του οπτικού φάσματος, τότε αυτό φαίνεται να διαθέτει κάποια χρωματική απόχρωση. Παραδείγματος χάρη, ένα σώμα που φαίνεται στο ανθρώπινο μάτι να έχει το κόκκινο χρώμα αντανακλά το φως εκείνο με μήκος κύματος γύρω από τα 700 nm, ενώ απορροφά την πλειοψηφία της ενέργειας του υπόλοιπου φάσματος.

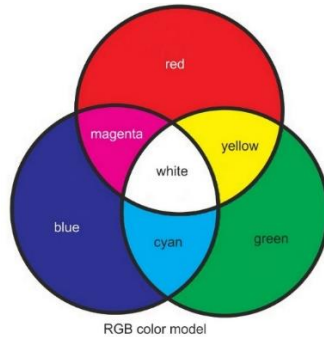
Στην περίπτωση άχρωμου φωτός, η μοναδική έννοια που μας ενδιαφέρει είναι η ένταση του φωτός. Χαρακτηριστικό παράδειγμα άχρωμου φωτός είναι αυτή ασπρόμαυρη τηλεόραση όπου παρατηρούμε μονάχα αλλαγές στην ποσότητα του φωτός. Έτσι εμφανίστηκαν ποικίλα επίπεδα του γκρι.

Λόγω του πως είναι δομημένο το μάτι μας, κάθε χρώμα είναι αποτέλεσμα του συνδυασμού των τριών βασικών χρωμάτων, του κόκκινου, του μπλε και του πράσινου. Τα τρία αυτά χρώματα έχουν τη δυνατότητα να δημιουργήσουν τα δευτερεύοντα χρώματα, τα οποία είναι το ματζέντα (κόκκινο + μπλε), το κυανό (μπλε + πράσινο) και το κίτρινο (πράσινο + μπλε). Αν και τα τρία βασικά χρώματα συνδυαστούν ταυτόχρονα, τότε δημιουργούν το άσπρο χρώμα, το οποίο δημιουργείται επίσης με το συνδυασμό ενός δευτερεύοντος με το αντίθετο βασικό χρώμα.

3.5.1 RGB

Το μοντέλο **RGB** αποτελεί το πλέον γνωστό χρωματικό σύστημα και χρησιμοποιείται ευρέως σε οθόνες τηλεοράσεων και υπολογιστών. Το μοντέλο αυτό έχει τη δυνατότητα παραμετροποίησης των χρωμάτων που προβάλλονται σε μία έγχρωμη οθόνη. Η λειτουργία

του RGB έχει τα θεμέλια της στον συνδυασμό των τριών βασικών χρωμάτων , στα οποία έχει γίνει αναφορά και παραπάνω.



Εικόνα 3.10: Μοντέλο RGB

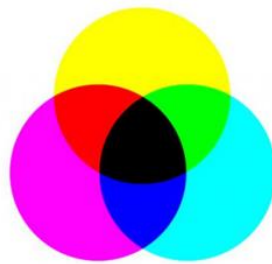
(Πηγή <https://www.vaklighting.com/light-and-color/>)

Το εκάστοτε χρώμα στο μοντέλο αυτό εξαρτάται από την ένταση καθενός από τα βασικά χρώματα, του κόκκινου, πράσινου και μπλε. Κάθε ένα από τα προαναφερθέντα χρώματα λαμβάνουν τιμές που κυμαίνονται από 0 ως 255 με τον συνδυασμό 0, 0, 0 να περιγράφει το απόλυτο μαύρο και το συνδυασμό 255, 255, 255 να αντιπροσωπεύει το απόλυτο άσπρο.

3.5.2 CMY

Το μοντέλο **CMY** ή **CMYK** κάνει χρήση των δευτερευόντων χρωμάτων που όπως έχουμε αναφέρει είναι το κυανό (Cyan) , το ματζέντα (Magenta) και το κίτρινο (Yellow) . Μέσω των παρακάτω σχέσεων μπορεί να γίνει μετάβαση στο μοντέλο RGB και αντίστροφα:

$$\begin{aligned} \rightarrow \text{Cyan} &= 1 - \text{Red} & \text{Red} &= 1 - \text{Cyan} \\ \rightarrow \text{Magenta} &= 1 - \text{Green} & \text{Green} &= 1 - \text{Magenta} \\ \rightarrow \text{Yellow} &= 1 - \text{Blue} & \text{Blue} &= 1 - \text{Yellow} \end{aligned}$$

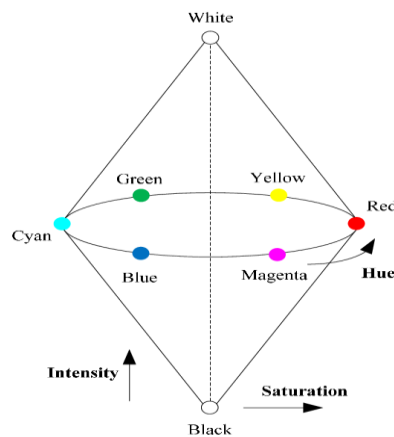


Εικόνα 3.11: Μοντέλο CMYK

(Πηγή <https://plumgroveinc.com/what-is-cmyk-color/>)

3.5.3 HSI

Το μοντέλο **HSI**, για τη λειτουργία του, κάνει χρήση της απόχρωσης (Hue), της καθαρότητας (Saturation) και της φωτεινότητας (Intensity). Η απόχρωση περιγράφει το χρώμα με τη μορφή γωνίας μεταξύ [0, 360] μοιρών. Η καθαρότητα περιγράφει πόσο το χρώμα αραιώνεται με λευκό φως. Το εύρος του S κυμαίνεται μεταξύ [0, 1]. Τέλος η φωτεινότητα λαμβάνει τιμές μεταξύ 0 (μαύρο) και 1 (λευκό).



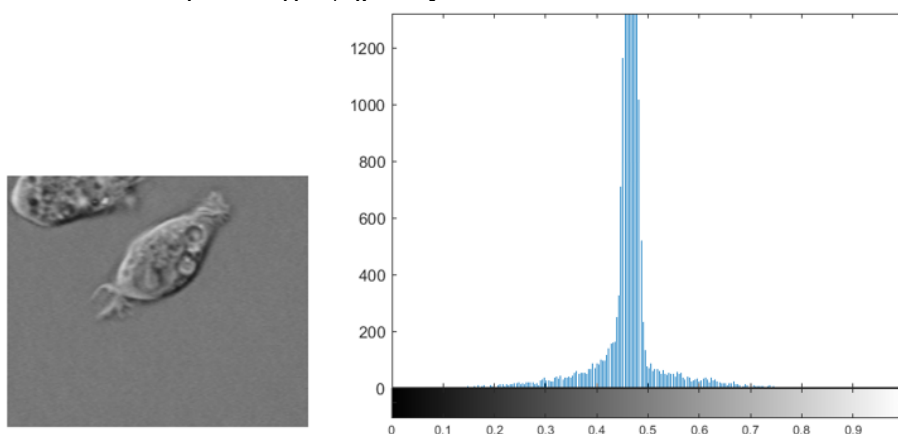
Εικόνα 3.12: Μοντέλο HIS

(Πηγή:https://www.researchgate.net/figure/HSI-color-space-change-accordingly-It-is-therefore-not-good-for-color-analysis-The-HSI_fig3_282544600)

3.6 Ιστόγραμμα

Το ιστόγραμμα εικόνας λειτουργεί ως γραφική αναπαράσταση της τονικής κατανομής σε μια ψηφιακή εικόνα. Αναπαριστά τον αριθμό των εικονοστοιχείων για κάθε τονική τιμή. Κοιτάζοντας το ιστόγραμμα για μια συγκεκριμένη εικόνα είμαστε σε θέση να κρίνουμε ολόκληρη την τονική κατανομή. Ο άξονας x του γραφήματος αντιπροσωπεύει τις τονικές παραλλαγές, ενώ ο άξονας y αντιπροσωπεύει το συνολικό αριθμό των εικονοστοιχείων στον συγκεκριμένο τόνο.

Η αριστερή πλευρά του οριζόντιου άξονα αντιπροσωπεύει τις σκοτεινές περιοχές, η μέση αντιπροσωπεύει τιμές μεσαίου τόνου και η δεξιά πλευρά αντιπροσωπεύει φωτεινές περιοχές. Ο κάθετος άξονας αντιπροσωπεύει το μέγεθος της περιοχής (συνολικός αριθμός εικονοστοιχείων) που καταγράφεται σε κάθε μία από αυτές τις ζώνες. Έτσι, το ιστόγραμμα για μια πολύ σκοτεινή εικόνα θα έχει τα περισσότερα από τα σημεία δεδομένων του στην αριστερή πλευρά και στο κέντρο του γραφήματος. Ενώ, το ιστόγραμμα για μια πολύ φωτεινή εικόνα με λίγες σκοτεινές περιοχές και/ή σκιές θα έχει τα περισσότερα από τα σημεία δεδομένων του στη δεξιά πλευρά και στο κέντρο του γραφήματος.



Εικόνα 3.3.13: Εικόνα κυττάρου και το αντίστοιχο ιστόγραμά της

4. Μετασχηματισμοί Εικόνας

Σε μία ψηφιακή εικόνα παρατηρείται σχέση αλληλεξάρτησης μεταξύ των φωτεινοτήτων των γειτονικών εικονοστοιχείων της, γεγονός που έχει ως απόρροια την μη αποτελεσματική κωδικοποίησή της, με άλλα λόγια υπάρχει περίπτωση να γίνεται χρήση μεγαλύτερου αριθμού bits για την αναπαράσταση τους από όσα θα απαιτούνταν κανονικά. Έναν βασικό λοιπόν σκοπό ενός μετασχηματισμού αποτελεί η απεικόνιση μίας ψηφιακής εικόνας με παράγοντες λιγότερο αλληλεξαρτημένους και να επιτευχθεί συνεπώς μια περισσότερο αποτελεσματική κωδικοποίηση.

4.1 Μετασχηματισμός Fourier

Ο μετασχηματισμός Fourier είναι μια αναπαράσταση μιας εικόνας ως άθροισμα πολύπλοκων εκθετικών μεταβλητών μεγεθών, συχνοτήτων και φάσεων. Ο μετασχηματισμός Fourier παίζει έναν κρίσιμο ρόλο σε ένα ευρύ φάσμα εφαρμογών επεξεργασίας εικόνας, συμπεριλαμβανομένης της βελτίωσης, ανάλυσης, αποκατάστασης και συμπίεσης.

Εάν $f(m, n)$ είναι συνάρτηση δύο χωριστών χωρικών μεταβλητών m και n , τότε ο δισδιάστατος μετασχηματισμός Fourier του $f(m, n)$ ορίζεται από τις σχέση:

$$F(\omega_1, \omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) e^{-j\omega_1 m} e^{-j\omega_2 n} \quad (4.1)$$

όπου οι μεταβλητές ω_1 και ω_2 αποτελούν κυκλικές συχνότητες και οι μονάδες τους είναι ακτίνια ανά δείγμα. Το $F(\omega_1, \omega_2)$ καλείται συχνά αναπαράσταση πεδίου συχνοτήτων του $f(m, n)$. Η συνάρτηση $F(\omega_1, \omega_2)$ είναι μια σύνθετη τιμή που είναι περιοδική τόσο στα ω_1 όσο και στα ω_2 , με περίοδο 2π .

Το αντίστροφο ενός μετασχηματισμού είναι μια λειτουργία που όταν εκτελείται σε μετασχηματισμένη εικόνα παράγει την αρχική εικόνα. Ο αντίστροφος δισδιάστατος μετασχηματισμός Fourier δίνεται από τη σχέση:

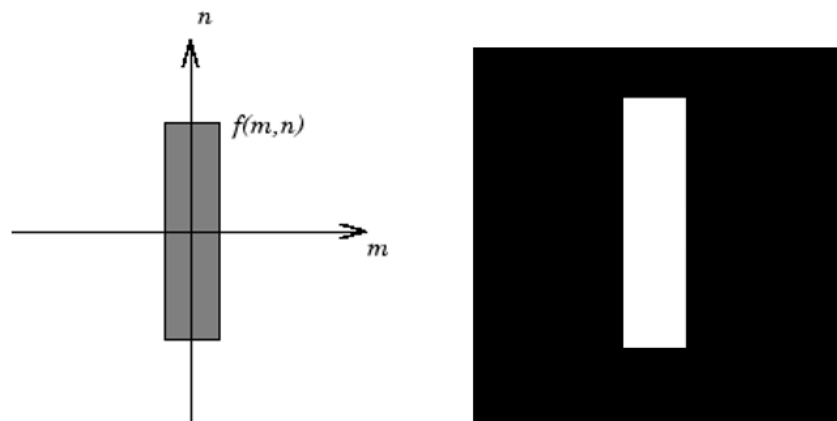
$$f(m, n) = \frac{1}{2\pi} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} F(\omega_1, \omega_2) e^{j\omega_1 m} e^{j\omega_2 n} \quad (4.2)$$

Ο διακριτός τρόπος αναπαράστασης στο πεδίο ορισμού της συχνότητας επιτυγχάνεται με την εφαρμογή του διακριτού μετασχηματισμού Fourier :

$$F(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j\frac{2\pi}{M}km} e^{-j\frac{2\pi}{N}ln} \quad (4.3)$$

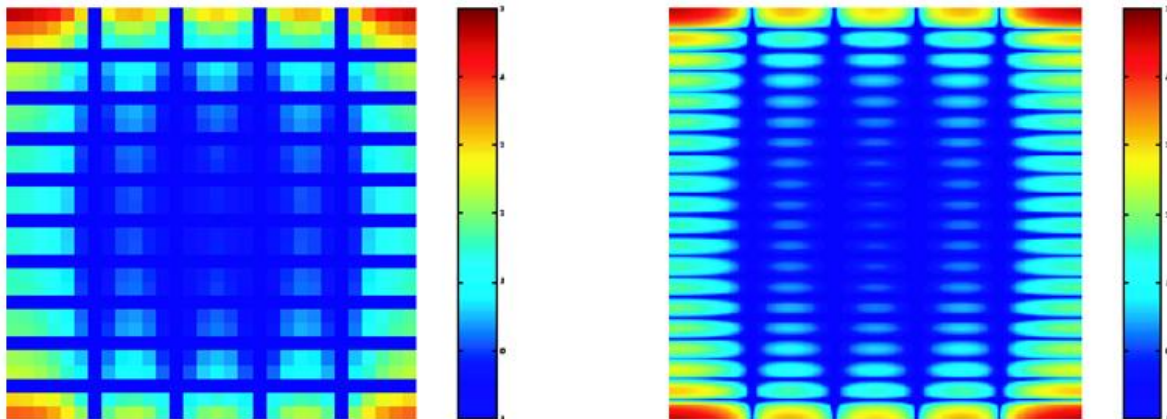
➤ Εφαρμογή απόκρισης με το μετασχηματισμό Fourier στο MATLAB :

- Θεωρούμε μια συνάρτηση $f(m, n)$ που ισούται με 1 σε μια ορθογώνια περιοχή και 0 οπουδήποτε αλλού και τη δημιουργούμε ως διακριτή εικόνα:



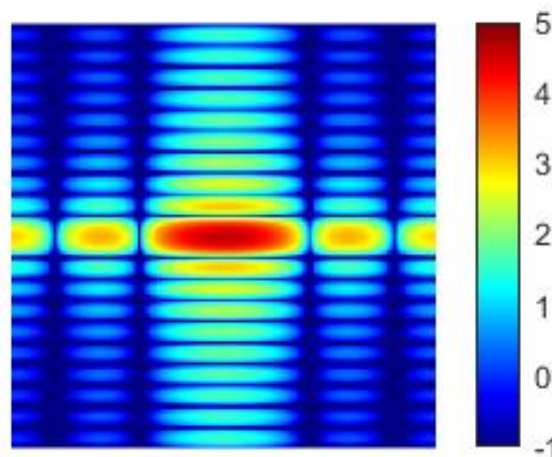
Εικόνα 4.4.1: Ορθογωνική συνάρτηση $f(m,n)$ (αριστερά) , διακριτή εικόνα (δεξιά)

- Υπολογίζεται και απεικονίζεται ο DFT της f , ενώ στη συνέχεια, για καλύτερη ανάλυση, συμπληρώνουμε την f με μηδενικά (zero padding):



Εικόνα 4.4.2: DFT της f χωρίς zero padding (αριστερά) , με zero padding (δεξιά)

- Εξαλείφουμε την DC συνιστώσα με ολίσηση αυτής και λαμβάνουμε το επιθυμητό αποτέλεσμα της ορθογωνικής συνάρτησης $f(m, n)$:



Εικόνα 4.4.3: DFT της ορθογωνικής $f(m,n)$

4.2 Μετασχηματισμός DCT

Ο μετασχηματισμός DCT (Discrete Cosine Transform) παρουσιάζει τα ακόλουθα χαρακτηριστικά:

- Είναι ορθογωνικός μετασχηματισμός
 - Η ενέργεια, όπου παρατηρείται αυξημένη συσχέτιση ανάμεσα στα κοντινά μεταξύ τους εικονοστοιχεία, μαζεύεται σε περιορισμένο αριθμό συντελεστών μετασχηματισμού, ενώ αποσυσχετίζει αρκετά αποτελεσματικά τους συντελεστές αυτούς.
 - Γίνεται χρήση του συχνά στη συμπίεση κατά JPEG και MPEG.
- Ο DCT περιγράφεται από την σχέση :

$$B(k, l) = \alpha_k \alpha_l \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A(m, n) \cos \frac{\pi(2m+1)k}{2M} \cos \frac{\pi(2n+1)l}{2N} \quad (4.4)$$

όπου το k παίρνει τιμές από 0 έως και $M-1$ και το l από 0 έως και $N-1$.

Πρακτικά ο όρος $A(m, n)$ αντιστοιχεί στην εικόνα που μετασχηματίζουμε, ενώ ο όρος $B(k, l)$ αντιστοιχεί στους συντελεστές μετασχηματισμού του A . Για την ανασύνθεση της A γίνεται χρήση του reverse DCT :

$$A(m, n) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \alpha_k \alpha_l B(k, l) \cos \frac{\pi(2m+1)k}{2M} \cos \frac{\pi(2n+1)l}{2N} \quad (4.5)$$

όπου το m παίρνει τιμές από 0 έως και $M-1$ και το n από 0 έως και $N-1$ και η $A(m, n)$ είναι ουσιαστικά το άθροισμα γινομένων των συντελεστών μετασχηματισμού.

➤ Παρακάτω θα δούμε την χρήση του DCT μετασχηματισμού σε συμπίεση JPEG η οποία υλοποιείται με βάση τα εξής βήματα:

- Χωρίζουμε την εικόνα σε 8x8 blocks ή 16x16 εικονοστοιχεία.
- Υπολογίζεται για κάθε block ο δισδιάστατος DCT και γίνεται κβάντιση, κωδικοποίηση και μετάδοση των συντελεστών μετασχηματισμού DCT.
- Ο αναγνώστης αρχείων JPEG αποκωδικοποιεί τους κβαντισμένους συντελεστές DCT, υπολογίζει τον αντίστροφο δισδιάστατο DCT κάθε μπλοκ και στη συνέχεια επανατοποθετεί τα μπλοκ μαζί σε μια ενιαία εικόνα. Για τυπικές εικόνες, πολλοί από τους συντελεστές DCT έχουν τιμές κοντά στο μηδέν. Αυτοί οι συντελεστές μπορούν να απορριφθούν χωρίς να επηρεαστεί σοβαρά η ποιότητα της ανακατασκευασμένης εικόνας.
- Αν και υπάρχει κάποια απώλεια ποιότητας στην ανακατασκευασμένη εικόνα, είναι σαφώς αναγνωρίσιμη, παρόλο που σχεδόν το 85% των συντελεστών DCT απορρίφθηκαν.



Εικόνα 4.4.4: Αρχική εικόνα και της ανακατασκευασμένη εικόνα με DCT

4.3 Μετασχηματισμός RADON

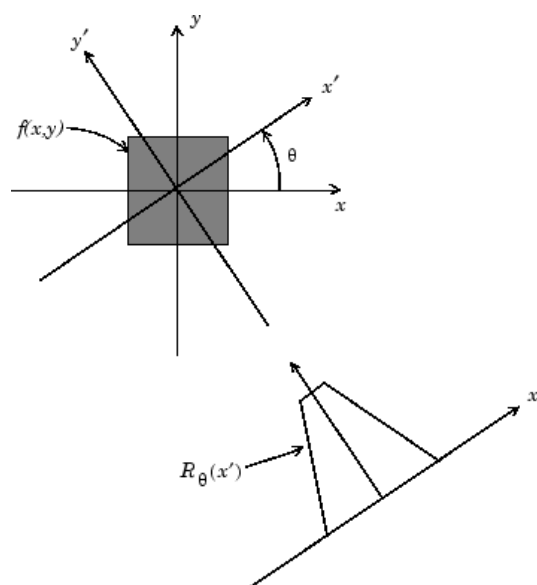
Ο μετασχηματισμός Radon περιγράφεται από την ακόλουθη σχέση:

$$R_{\theta}(x') = \int_{-\infty}^{\infty} f(x' \cos \theta - y' \sin \theta, x' \sin \theta + y' \cos \theta) dy' \quad (4.6)$$

όπου:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Ο μετασχηματισμός Radon μπορεί να θεωρηθεί ως η προβολή κάποιας συνάρτησης $f(x,y)$ (υπό τη μορφή εικόνας) στον άξονα x' , που με τη σειρά του συνθέτει μία γωνία θ με τον άξονα x , ενώ η ανασύσταση της γνήσιας εικόνας επιτυγχάνεται με τη χρήση του reverse Radon.



Εικόνα 4.4.5: Γεωμετρία Radon

(Πηγή : <https://uk.mathworks.com/help/images/radon-transform.html>)

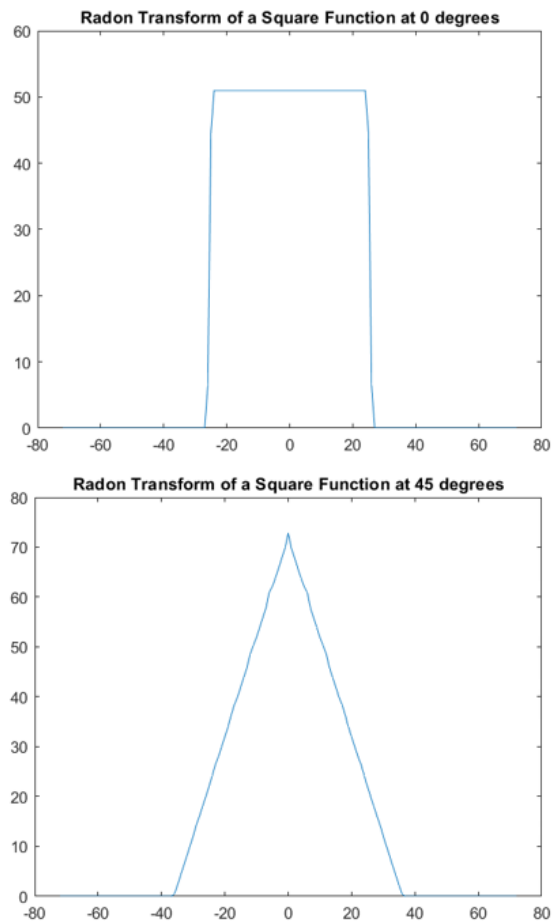
➤ Παρακάτω θα δούμε ένα παράδειγμα του μετασχηματισμού Radon:

- Δημιουργούμε μια εικόνα για αυτό το παράδειγμα που αποτελείται από ένα τετράγωνο αντικείμενο και την εμφανίζουμε .



Εικόνα 4. 4.6: Εικόνα για Radon μετασχηματισμό

- Υπολογίζουμε τον μετασχηματισμό Radon της εικόνας για τις γωνίες 0 μοίρες και 45 μοίρες.



Εικόνα 4.4.7: Αποτέλεσμα Radon μετασχηματισμο

5. Βελτιστοποίηση Εικόνας

Με τον όρο **βελτιστοποίηση** εικόνας (image enhancement) ουσιαστικά κάνουμε αναφορά σε τεχνικές επεξεργασίας μιας δεδομένης εικόνας με απώτερο σκοπό να γίνει η εικόνα αυτή όσο το δυνατό πιο κατάλληλη από την αρχική εικόνα για μια συγκεκριμένη εφαρμογή. Η διαδικασία βελτίωσης δεν αυξάνει το εγγενές περιεχόμενο πληροφοριών στα δεδομένα, αλλά αυξάνει το δυναμικό εύρος των επιλεγμένων χαρακτηριστικών, ώστε να μπορούν να ανιχνευθούν εύκολα. Οι τεχνικές αυτές περιλαμβάνουν τη μεταβολή της αντίθεσης (contrast), τη μεταβολή της φωτεινότητας (intensity), την καλυτέρευση του λόγου σήματος ως προς τον θόρυβο, την βελτίωση της ευκρίνειας και της οξύτητας της εικόνας κλπ. Είθισται οι τεχνικές βελτιστοποίησης των εικόνων να διακρίνονται σε:

- Μέθοδοι σημείου ή εικονοστοιχείου
- Μέθοδοι στο πεδίο του χώρου
- Μέθοδοι στο πεδίο της συχνότητας

5.1 Τεχνικές Σημείου

Οι μέθοδοι που συναντάμε στην κατηγορία αυτή δρουν στα εικονοστοιχεία των εκάστοτε εικόνων. Τα εικονοστοιχεία αυτά διαθέτουν 256 αποχρώσεις του γκριζου (χρήση Βάθους χρώματος 8 bit). Η δράση μίας τέτοιας τεχνικής μπορεί να περιγραφεί από τον ακόλουθο τύπο:

$$I'(x, y) = f([I(x, y)]) \quad (5.1)$$

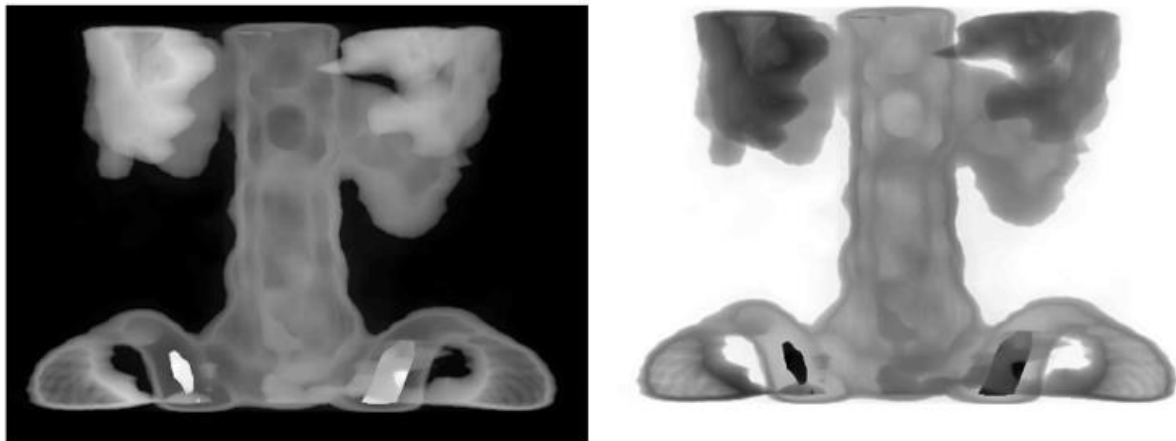
όπου ο όρος $I(x, y)$ αντιστοιχεί στην φωτεινότητα στη θέση (x, y) της εικόνας και λαμβάνει τιμές με εύρος από 0 έως και L , ο όρος f αντιστοιχεί στην συνάρτηση που δρα στη θέση αυτή και ο όρος $I'(x, y)$ αντιστοιχεί στην φωτεινότητα της θέσης (x, y) αφού έχει επέλθει η επεξεργασία της εικόνας και λαμβάνει τιμές όμοιες της

5.1.1 Αντιστροφή Φωτεινότητας

Η τεχνική της αντιστροφής φωτεινότητας (intensity inversion) υλοποιείται κατά τον ακόλουθο τύπο:

$$I'(x, y) = L - [I(x, y)] \quad (5.2)$$

Μέσω της αντιστροφής φωτεινότητας πρακτικά λαμβάνουμε το negative μιας εικόνας, γεγονός που καθιστά κάποια σημεία της εικόνας ευκολότερα αντιληπτά από το ανθρώπινο μάτι. Αμέσως πιο κάτω παρατίθεται ένα παράδειγμα της τεχνικής αυτής (το τμήμα του κώδικα που χρησιμοποιήθηκε παρατίθεται στα παραρτήματα της εργασίας) :

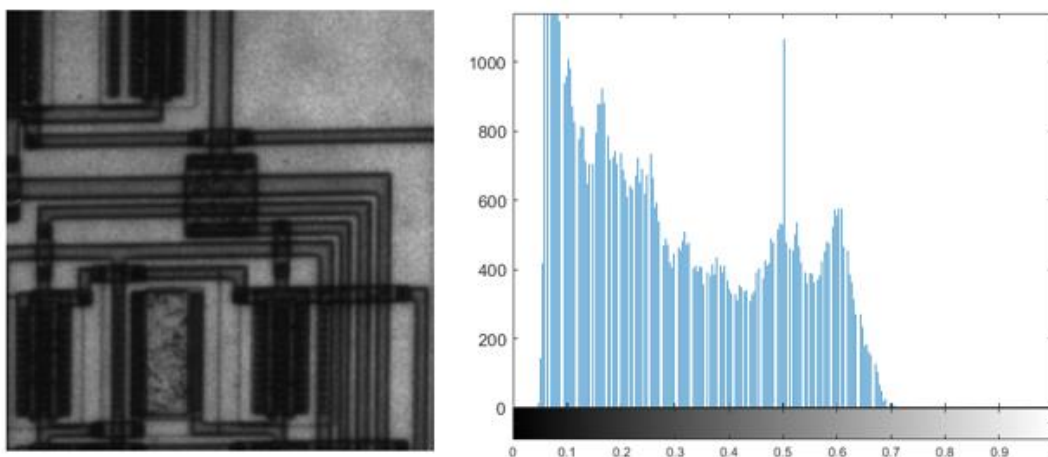


Εικόνα 5.5.1: Αποτέλεσμα Αντιστροφής φωτεινότητας

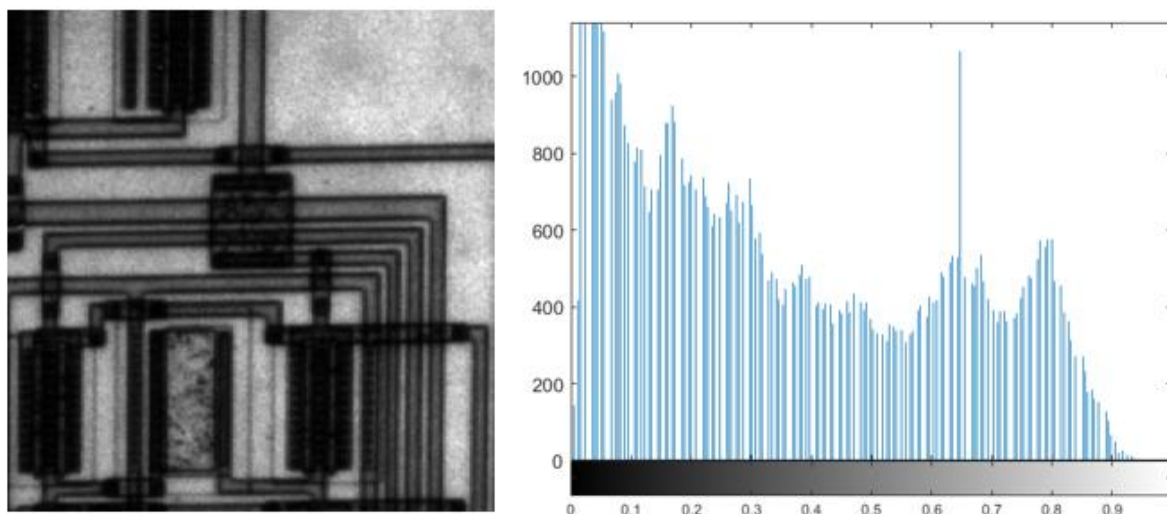
5.1.2 Επέκταση της Αντίθεσης

Λόγω των κακών συνθηκών κατά τις οποίες λαμβάνουμε συχνά κάποια εικόνα, αυτό που βλέπουμε δεν μας ικανοποιεί. Έτσι λοιπόν γίνεται χρήση της τεχνικής της επέκτασης της αντίθεσης η οποία είναι μια απλή μέθοδος βελτίωσης εικόνας που προσπαθεί να βελτιώσει την αντίθεση σε μια εικόνα επεκτείνοντας το εύρος τιμών της φωτεινότητας που περιέχει για να καλύψει ένα επιθυμητό εύρος τιμών, π.χ. το πλήρες εύρος τιμών pixel που επιτρέπει ο σχετικός τύπος εικόνας.

Στο παρακάτω παράδειγμα χρησιμοποιούμε μία grayscale εικόνα της οποίας η αντίθεση μπορεί να βελτιωθεί, καθώς και το ιστόγραμμα της. Θα παρατηρήσουμε τα περισσότερα από τα σημεία δεδομένων του στην αριστερή πλευρά και στο κέντρο του γραφήματος, καθώς η εικόνα μας είναι αρκετά σκοτεινή. Η επέκταση της αντίθεσης στην περίπτωση μας αυξάνει τη φωτεινότητά της, με αποτέλεσμα να παρουσιάζονται δεδομένα σχεδόν σε ολόκληρο το ιστόγραμμα.



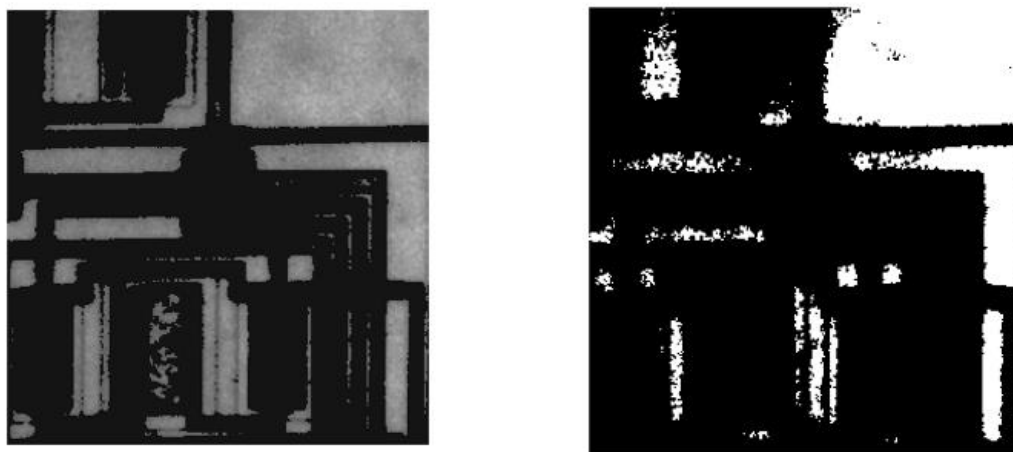
Εικόνα 5.5.2: Εικόνα πριν την επέκταση της αντίθεσης και το αντίστοιχο ιστόγραμμα



Εικόνα 5.5.3: Εικόνα μετά την επέκταση της αντίθεσης και το αντίστοιχο ιστόγραμμα

5.1.3 Κόψιμο Φωτεινότητας

Το κόψιμο φωτεινότητας σαν μέθοδος έχει κάποιες ομοιότητες με την επέκταση της αντίθεσης. Η περιοχή της εικόνας όπου έχει εφαρμοστεί η τεχνική αυτή θα εμφανίζεται συνήθως ως ομοιόμορφη περιοχή της ελάχιστης ή της μέγιστης φωτεινότητας, χάνοντας κάθε λεπτομέρεια της εικόνας, σε αντίθεση με την επέκταση της αντίθεσης, όπου η φωτεινότητα λαμβάνει πολλές διαφορετικές τιμές. Το κόψιμο της φωτεινότητας εξαρτάται από το threshold που ορίζουμε, με το οποίο επιλέγουμε να δώσουμε την ελάχιστη τιμή της φωτεινότητας για τιμές φωτεινότητας κάτω από το threshold και αντίστοιχα την μεγαλύτερη για τιμές μεγαλύτερες από το threshold.



Εικόνα 5.5.4: Εικόνα 'circuit.tif' πριν (αριστερά) και μετά (δεξιά) το κόψιμο φωτεινότητας

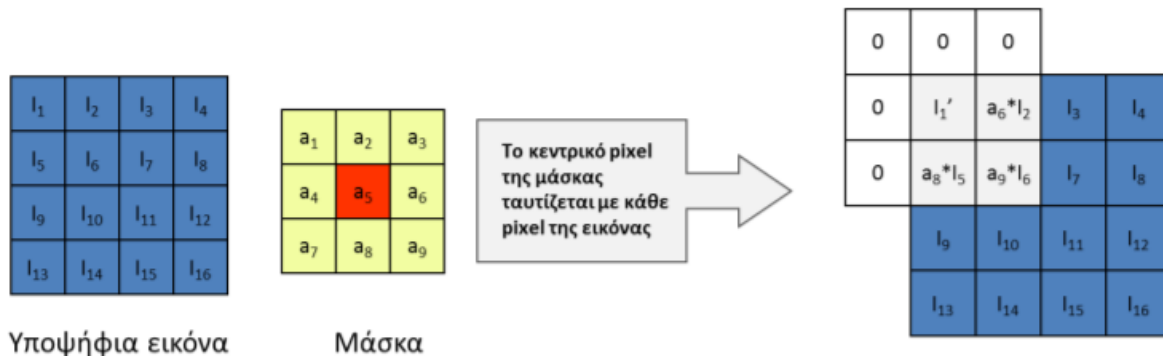
5.2 Τεχνικές Χώρου

Οι μέθοδοι που συναντάμε στην κατηγορία αυτή τροποποιούν την τιμή κάθε εικονοστοιχείου με βάση ενός κώδικα που δρα στα γειτονικά εικονοστοιχεία του εξεταζόμενου εικονοστοιχείου. Η τροποποίηση της τιμή των εικονοστοιχείων γίνεται με τη χρήση μίας μάσκας, της οποίας ο πυρήνας εφαρμόζει πάνω στο εικονοστοιχείο που μας ενδιαφέρει και

τροποποιεί την τιμή του εικονοστοιχείου αυτού με βάση τον κώδικα που χρησιμοποιούμε. Η τροποποίηση αυτή περιγράφεται από την παρακάτω σχέση:

$$I'(x, y) = \sum_{m=-r}^r \sum_{n=-c}^c w(m, n) I(x + m, y + n) \quad (5.3)$$

όπου $m \times n$ το μέγεθος της μάσκας, $I(x, y)$ η εικόνα στην οποία θέλουμε να τοποθετήσουμε τη μάσκα και $I'(x, y)$ η εικόνα αφού έχει τοποθετηθεί η μάσκα.



Υποψήφια εικόνα

Μάσκα

Εικόνα 5.5.5: Τοποθέτηση μάσκας για βελτίωση εικόνας στο πεδίο του χώρου

Εφόσον όλοι οι συντελεστές της μάσκας έχουν την τιμή 1, η τιμή του I_1' θα προκύπτει από το μέσο όρο των τιμών των εικονοστοιχείων που περικλύζουν το αρχικό εικονοστοιχείο της αρχικής εικόνας. Κατά τη διεργασία αυτή, συχνά υπάρχει πρόβλημα με τα pixels που ευρίσκονται στα όρια της εικόνας. Μία περίπτωση είναι τα pixels που δεν υπάρχουν να θεωρούνται μηδενικής τιμής (zero padding). Αυτό προφανώς δημιουργεί παραμόρφωση στα όρια της εικόνας.

5.2.1 Laplacian filter

Μία διαδεδομένη τεχνική βελτίωσης μίας εικόνας στο πεδίο του χώρου, αποτελεί η χρήση του Laplacian φίλτρου, που περιγράφεται από τη σχέση:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (5.4)$$

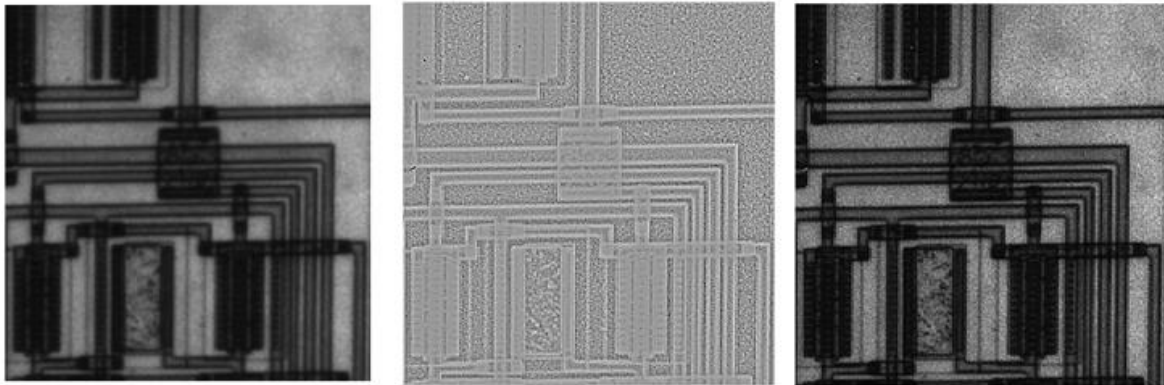
,όπου

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad (5.5)$$

Άρα προκύπτει:

$$\nabla^2 f = f(x + 1, y) + f(x - 1, y) - 4f(x, y) + f(x, y + 1) + f(x, y - 1) \quad (5.6)$$

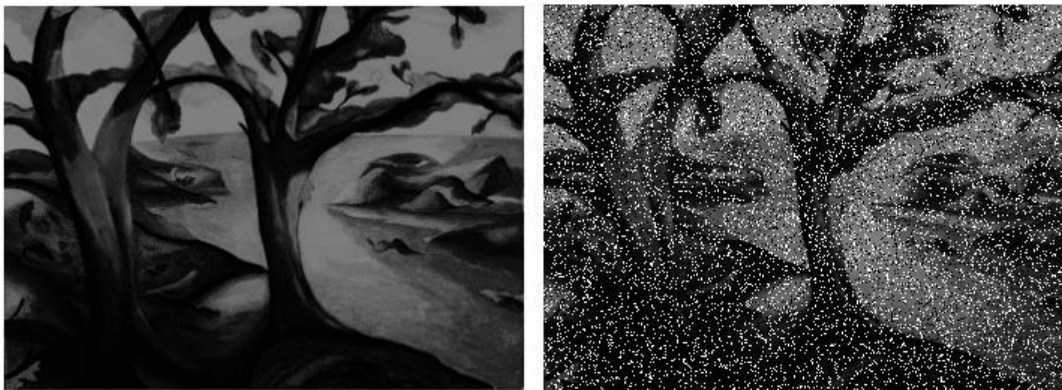


Εικόνα 5.5.6: Τονισμένες ακμές , αποτέλεσμα εφαρμογής Laplacian filter

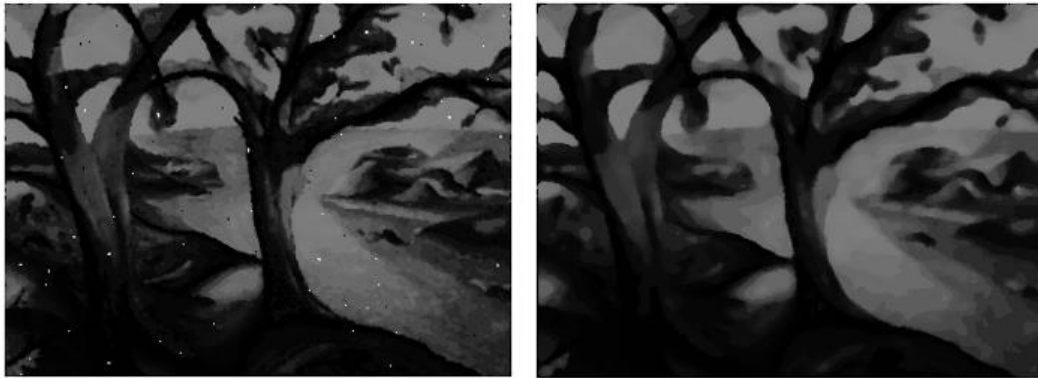
Βλέπουμε ότι το φίλτρο αυτό εντοπίζει τα σημεία όπου η φωτεινότητα αλλάζει δραστικά και καθιστά την εικόνα σημαντικά πιο ευδιάκριτη , δίνοντας έμφαση στις ακμές.

5.2.2 Φίλτρο Μεσαίας Τιμής

Το **φίλτρο μεσαίας τιμής** (median filter) είναι μια μη γραμμική τεχνική ψηφιακού φιλτραρίσματος, που χρησιμοποιείται συχνά για την εξάλειψη του θορύβου, που κυρίως παρουσιάζεται υπο τη μορφή λευκών ή μαύρων κηλίδων, του λεγόμενου θορύβου μορφής «salt and pepper», από μια εικόνα ή ένα σήμα. Το median filter θέτει στο εκάστοτε εικονοστοιχείο που εξετάζεται την μέση τιμή των εικονοστοιχείων με τα οποία βρίσκεται σε επαφή. Ακολουθούν παρακάτω δύο παραδείγματα με χρήση διαφορετικού μεγέθους μάσκα στην κάθε περίπτωση:



Εικόνα 5.5.7: Εικόνα 'trees.tif' με πυκνότητα θορύβου 0,2



Εικόνα 5.5.8: Χρήση median filter μάσκας 3x3 και χρήση median filter μάσκας 7x7

Παρατηρούμε πως, ενώ και στις δύο περιπτώσεις εφαρμόζεται median filter με zero padding και ίδιας πυκνότητας θόρυβος, στην πρώτη περίπτωση όπου χρησιμοποιείται μικρότερων διαστάσεων μάσκα (3x3), δεν έχουμε απόλυτη εξάλειψη του θορύβου. Αντιθέτως, με τη χρήση μεγαλύτερων διαστάσεων μάσκα (7x7) γίνεται απόλυτη απαλοιφή του θορύβου, θυσιάζοντας παρόλα αυτά αρκετές λεπτομέρειες της εικόνας, κάτι που δεν παρατηρείται στην πρώτη περίπτωση.

6. Αποκατάσταση Εικόνας

Όπως και στη βελτίωση της εικόνας, ο τελικός στόχος των τεχνικών **αποκατάστασης** είναι να βελτιωθεί μια δεδομένη εικόνα κατά μία έννοια. Για λόγους διαφοροποίησης, θεωρούμε ότι η αποκατάσταση είναι μια διαδικασία που επιχειρεί να ανασυγκροτήσει ή να ανακτήσει μια εικόνα που έχει υποβαθμιστεί χρησιμοποιώντας κάποια εκ των προτέρων γνώση του φαινομένου της υποβάθμισης. Έτσι, οι τεχνικές αποκατάστασης προσανατολίζονται προς τη μοντελοποίηση της υποβάθμισης και την εφαρμογή της αντίστροφης διαδικασίας για την ανάκτηση της αρχικής εικόνας. Οι τεχνικές βελτίωσης, από την άλλη πλευρά, είναι βασικά χειριστικές διαδικασίες που έχουν σχεδιαστεί για να χειριστούν μια εικόνα προκειμένου να επωφεληθούν από τις ψυχοφυσικές πτυχές του ανθρώπινου οπτικού συστήματος. Για παράδειγμα, η επέκταση αντίθεσης θεωρείται μια τεχνική βελτίωσης επειδή βασίζεται κυρίως στις ευχάριστες πτυχές που μπορεί να παρουσιάσει στον θεατή, ενώ η αφαίρεση της θόλωσης της εικόνας με την εφαρμογή μιας κατάλληλης λειτουργίας θεωρείται πρόβλημα αποκατάστασης.

Οι πρώτες τεχνικές για την αποκατάσταση της ψηφιακής εικόνας προήλθαν κυρίως από έννοιες πεδίου συχνότητας. Η έμφαση πλέον επικεντρώνεται εδώ, ωστόσο, σε μια πιο σύγχρονη, αλγεβρική προσέγγιση του προβλήματος. Αυτή η προσέγγιση έχει το πλεονέκτημα ότι επιτρέπει την εξαγωγή πολυάριθμων τεχνικών αποκατάστασης ξεκινώντας από τις ίδιες βασικές αρχές. Αν και μια άμεση λύση με τη χρήση των αλγεβρικών μεθόδων περιλαμβάνει γενικά τον χειρισμό μεγάλων συστημάτων ταυτόχρονων εξισώσεων, θα παρατηρήσουμε ότι καθίσταται, υπό ορισμένες συνθήκες, δυνατόν να μειωθεί η πολυπλοκότητα του υπολογισμού. Η παρουσία του θορύβου σε μία εικόνα συνδέεται άμεσα με τον τρόπο κατά τον οποίο η εκάστοτε εικόνα μεταδίδεται ή ψηφιοποιείται.

6.1 Μορφές θορύβου

Ο θόρυβος που παρουσιάζεται στις ψηφιακές εικόνες μπορεί να περιγραφεί ως οι ανεπιθύμητες πληροφορίες στις ψηφιακές αυτές εικόνες. Ο θόρυβος παράγει ανεπιθύμητα εφέ όπως τεχνουργήματα, μη ρεαλιστικά άκρα, αόρατες γραμμές, γωνίες, θολά αντικείμενα και παρενοχλεί σκηνές φόντου. Με σκοπό τον περιορισμό των ανεπιθύμητων αυτών ενεργειών, η εκ των προτέρων εκμάθηση των μοντέλων θορύβου καθίσταται αναγκαία με σκοπό την περαιτέρω επεξεργασία. Ο ψηφιακός θόρυβος είναι πιθανό να οφείλει την ύπαρξή του σε ποικίλα είδη πηγών, όπως αισθητήρες συζευγμένης φόρτισης (CCD) και αισθητήρες συμπληρωματικού ημιαγωγού οξειδίου μετάλλου (CMOS). Η λειτουργία της πυκνότητας πιθανότητας (PDF) ή του ιστογράμματος εΐθισται να αξιοποιείται για το σχεδιασμό και τον χαρακτηρισμό των μορφών θορύβου. Στη συνέχεια θα γίνει μια σύντομη αναφορά σε κάποιες μορφές θορύβου που συναντώνται συχνά.

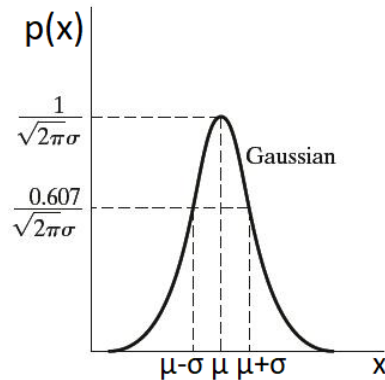
6.1.1 Gaussian ή κανονικός θόρυβος

Μπορεί να γίνει αναφορά στον κανονικό θόρυβο επίσης και ως ηλεκτρονικός θόρυβος επειδή εμφανίζεται σε ενισχυτές ή ανιχνευτές. Η μορφή του θορύβου αυτού προέρχεται κυρίως από φυσικές πηγές όπως η θερμική δόνηση των ατόμων και η διακριτή φύση της ακτινοβολίας των θερμών αντικειμένων. Ο Gaussian θόρυβος γενικά διαταράσσει τις τιμές του γκρι στις ψηφιακές εικόνες. Γενικά το μαθηματικό μοντέλο θορύβου Gauss αντιπροσωπεύει τη σωστή

προσέγγιση του πραγματικού κόσμου. Η κατανομή του θορύβου τέτοιας μορφής περιγράφεται από τον ακόλουθο τύπο:

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-m)^2/2\sigma^2} \quad (6.1)$$

,όπου μ η μέση τιμή, σ η τυπική απόκλιση και σ^2 η διασπορά.



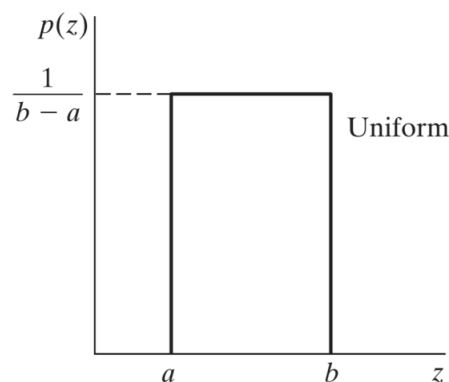
Εικόνα 6.6.1: Κατανομή Gaussian noise

6.1.2 Ομοιόμορφος θόρυβος (Uniform noise)

Η εμφάνιση της μορφής του θορύβου αυτού είναι εγγενής στη διαδικασία κβαντισμού εύρους. Γενικότερα παρουσιάζεται λόγω αναλογικών δεδομένων που μετατρέπονται σε ψηφιακά δεδομένα. Σε αυτό το μοντέλο θορύβου, ο λόγος του σήματος ως προς τον θόρυβο (SNR) περιορίζεται από την ελάχιστη και τη μέγιστη τιμή εικονοστοιχείου. Ισχύει ότι:

$$P(Z) = \frac{1}{b-a} \text{ για τιμές του } Z \text{ από } a \text{ έως } b$$
$$P(Z) = 0 \text{ για τις υπόλοιπες τιμές του } Z \quad (6.2)$$

$$\mu = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12} \quad (6.3)$$



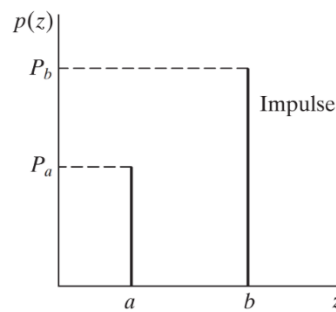
Εικόνα 6.6.2: Κατανομή uniform noise

6.1.3 Σημειακός θόρυβος (Salt and Pepper noise)

Ο σημειακός ή «salt and pepper» θόρυβος ονομάζεται επίσης θόρυβος πτώσης δεδομένων επειδή στατιστικά ρίχνει τις αρχικές τιμές δεδομένων. Ωστόσο, η εικόνα δεν καταστρέφεται ολοκληρωτικά από το θόρυβο της μορφής αυτή, αλλά μεταβάλλονται μεμονωμένα οι τιμές του εκάστοτε εικονοστοιχείου της ψηφιακής εικόνας. Ο θόρυβος αλατοπίπερου παρατηρείται κυρίως στη μετάδοση δεδομένων. Οι τιμές των εικονοστοιχείων της ψηφιακής εικόνας

αντικαθίστανται από τιμές «κατεστραμμένων» εικονοστοιχείων είτε μέγιστης είτε ελάχιστης τιμής εικονοστοιχείου, για παράδειγμα με 255 ή 0 αντίστοιχα, εάν ο αριθμός των bit που είναι διαθέσιμα για τη μετάδοση είναι 8. Για την κατανομή του συγκεκριμένου θορύβου ισχύει ότι:

$$P(z) = \begin{cases} Pa \text{ για } z = a \\ Pb \text{ για } z = b \\ 0 \text{ για άλλες τιμές του } z \end{cases} \quad (6.4)$$



Εικόνα 6.6.3: Κατανομή impulse noise

6.2 Φίλτρα Εξάλειψης Θορύβου

Όπως έχουμε αναφέρει και παραπάνω, οι ψηφιακές εικόνες μπορούν να αλλοιωθούν από μια πληθώρα μορφών θορύβου, του κυριότερους από τους οποίους έχουμε ήδη αναλύσει προηγουμένως. Ειδικότερα οι ψηφιακές εικόνες των οποίων η προέλευση είναι από κάποια ιατρικό εξοπλισμό. Οι διάφοροι τύποι θορύβου εμφανίζονται σε μια ψηφιακή εικόνα με ποικίλες μορφές, σύμφωνα με τον τρόπο κατά τον οποίο έχουν αυτές κατασκευαστεί. Για να επιτευχθεί η εξάλειψη του θορύβου που εμφανίζεται σε μια εικόνα χρησιμοποιούμε ποικίλα φίλτρα, σύμφωνα με το είδος του θορύβου. Με τον όρο φιλτράρισμα κάνουμε αναφορά σε κάποιες μεθόδους που καθιστούν δυνατή την τροποποίηση ή προβολή κάποιας ψηφιακής εικόνας. Παραδείγματος χάρη, υπάρχει η δυνατότητα να εφαρμοστεί φίλτρο σε κάποια εικόνα με απώτερο σκοπό την έμφαση διαφόρων χαρακτηριστικών ή την εξάλειψή τους. Κατά την εφαρμογή των φίλτρων γίνεται επέμβαση στα γειτονικά εικονοστοιχεία, όπου η τιμή κάθε εικονοστοιχείου της ψηφιακής εικόνας στο οποίο έχει δράσει το επιλεγμένο φίλτρο ορίζεται με την εφαρμογή κάποιου κώδικα στις τιμές των γειτονικών εικονοστοιχείων που ανήκουν στην εικόνα αυτή σε αυτή εφαρμοστεί το επιλεγμένο φίλτρο.

6.2.1 Φίλτρο μέσης τιμής (Arithmetic Mean Filter)

Το φίλτρο αυτό, στο οποίο έχει γίνει ήδη μία σύντομη αναφορά όταν εφαρμοστεί σε μια ψηφιακή εικόνα επιτυγχάνει την απαλοιφή του θορύβου, όπως ο ομοιόμορφος (uniform) και θόρυβος τύπου Gaussian από την εικόνα, θυσιάζοντας όμως κάποια χαρακτηριστικά της ψηφιακής εικόνας τα οποία καθιστά λιγότερο ευδιάκριτα λόγω της δημιουργίας θόλωσης της. Το φίλτρο μέσης τιμής μπορεί να προσδιοριστεί ως ο μέσος όρος όλων των εικονοστοιχείων σε ένα τμήμα μιας ψηφιακής εικόνας. Τα pixel που περιλαμβάνονται στη λειτουργία μέσου όρου καθορίζονται από μια μάσκα. Όσο μεγαλύτερη είναι η μάσκα φιλτραρίσματος τόσο πιο κυρίαρχη γίνεται η θόλωση και τόσο περισσότερο λιγοστεύουν οι λεπτομέρειες οι οποίες γίνονται αντιληπτές στην εικόνα. Η σχέση που περιγράφει το φίλτρο αριθμητικής μέσης τιμής είναι η ακόλουθη :

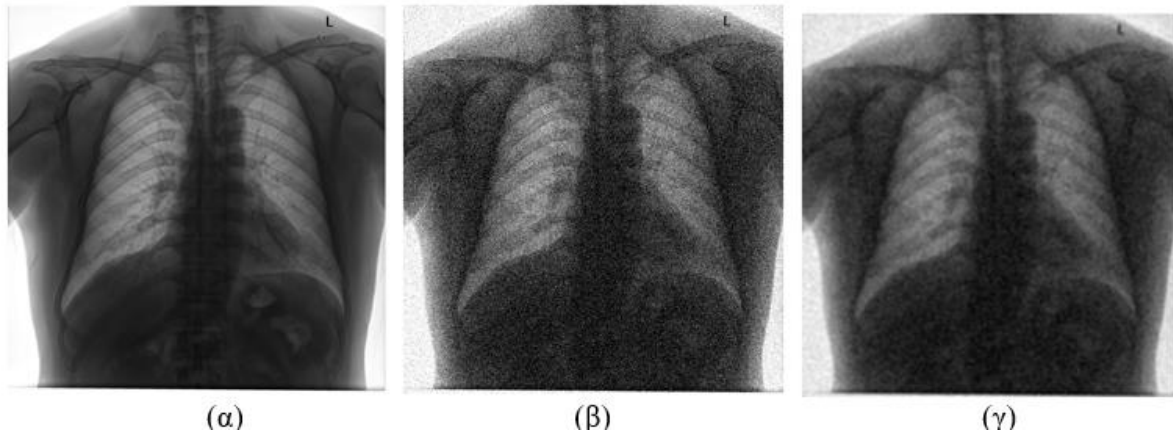
$$\hat{f}(x, y) = \frac{1}{mn} \sum_{Axy} g(x, y) \quad (6.5)$$

,όπου $g(x, y) \rightarrow$ θορυβώδης εικόνα

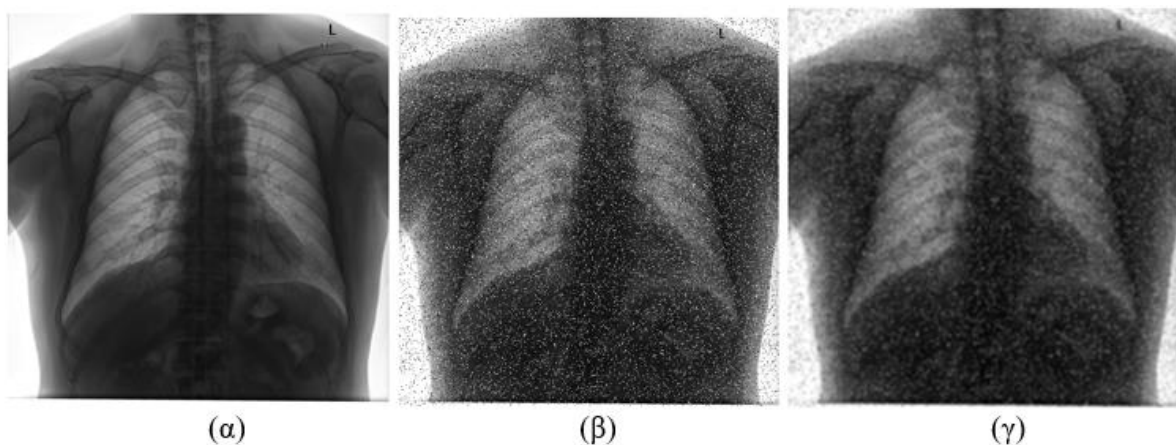
$\hat{f}(x, y) \rightarrow$ Αποτέλεσμα εφαρμογής του φίλτρου

$Axy \rightarrow$ Μάσκα μεγέθους $m \times n$

Παρακάτω παρατίθενται δύο παραδείγματα εφαρμογής του φίλτρου αριθμητικής μέσης τιμής. Στην πρώτη περίπτωση γίνεται εισαγωγή κανονικού θορύβου με τυπική απόκλιση ίση με 20, επιθυμώντας μέτρια ένταση θορύβου, ενώ στη δεύτερη προσθέτουμε σημειακό θόρυβο, με πυκνότητα θορύβου ίση με 0,1. Και στις δύο περιπτώσεις γίνεται χρήση μάσκας 5×5 για να έχουμε μέτρια θόλωση της εικόνας μετά την εφαρμογή του φίλτρου.



Εικόνα 6.6.4: (α) Chest MRI, (β) Chest MRI με κανονικό θόρυβο, (γ) Average filter



Εικόνα 6.6.5: (α) Chest MRI, (β) Chest MRI με σημειακό θόρυβο, (γ) Average filter

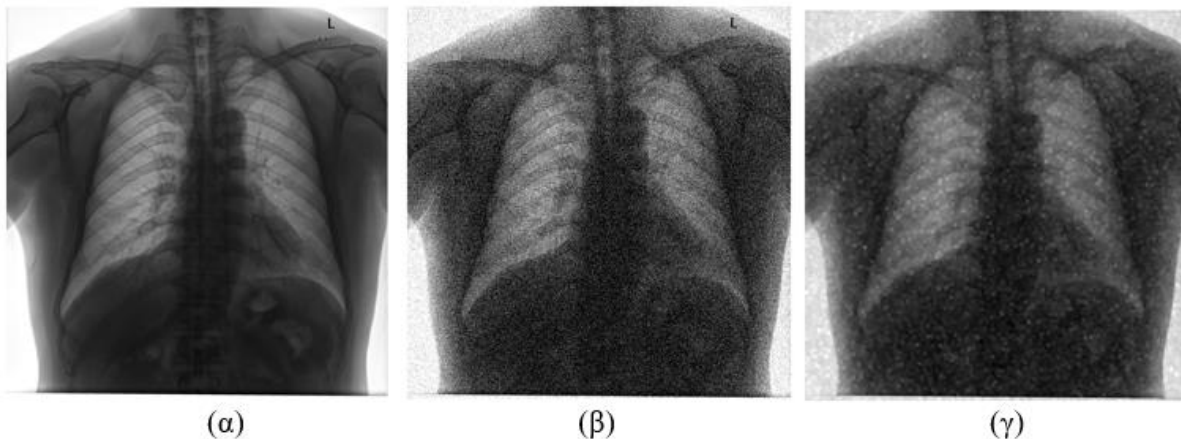
Παρατηρούμε ότι το φίλτρο αυτό είναι λιγότερο αποτελεσματικό όσον αφορά τη διατήρηση των λεπτομερειών στην τελική εικόνα σε σχέση με το median φίλτρο το οποίο ενεργεί επίσης στο χωρικό πεδίο και εξετάστηκε στις τεχνικές βελτίωσης της ψηφιακής εικόνας.

6.2.2 Φίλτρο μέγιστης τιμής (Maximum Filter)

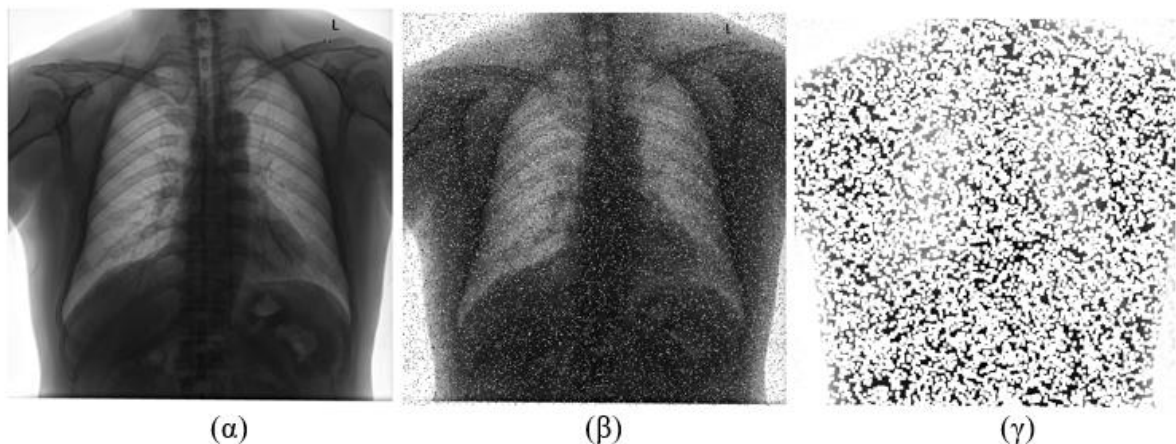
Με την εφαρμογή του φίλτρου μέγιστης τιμής ορίζεται ως το μέγιστο όλων των εικονοστοιχείων σε μια τοπική περιοχή μιας εικόνας. Εδώ δηλαδή οποιοδήποτε εικονοστοιχείο της θορυβώδους περιοχής λαμβάνει την «maximum» τιμή των εικονοστοιχείων που το περικλύζουν. Το φίλτρο αυτό ορίζεται με τη βοήθεια του παρακάτω τύπου:

$$\hat{f}(x, y) = \max g(x, y) \quad (6.6)$$

Παρακάτω παρατίθενται δύο παραδείγματα εφαρμογής του φίλτρου μέγιστης τιμής. Στην πρώτη περίπτωση γίνεται εισαγωγή κανονικού θορύβου με τυπική απόκλιση ίση με 20 , επιθυμώντας μέτρια ένταση θορύβου , ενώ στη δεύτερη προσθέτουμε σημειακό θόρυβο , με πυκνότητα θορύβου ίση με 0,1. Και στις δύο περιπτώσεις γίνεται χρήση μάσκας 5 x 5 για να έχουμε μέτρια θόλωση της εικόνας μετά την εφαρμογή του φίλτρου.



Εικόνα 6.6.6: (α) Chest MRI, (β) Chest MRI με κανονικό θόρυβο, (γ)Max filter



Εικόνα 6.6.7: (α) Chest MRI, (β) Chest MRI με σημειακό θόρυβο, (γ)Max filter

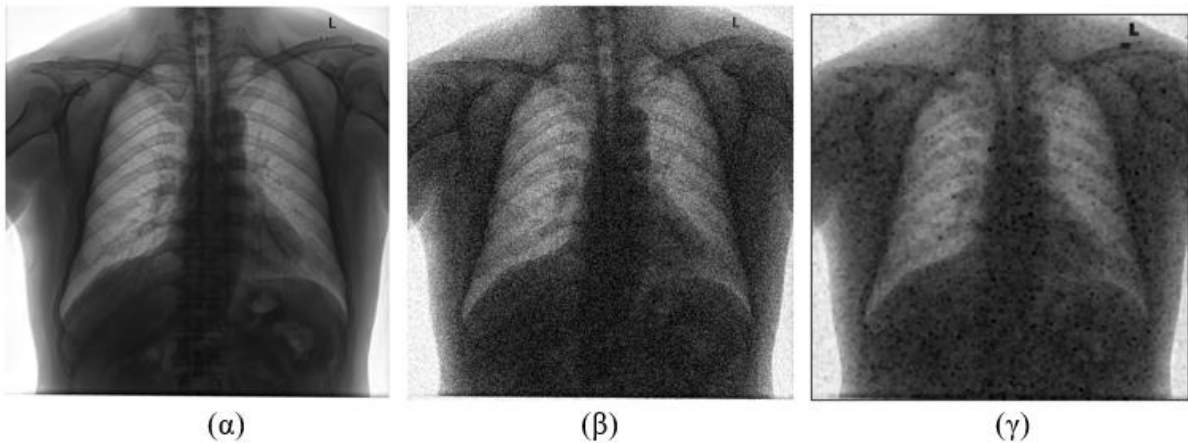
Είναι φανερό ότι το maximum filter προσφέρει λιγότερο ικανοποιητικό αποτέλεσμα σε σχέση με το φίλτρο αριθμητικής μέσης τιμής.

6.2.2 Φίλτρο ελάχιστης τιμής (Minimum Filter)

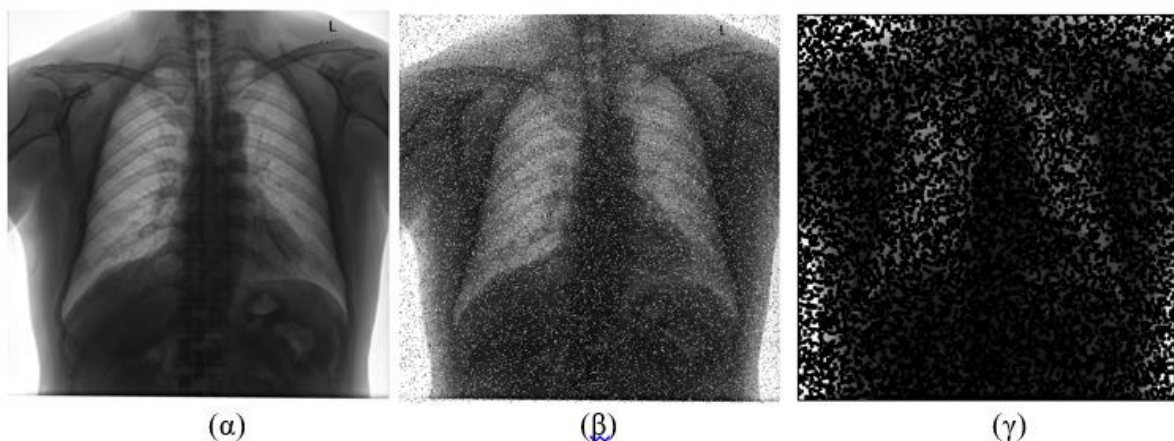
Με την εφαρμογή του φίλτρου ελάχιστης τιμής ορίζεται ως το ελάχιστο όλων των εικονοστοιχείων σε μια τοπική περιοχή μιας εικόνας. Εδώ δηλαδή οποιοδήποτε εικονοστοιχείο της θορυβώδους περιοχής λαμβάνει την «minimum» τιμή των εικονοστοιχείων που το περικλύζουν. Το φίλτρο αυτό ορίζεται με τη βοήθεια του παρακάτω τύπου:

$$\hat{f}(x, y) = \min g(x, y) \quad (6.7)$$

Παρακάτω παρατίθενται δύο παραδείγματα εφαρμογής του φίλτρου ελάχιστης τιμής. Στην πρώτη περίπτωση γίνεται εισαγωγή κανονικού θορύβου με τυπική απόκλιση ίση με 20 , επιθυμώντας μέτρια ένταση θορύβου , ενώ στη δεύτερη προσθέτουμε σημειακό θόρυβο , με πυκνότητα θορύβου ίση με 0,1. Και στις δύο περιπτώσεις γίνεται χρήση μάσκας 5 x 5 για να έχουμε μέτρια θόλωση της εικόνας μετά την εφαρμογή του φίλτρου.



Εικόνα 6.6.8: (α) Chest MRI, (β) Chest MRI με κανονικό θόρυβο, (γ)Min filter



Εικόνα 6.6.9: (α) Chest MRI, (β) Chest MRI με σημειακό θόρυβο, (γ)Min filter

Παρατηρούμε πως το minimum filter προσφέρει και αυτό λιγότερο ικανοποιητικό αποτέλεσμα σε σχέση με το φίλτρο αριθμητικής μέσης τιμής επίσης , αλλά εδώ όπως είναι λογικό , στο

Ψηφιακή επεξεργασία και κατάτμηση βιοϊατρικών εικόνων με χρήση τεχνητών νευρωνικών δικτύων

αποτέλεσμα εμφανίζονται μαύρα τα εικονοστοιχεία στα οποία είχαμε προσθέσει θόρυβο , σε αντίθεση με το maximum filter , όπου εμφανίζονται λευκά.

7. Κατάτμηση Εικόνας

Σε πολλές περιπτώσεις στις καθημερινές εργασίες μας, αλλά κυρίως σε πιο εξειδικευμένες εργασίες που αφορούν ποικίλους κλάδους της επιστήμης, έχει εμφανιστεί η ανάγκη να προσδώσει στους υπολογιστές τη δυνατότητα να καταλήγουν σε διάφορα πορίσματα και να συλλέγουν δεδομένα μέσα από μία πληθώρα ψηφιακών εικόνων, δρώντας σαν να διαθέτουν την κριτική σκέψη του ανθρώπου και γενικά τον νου του. Η απαίτηση συλλογής των δεδομένων προκάλεσε την εμφάνιση της διαδικασίας της κατάτμησης της ψηφιακής εικόνας.

Η κατάτμηση της ψηφιακής εικόνας είναι ένας κλάδος της ψηφιακής επεξεργασίας εικόνας που επικεντρώνεται στη διαίρεση μιας εικόνας σε διαφορετικά μέρη ανάλογα με τα χαρακτηριστικά και τις ιδιότητές τους. Ο κυριότερος σκοπός της κατάτμησης εικόνας είναι η απλοποίηση των διαφόρων χαρακτηριστικών της με απώτερο στόχο να τα καταστούν αυτά ακόμα πιο ευδιάκριτα για την περαιτέρω ανάλυση τους. Κατά τη διαδικασία της κατάτμησης της ψηφιακής εικόνας, χωρίζεται μια εικόνα σε ξεχωριστά τμήματα τα οποία παρουσιάζουν διάφορα κοινά χαρακτηριστικά. Τα μέρη στα οποία τμηματοποιείται η ψηφιακή εικόνα ονομάζονται αντικείμενα εικόνας.

Χρησιμοποιώντας τεχνικές κατάτμησης εικόνας, μας δίνεται η δυνατότητα να χωρίσουμε και να ομαδοποιήσετε εικονοστοιχεία από μια εικόνα, να τα αντιστοιχίσουμε σε ποικίλες επιθυμητές ετικέτες και να ταξινομήσουμε στη συνέχεια τα εικονοστοιχεία αυτά σύμφωνα με αυτές τις ετικέτες. Έχουμε την ευχέρεια να σχεδιάσουμε γραμμές, να ορίσουμε περιγράμματα και να διαχωρίσουμε συγκεκριμένα αντικείμενα (σημαντικά στοιχεία) σε μια εικόνα από τα υπόλοιπα αντικείμενα (ασήμαντα στοιχεία).

Στον πιο ειδικό τομέα της μηχανικής μάθησης, είναι δυνατό να γίνει χρήση αυτών των ετικετών που έχουν προκύψει από την κατάτμηση εικόνας για εποπτευόμενη και χωρίς επίβλεψη εκπαίδευση. Αυτό θα σας επιτρέψει να προσφέρουμε λύσεις σε διάφορα προβλήματα. Η κατάτμηση των ψηφιακών εικόνων έχει πολλές εφαρμογές σε πολυάριθμους κλάδους. Μερικές από τις αξιοσημείωτες περιοχές όπου χρησιμοποιείται κατά πολύ η κατάτμηση των ψηφιακών εικόνων είναι οι ακόλουθες:

➤ Αναγνώριση προσώπου

Η τεχνολογία αναγνώρισης προσώπου που συναντάμε στα κινητά μας τηλέφωνα και στα προηγμένα συστήματα ασφαλείας εκμεταλλεύεται τη χρήση της κατάτμησης της ψηφιακής εικόνας για την αναγνώριση του προσώπου μας. Έτσι λοιπόν, είναι δυνατός ο προσδιορισμός των μοναδικών χαρακτηριστικών πάνω στο πρόσωπό μας, με άμεσο σκοπό τον αποκλεισμό της πρόσβασης κάποιου ανεπιθύμητου ατόμου στο τηλέφωνο ή το σύστημά σας.

➤ Αναγνώριση αριθμού πινακίδας

Πολλά φανάρια και κάμερες κάνουν χρήση της αναγνώρισης πινακίδων για να αποδώσουν το πρόστιμο που αναλογεί στους παραβάτες του κώδικα οδικής κυκλοφορίας. Η τεχνολογία αναγνώρισης πινακίδων επιτρέπει σε ένα σύστημα κυκλοφορίας την αναγνώριση ενός οχήματος και τη συλλογή δεδομένων που αφορούν τον ιδιοκτήτη του οχήματος αυτού. Γίνεται

χρήση της κατάτμησης της εικόνας για να διαχωριστεί ο αριθμός που αντιστοιχεί στην εκάστοτε πινακίδα, όπως επίσης και τα δεδομένα από τα υπόλοιπα αντικείμενα που υπάρχουν σε αυτή. Αυτή η μέθοδος έχει απλουστεύσει σε μεγάλο βαθμό τη διαδικασία της απόδοσης προστίμων σε άτομα που παρουσιάζουν παραβατική συμπεριφορά.

➤ Αναζήτηση βάση εικόνας

Η Google και ένα πλήθος μηχανών αναζήτησης, οι οποίες δίνουν τη δυνατότητα αναζήτησης σύμφωνα με κάποια εικόνα, την οποία παραθέτουμε εμείς σε αυτές, χρησιμοποιούν μεθόδους κατάτμησης ψηφιακών εικόνων για να προχωρήσουν στον προσδιορισμό των αντικειμένων που λαμβάνουν χώρα στην εικόνα που εμείς δώσαμε προς αναζήτηση. Με τον τρόπο αυτό πραγματοποιήσουν τη σύγκρισή τους με τις παρόμοιες εικόνες που θεωρούν σχετικές, έτσι ώστε να μας προσφέρουν τα αποτελέσματα της αναζήτησης μας.

➤ Ιατρική απεικόνιση

Στον τομέα της ιατρικής, πραγματοποιείται χρήση της τεχνολογίας της κατάτμησης των βιοϊατρικών εικόνων για τον εντοπισμό και την αναγνώριση καρκινικών κυττάρων, τον προσδιορισμό του μεγέθους των ιστών, την διεξαγωγή εικονικών χειρουργικών προσομοιώσεων και την διεξαγωγή ενδοχειρουργικής πλοήγησης. Η κατάτμηση εικόνας βοηθά στον εντοπισμό των πληγέντων περιοχών με σκοπό τον προγραμματισμό των θεραπειών που κρίνονται κατάλληλες για την εξυγίανση τους.

Πέρα από τους κλάδους που έχουμε αναφέρει πιο πάνω, η κατάτμηση εικόνας έχει κάνει την εμφάνιση της στη βιομηχανία, τη γεωργία, την ασφάλεια και πληθώρα άλλων περιοχών ενδιαφέροντος και αναμένεται μελλοντικά η χρήση της τεχνολογίας αυτής να επεκταθεί σε όλο και μεγαλύτερο αριθμό τομέων.

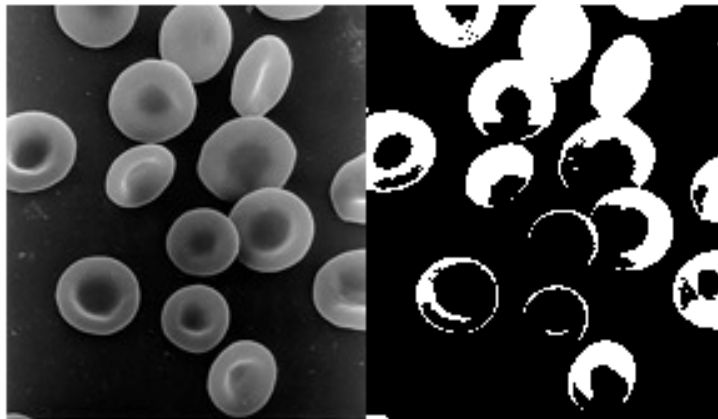
7.1 Κατάτμηση με κατωφλίωση

Η πιο απλή τεχνική κατάτμησης μίας ψηφιακής εικόνας είναι η τεχνική της κατωφλίωσης. Διακρίνει τα εικονοστοιχεία που απαρτίζουν την εικόνα κάνοντας σύγκριση της φωτεινότητας του κάθε εικονοστοιχείου με μια τιμή που έχει εκ των προτέρων οριστεί και αποτελεί το «κατώφλι». Η τεχνική αυτή είναι αποτελεσματική όταν τα επιθυμητά αντικείμενα έχουν μεγαλύτερη φωτεινότητα από το background (αδιάφορα αντικείμενα). Η μέθοδος της κατωφλίωσης μετατρέπει μια grayscale εικόνα σε δυαδική εικόνα διαιρώντας την σε δύο τμήματα (επιθυμητά και ανεπιθυμητά τμήματα). Με βάση τις διαφορετικές τιμές κατωφλίου, κάποιες κατηγορίες τεχνικών κατάτμησης κατωφλίου τις οποίες μπορούμε να διακρίνουμε, είναι οι παρακάτω:

7.1.1 Κατάτμηση με απλή κατωφλίωση

Στην μέθοδο αυτή, το κάθε εικονοστοιχείο παρουσιάζεται είτε με μαύρο, είτε με λευκό χρώμα, σύμφωνα με το κατώφλι που καθορίζουμε και αντίστοιχα αν η τιμή του εικονοστοιχείου βρίσκεται κάτω ή πάνω από το κατώφλι αυτό. Συγκεκριμένα, αν είναι χαμηλότερα, εμφανίζεται μαύρο, ενώ αν είναι ψηλότερα, εμφανίζεται άσπρο.

➤ Παρακάτω παρουσιάζεται ένα παράδειγμα εφαρμογής απλής κατωφλίωσης:

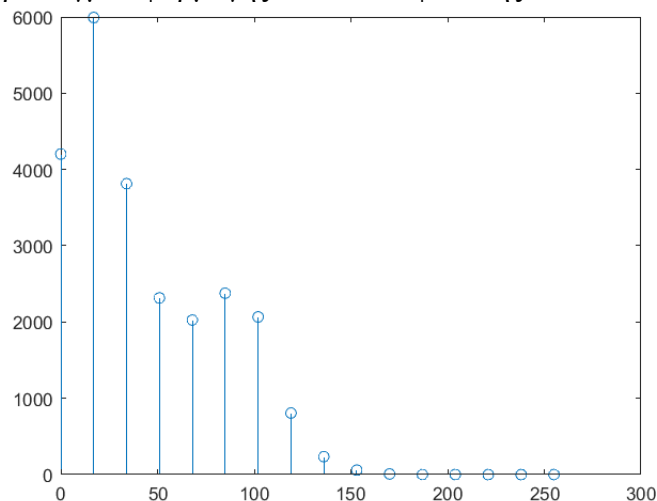


Εικόνα 7.7.1: Κατάτμηση εικόνας με απλή κατωφλίωση και κατώφλι $T=0,5$
(Πηγή (αρχικής εικόνας): https://en.wikipedia.org/wiki/Red_blood_cell)

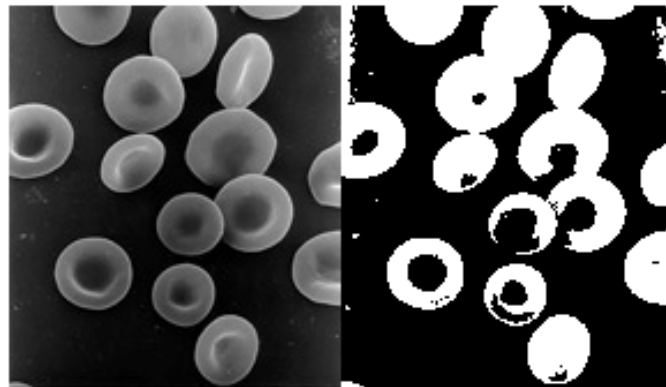
7.1.2 Κατάτμηση με κατωφλίωση Otsu

Στην απλή κατωφλίωση ο καθένας που θέλει να την εφαρμόσει, καθορίζει αυθαίρετα την επιθυμητή τιμή κατωφλίου με στόχο την αποτελεσματικότερη κατάτμηση της επιλεγμένης εικόνας. Παρόλα αυτά, για να βρεθεί η καταλληλότερη τιμή του κατωφλίου απαιτούνται δοκιμές με ποικίλες τιμές. Στην κατωφλίωση με τη μέθοδο Otsu, η επιλογή του καταλληλότερου κατωφλίου γίνεται αυτόματα, με βάση του ιστογράμματος της εικόνας, στο οποίο παρουσιάζονται δύο κορυφές, μία για το παρασκήνιο και μία για το προσκήνιο. Το κατώφλι προκύπτει από την ενδιάμεση τιμή αυτών των δύο κορυφών. Η τεχνική αυτή χρησιμοποιείται συχνά στην αφαίρεση του χρώματος, στην αναγνώριση μοτίβων και στο σκανάρισμα εγγράφων.

➤ Ακολουθεί ένα παράδειγμα εφαρμογής Otsu κατωφλίωσης :



Εικόνα 7.7.2: Ιστόγραμμα εικόνας 'Redbloodcells.jpg'



Εικόνα 7.7.3: Κατάτμηση εικόνας 'Redbloodcells.jpg' με Otsu κατωφλίωση

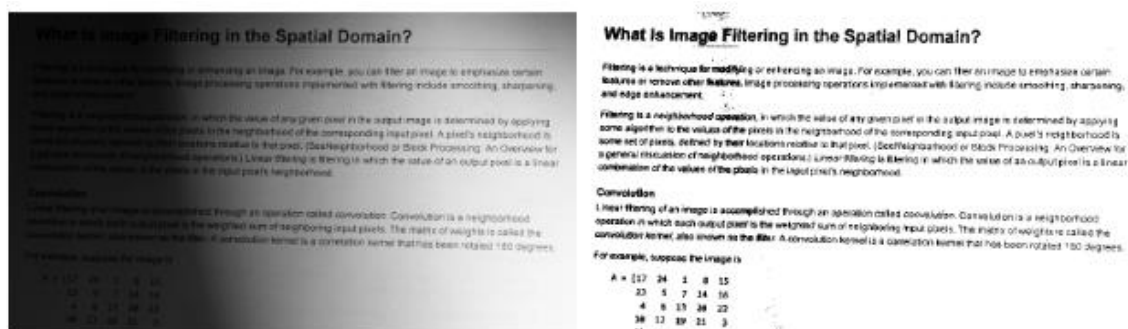
(Πηγή (αρχικής εικόνας): https://en.wikipedia.org/wiki/Red_blood_cell)

Παρατηρούμε, όπως περιμέναμε άλλωστε, ότι στην δεύτερη περίπτωση (Otsu), τα αντικείμενα στο προσκήνιο έχουν διαχωριστεί πιο αποτελεσματικά από το παρασκήνιο, σε σχέση με την πρώτη περίπτωση (απλή κατωφλίωση).

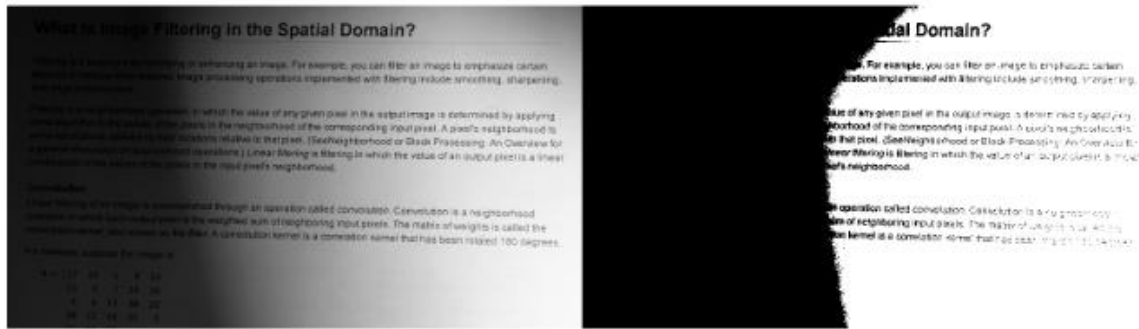
7.1.3 Κατάτμηση με προσαρμοσμένη κατωφλίωση (adaptive thresholding)

Η ύπαρξη ενός σταθερού κατωφλίου δεν είναι η καταλληλότερη λύση για την κατάτμηση οποιασδήποτε εικόνας. Πληθώρα εικόνων διαθέτουν διαφορετικά παρασκήνια και συνθήκες που επηρεάζουν τις ιδιότητες τους. Για το λόγο αυτό, επιλέγεται συχνά η χρήση μεταβαλλόμενου κατωφλίου για την πραγματοποίηση κατάτμησης τέτοιων εικόνων, αντί για τη χρήση σταθερού κατωφλίου. Η μέθοδος αυτή ενεργεί καλύτερα σε εικόνες με διαφορετικές συνθήκες φωτεινότητας στην έκτασή τους. Έτσι χρειάζεται ένας αλγόριθμος που χωρίζει την εικόνα σε μικρότερα τμήματα και υπολογίζει το κατώφλι για καθένα από αυτά.

- Παρακάτω βλέπουμε ένα παράδειγμα εφαρμογής προσαρμοσμένης κατωφλίωσης και τη σύγκριση με την εφαρμογή ενός σταθερού κατωφλίου (Otsu) στην ίδια εικόνα :



Εικόνα 7.7.4: Κατάτμηση εικόνας 'printedtext.png' με προσαρμοσμένη κατωφλίωση



Εικόνα 7.7.5: Κατάτμηση εικόνας 'printedtext.png' με Otsu κατωφλίωση

Παρατηρούμε ότι στην συγκεκριμένη εικόνα, όπου η φωτεινότητα δεν είναι ομαλή σε όλη την έκταση της, η χρήση προσαρμοσμένης κατωφλίωσης είναι πολύ πιο κατάλληλη για την κατάτμηση του κειμένου, σε σχέση με την κατάτμηση με χρήση ενός σταθερού κατωφλίου για όλη την έκταση της εικόνας.

7.2 Κατάτμηση Βάσει Περιοχής (Region Based Segmentation)

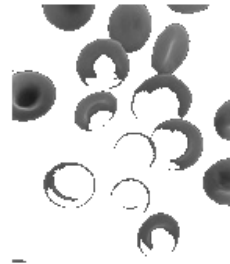
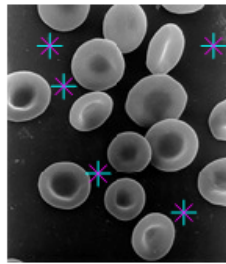
Η τεχνική της κατάτμησης βάσει περιοχής βασίζεται στο διαχωρισμό της εικόνας σε τμήματα με χαρακτηριστικά που παρουσιάζουν κάποιες ομοιότητες. Τα τμήματα αυτά είναι ουσιαστικά κάποιες ομάδες από εικονοστοιχεία, οι οποίες εντοπίζονται από τον αλγόριθμο, εντοπίζοντας σημείο σπόρου, το οποίο υπάρχει περίπτωση να αντιστοιχεί σε ένα μικρό ή μεγάλο τμήμα της αρχικής εικόνας. Αφού έχει γίνει ο εντοπισμός των σημείων αυτών, ο αλγόριθμος της κατάτμησης θα τα εμπλουτίσει με περισσότερα εικονοστοιχεία ή θα τους αφαιρέσει εικονοστοιχεία.

7.2.1 Κατάτμηση με ανάπτυξη περιοχής (Region Growing Segmentation)

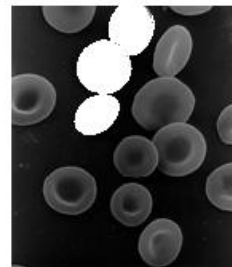
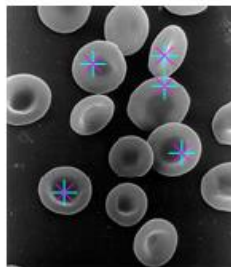
Σύμφωνα με την τεχνική αυτή, αρχίζουμε με ένα σχετικά περιορισμένο σύνολο εικονοστοιχείων και στη ακολούθως προχωράμε επαναληπτικά στη συγχώνευση περισσότερων εικονοστοιχείων σύμφωνα με συγκεκριμένες συνθήκες ομοιότητας. Ένας αλγόριθμος ανάπτυξης περιοχής διαλέγει αυθαίρετα ένα τυχαίο εικονοστοιχείο σπόρου στην εικόνα, το συγκρίνει με τα γειτονικά εικονοστοιχεία και προχωράει στον εμπλουτισμό της περιοχής του σπόρου εντοπίζοντας σημεία που παρουσιάζουν ομοιότητες με το σημείο του σπόρου.

Σε περίπτωση που κάποια περιοχή πάψει να διογκώνεται περαιτέρω, ο αλγόριθμος αποφασίζει να διαλέξει ένα άλλο εικονοστοιχείο σπόρου το οποίο είναι πιθανό να μην ανήκει σε καμία περιοχή. Μια περιοχή είναι συχνά πιθανό να εμφανίζει πληθώρα χαρακτηριστικών, γεγονός που την υποχρεώνει να αντιπροσωπεύσει εσφαλμένα τη μεγαλύτερη έκταση της εικόνας. Για να αποφευχθεί ένα τέτοιο σφάλμα, οι αλγόριθμοι ανάπτυξης περιοχής αναπτύσσουν πολλές περιοχές ταυτόχρονα.

- Παρακάτω βλέπουμε ένα παράδειγμα εφαρμογής κατάτμησης με ανάπτυξη περιοχής, με διαφορετικά σημεία τοποθέτησης των seed points :



Εικόνα 7.7.6: Κατάτμηση εικόνας 'Redbloodcells.jpg' με 5 seeds



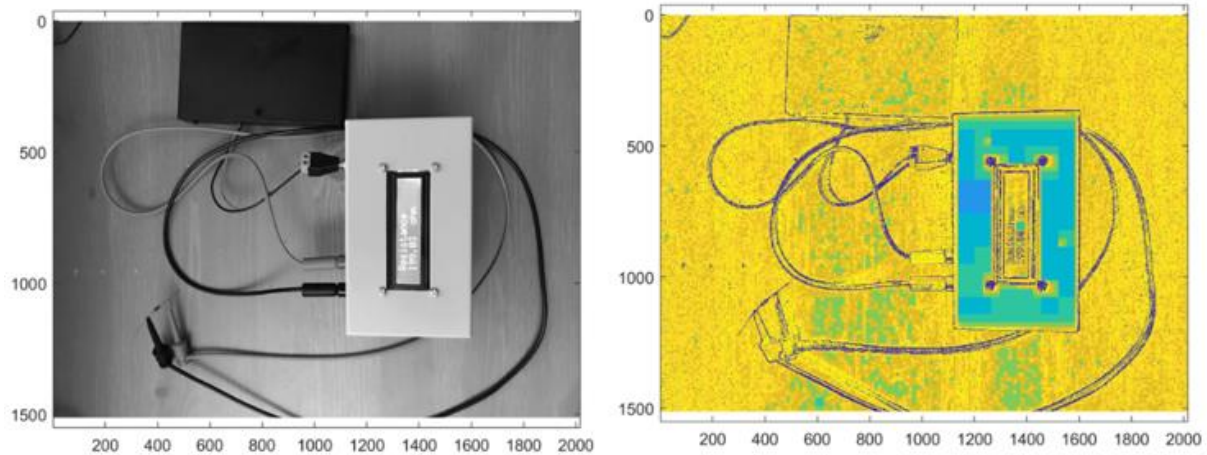
Εικόνα 7.7.7: Κατάτμηση εικόνας 'Redbloodcells.jpg' με 5 seeds

7.2.2 Κατάτμηση με διαχωρισμό και συγχώνευση(Region Splitting and Merging)

Σε αυτήν την τεχνική κατάτμησης όπως είναι φανερό και από την ονομασία, οι προς επεξεργασία ψηφιακές εικόνες χωρίζονται σε τμήματα και μετά επανενώνονται. Πιο συγκεκριμένα, σε πρώτο στάδιο, ο αλγόριθμος μοιράζει την εικόνα σε τμήματα που παρουσιάζουν κοινά χαρακτηριστικά και προχωράει στη συγχώνευση των παρακείμενων τμημάτων τους που είναι παρόμοια μεταξύ τους. Αμέσως μετά από κάθε μοίρασμα, γίνεται ένας έλεγχος, για να ληφθεί η απόφαση, εάν το εκάστοτε τμήμα επιδέχεται περαιτέρω τμηματοποίηση. Η απόφαση αυτή λαμβάνεται με βάση το κριτήριο της ομοιογένειας, για παράδειγμα εάν τα επίπεδα της κλίμακας του γκρι δεν μεταβάλλονται έστω βρίσκονται κάτω από μία συγκεκριμένη τιμή κατωφλίου, τότε θεωρείται πως υπάρχει ομοιογένεια.

Σε αντίθεση με την περιοχή της ανάπτυξης περιοχής, εδώ ο αλγόριθμος λαμβάνει υπόψη όλη την έκταση της εικόνας για να προχωρήσει την κατάτμησή της, ενώ στην ανάπτυξη περιοχής το σημείο ενδιαφέροντος του αλγορίθμου αφορά συγκεκριμένα εικονοστοιχεία, όπως είδαμε παραπάνω.

- Παρακάτω βλέπουμε ένα παράδειγμα της τεχνικής αυτής, όπου τα τμήματα της εικόνας των οποίων τα εικονοστοιχεία διαφέρουν κάτω από την τιμή 10 στην κλίμακα του γκρι, θεωρούνται ως ομοιογενή τμήματα της εικόνας και συγχωνεύονται μεταξύ τους.



Εικόνα 7.7.8: Κατάτμηση εικόνας με διαχωρισμό και συγχώνευση

(Πηγή : https://en.wikipedia.org/wiki/Split_and_merge_segmentation#cite_note-2)

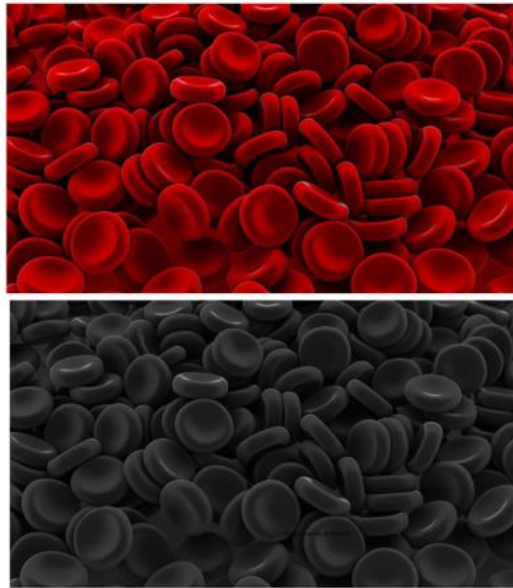
7.3 Κατάτμηση με μέθοδο Watershed

Με την αναφορά στη μέθοδο αυτή γενικά περιγράφουμε έναν μετασχηματισμό σε μία εικόνα της κλίμακας του γκρι. Η τεχνική του watershed χρησιμοποιεί έναν αλγόριθμο ο οποίος συμπεριφέρεται στην ψηφιακή εικόνα σαν αυτή να αποτελεί έναν τοπογραφικό χάρτη. Αντιλαμβάνεται λοιπόν την φωτεινότητα του εκάστοτε εικονοστοιχείου σαν αυτό να είναι μία κορυφή του τοπογραφικού χάρτη και εντοπίζει τις γραμμές κατά μήκος της των κορυφών αυτών. Στην πράξη, όταν μιλάμε για κορυφές, εννοούμε την απόσταση του κάθε εικονοστοιχείου από το πλησιέστερο εικονοστοιχείου με τιμή διάφορη του 0.

Η τεχνική του watershed διαθέτει ποικίλους τεχνικούς ορισμούς και εφαρμόζεται σε πληθώρα διαφόρων κλάδων. Πέρα από τον προσδιορισμό των κορυφών του τοπογραφικού χάρτη (εικόνα), ο αλγόριθμος που χρησιμοποιείται για την μέθοδο αυτή, εντοπίζει επίσης τις αυλακώσεις τις οποίες γεμίζει εωσότου ενωθούν με τις κορυφές. Έτσι η ψηφιακή εικόνα με τη μέθοδο αυτή μοιράζεται σε περιοχές με βάση τη διαφορά του «ύψους» των εικονοστοιχείων πάνω στην εικόνα.

Ακολουθεί ένα παράδειγμα κατάτμησης με την τεχνική Watershed με την εισαγωγή εσωτερικών και εξωτερικών markers (σημαδιών), που ορίζουν ουσιαστικά τα σημεία εκκίνησης του γεμίματος και τα σημεία όπου αυτό θα τερματίζεται:

- Επιλογή εικόνας και μετατροπή της σε grayscale εικόνα:



Εικόνα 7.7.9: Εικόνα ερυθρών κυττάρων σε rgb και grayscale

(Πηγή : <https://www.medicalnewstoday.com/articles/319544>)

- Υπολογισμός του gradient magnitude, το οποίο υψηλό στα όρια των αντικειμένων και χαμηλό (κυρίως) μέσα στα αντικείμενα.

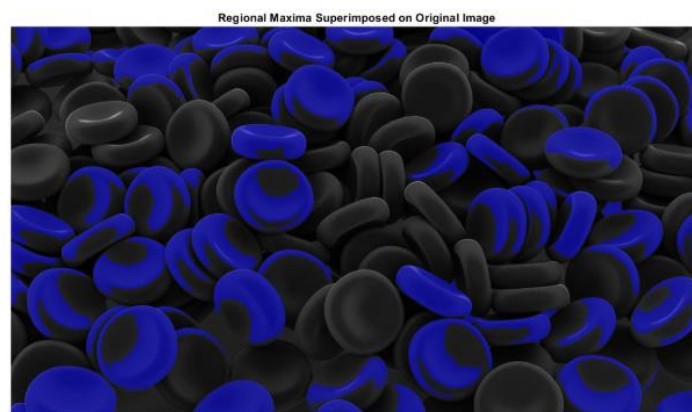


Εικόνα 7.7.10: Gradient magnitude

- Δημιουργία εσωτερικών markers για το «μαρκάρισμα» των αντικειμένων στο προσκήνιο με την αξιοποίηση της τεχνικής εντοπισμού των τοπικών μεγίστων και εφαρμογή των εσωτερικών markers στην αρχική grayscale εικόνα:

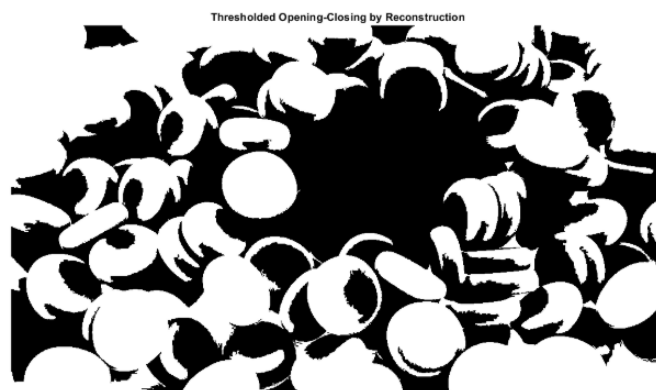


Εικόνα 7.7.11: Εντοπισμός τοπικών μεγίστων

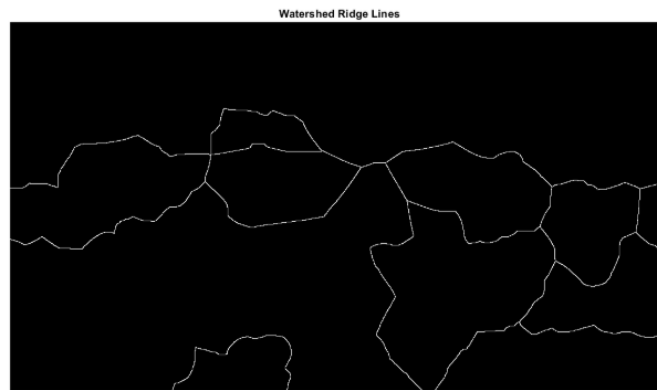


Εικόνα 7.7.12: Εφαρμογή εσωτερικών markers

- Δημιουργία εξωτερικών markers για το «μαρκάρισμα» των αντικειμένων στο παρασκήνιο με την χρήση της μεθόδου της κατωφλίωσης. Τα μαύρα εικονοστοιχεία ανήκουν στο παρασκήνιο.

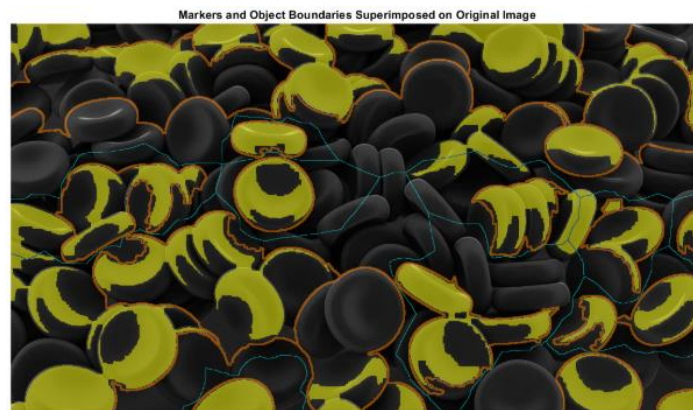


Εικόνα 7.7.13: Εντοπισμός αντικειμένων παρασκήνιου



Εικόνα 7.7.14: Εξωτερικοί markers

- Τροποποιείται η εικόνα gradient magnitude , με σκοπό να συναντάμε τοπικά ελάχιστα μόνο στα εικονοστοιχεία των εσωτερικών και εξωτερικών markers και εφαρμόζονται όλοι οι παραπάνω υπολογισμοί για την εμφάνιση του τελικού διαχωρισμού.



Εικόνα 7.7.15: Αποτέλεσμα Watershed

7.4 Κατάτμηση με Αλγόριθμους Ταξινόμησης (Clustering-Based Segmentation)

Η κατάτμηση της ψηφιακής εικόνας με την τεχνική της ομαδοποίησης, κάνει χρήση διαφόρων αλγορίθμων, οι οποίοι είναι μη επιβλεπόμενοι και μας καθιστούν ικανούς να εντοπίζουμε κρυμμένα δεδομένα των εικόνων, που ίσως δεν είναι εύκολα αντιληπτά από το ανθρώπινο μάτι άμεσα, όπως για παράδειγμα πληροφορίες που αφορούν την ομάδα, τις δομές, τις σκιάσεις κ.λπ. Όπως είναι φανερό και από την ονομασία της τεχνικής αυτή, οι αλγόριθμοι που χρησιμοποιούνται, τμηματοποιούν την εικόνα σε διάφορες ομάδες εικονοστοιχείων που παρουσιάζουν όμοια συμπεριφορά. Χωρίζονται λοιπόν τα στοιχεία των δεδομένων σε ομάδες , όπου τα στοιχεία της εκάστοτε ομάδας εμφανίζουν μεταξύ τους περισσότερες ομοιότητες σε σχέση με τα στοιχεία που ανήκουν σε άλλες ομάδες.

Κάποιοι από τους ευρέως πιο διαδεδομένους αλγορίθμους, που χρησιμοποιούνται για την κατάτμηση με ταξινόμηση είναι οι αλγόριθμοι fuzzy c-means και k-means. Στις εφαρμογές κατάτμησης εικόνας χρησιμοποιείται συχνότερα ο αλγόριθμος k-means για την ομαδοποίηση, καθώς δίνει αρκετά ικανοποιητικά αποτελέσματα, χωρίς να έχει μεγάλη περιπλοκότητα.

7.4.1 Κατάτμηση με Αλγόριθμους Ταξινόμησης Fuzzy C means

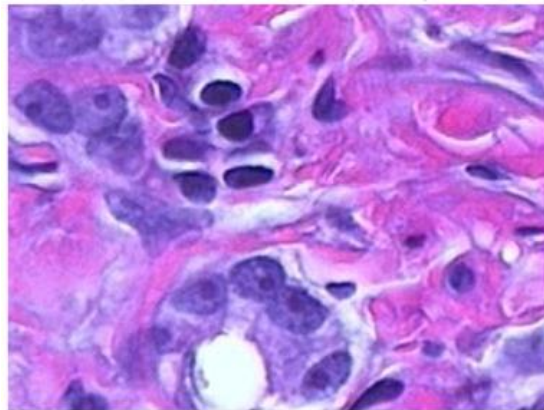
Με τον αλγόριθμο της μεθόδου Fuzzy C means , τα εικονοστοιχεία σε μίας εικόνα είναι πιθανό να ανήκουν σε πολλαπλές ομάδες. Παρόλα αυτά τα επίπεδα ομοιότητας κάθε εικονοστοιχείου με κάθε ομάδα ποικίλουν. Ο αλγόριθμος αυτός διαθέτει μία διαδικασία βελτιστοποίησης, η οποία καθιστά τα αποτελέσματα που λαμβάνουμε, πιο ικανοποιητικά.

7.4.2 Κατάτμηση με Αλγόριθμους Ταξινόμησης K-means Clustering

Ο αλγόριθμος K-means αποτελεί ένα απλό μη επιβλεπόμενο είδος αλγορίθμου μηχανικής μάθησης. Αντιστοιχίζει μία εικόνα σε μια ομάδα μέσα από ένα πλήθος ομάδων. Ο αλγόριθμος σε πρώτο στάδιο μοιράζει την έκταση της εικόνας σε k αριθμό ομάδων. Αμέσως μετά γίνεται υπολογισμός του κέντρου ή αλλιώς το κεντροειδές της κάθε ομάδας και πραγματοποιεί καινούριο μοίρασμα με σκοπό το καθένα εικονοστοιχείο να αντιστοιχίζεται με το πλησιέστερο κεντροειδές. Στη συνέχεια προσδιορίζονται ξανά τα κέντρα για τα καινούρια clusters. Ο αλγόριθμος πραγματοποιείται κατά τις προαναφερθείσες διαδικασίες, εωσότου τα εικονοστοιχεία να μην είναι σε θέση να αντιστοιχιστούν σε διαφορετική ομάδα. Ο K-means αποτελεί έναν πολύ διαδεδομένο αλγόριθμο, καθώς καταφέρνει να φτάσει σε τέλμα σε σύντομο χρονικό διάστημα, χωρίς να έχουμε βέβαια συνέχεια την απόλυτη ακρίβεια. Ένας περιορισμός του αλγορίθμου αυτού είναι ο προκαθορισμένος αριθμός των clusters.

Παρακάτω θα δούμε ένα παράδειγμα κατάτμησης των μπλε πυρήνων μίας εικόνας ιστού και τη χρώση του με αιμοτοξυλίνη και ηωσίνη, μέθοδος η οποία βοηθά στην αναγνώριση του τύπου των ιστών.

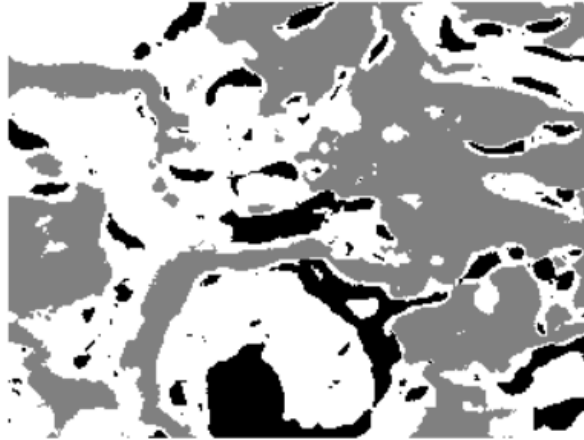
- Εισάγεται η αρχική μας εικόνα:



Εικόνα 7.7.16: Ιστός με χρώση του με αιμοτοξυλίνη και ηωσίνη

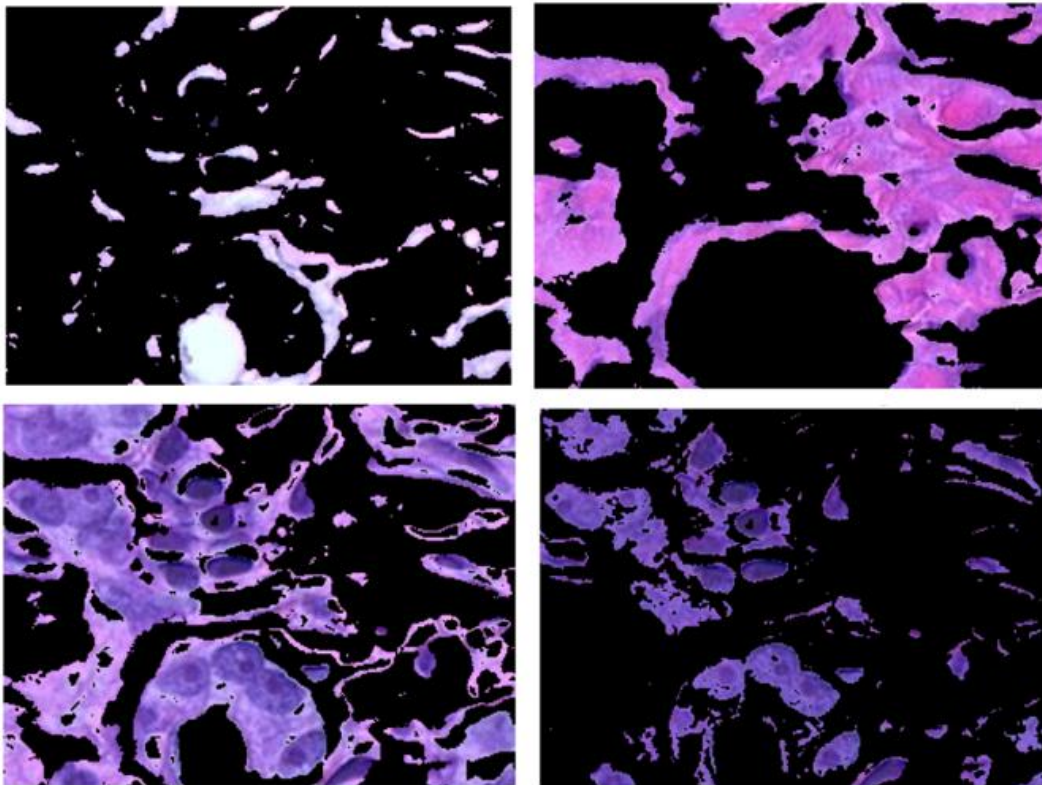
- Μεταφορά της εικόνας από τον RGB χρωματικό χώρο στον $L^*a^*b^*$ (επίσης γνωστός ως CIELAB ή CIE $L^*a^*b^*$) για να καταστεί δυνατή η ποσοτικοποίηση των οπτικών διαφορών.
- Εφαρμογή των labels των ομάδων σύμφωνα με την μέθοδο του K-Means clustering. αντιμετωπίζει κάθε αντικείμενο ως μια θέση στο χώρο. Βρίσκει διαμερίσματα τέτοια ώστε τα αντικείμενα μέσα σε κάθε σύμπλεγμα να είναι όσο το δυνατόν πιο κοντά το ένα στο άλλο και όσο το δυνατόν πιο μακριά από αντικείμενα σε άλλες ομάδες. Η ομαδοποίηση K-

means προϋποθέτει τον καθορισμό εκ των προτέρων του αριθμού των ομάδων και μια μονάδα μέτρησης απόστασης για ποσοτικοποιηθεί η απόσταση μεταξύ των αντικειμένων:



Εικόνα 7.17: Εφαρμογή των labels των ομάδων

- Δημιουργία εικόνων κατάτμησης των αντικειμένων του κάθε χρώματος από την αρχική εικόνα με σκοπό την τελική κατάτμηση των μπλε πυρήνων :



Εικόνα 7.18: Ομάδα πρώτη (πάνω αριστερά), Ομάδα δεύτερη (πάνω δεξιά), Ομάδα τρίτη (κάτω αριστερά), Μπλε πυρήνες (κάτω δεξιά)

8. Κατάτμηση Βιοϊατρικών Εικόνων μέσω Συνελικτικών Νευρωνικών Δικτύων.

8.1 Συνοπτική περιγραφή

Πραγματοποιείται στο περιβάλλον αριθμητικής υπολογιστικής MATLAB η υλοποίηση δύο διαφορετικών αρχιτεκτονικών CNN, για την κατάτμηση βιοϊατρικών εικόνων, και συγκεκριμένα την κατάτμηση της περιοχής των πνευμόνων από τις ακτινογραφίες θώρακος, καθώς επίσης εκτιμάται οπτικά και ποσοτικά η αποτελεσματικότητα του εκάστοτε δικτύου εξ αυτών, ενώ παρατίθενται και όποια συμπεράσματα απορρέουν από την ανωτέρω υλοποίηση. Τέλος σημειώνεται ότι τα τμήματα του κώδικα των οποίων έγινε χρήση, προσκομίζονται στο κλείσιμο της εργασίας το πεδίο των παραρτημάτων.

8.2 Στόχος

Στόχος της προαναφερθείσας υλοποίησης είναι η σύγκριση των δύο συνελικτικών νευρωνικών δικτύων, ενός CNN τυπικής αρχιτεκτονικής με τέσσερα επίπεδα συνέλιξης (convolutional layers) και του U-net, ενός συνελικτικού νευρωνικού δικτύου που αναπτύχθηκε ειδικά για την κατάτμηση βιοϊατρικής εικόνας. Το τελευταίο βασίζεται στα πλήρως συνελικτικά δίκτυα και η αρχιτεκτονική του τροποποιήθηκε και επεκτάθηκε ώστε να λειτουργεί με λιγότερες εικόνες εκπαίδευσης και να παρέχει εξίσου ακριβείς κατατμήσεις. Πιο συγκεκριμένα, σκοπός μας είναι να παρατηρήσουμε τα αποτελέσματα της κατάτμησης για το εκάστοτε δίκτυο, πριν και μετά την τροποποίηση ενός παράγοντα, του mini-batch, δηλαδή του υποσυνόλου των δειγμάτων (εικόνων) των οποίων γίνεται χρήση από το κάθε δίκτυο.

8.3 Θεωρητικό Πλαίσιο Πειράματος

8.3.1 Matlab και Τύποι Εικόνων στο Matlab

Η υλοποίηση για την οποία θα γίνει εκτενής περιγραφή στη συνέχεια, πραγματοποιήθηκε όπως προαναφέραμε στο περιβάλλον αριθμητικής υπολογιστικής του Matlab. Η χρήση του Matlab αφορά κυρίως μαθηματικά προβλήματα, μπορεί όμως να το χρησιμοποιήσουμε και για την διευθέτηση πιο απαιτητικών εργασιών, όπως για παράδειγμα για να επεξεργαστούμε σήματα και εικόνες.

Όσον αφορά τις ιατρικές εικόνες, κύριο αντικείμενο της υλοποίησης, συναντάμε στο περιβάλλον του Matlab σημαντικά εργαλεία και βιβλιοθήκες που μας καθιστούν ικανούς να απεικονίσουμε, να επεξεργαστούμε και να αναλύσουμε τέτοιου τύπου εικόνες. Για την καλύτερη κατανόηση της επεξεργασίας αυτής, σημαντικό είναι να γίνει αναφορά στους τύπους των εικόνων και την αντιμετώπισή τους από το υπολογιστικό περιβάλλον αυτό.

Πιο συγκεκριμένα, οι εικόνες στο Matlab είναι πίνακες αριθμητικών στοιχείων, ενός εκ των παρακάτω τύπων:

- **double:** Πραγματικοί αριθμοί διπλής ακρίβειας, μήκους 8 bytes. Συναντάμε τα στοιχεία του τύπου αυτού κυρίως στο διάστημα $[0,1]$, όπου το 0 αντιστοιχεί στο μαύρο και το 1 στο λευκό. Ενδεικτικά μία εικόνα διαστάσεων 512 x 512 εικονοστοιχείων και τύπου double απαιτεί χωρητικότητα 2 MB. Το Matlab λειτουργεί συνήθως με συναρτήσεις στοιχείων αυτού του τύπου, οπότε είναι σημαντικό να μετατρέψουμε τις εικόνες, στις οποίες θέλουμε να εφαρμόσουμε κάποια συνάρτηση, στο διάστημα $[0, 1]$, για την αποτελεσματικότερη επεξεργασία
- **uint8:** Ακέραιοι μη προσημασμένοι αριθμοί με μήκος 1 byte. Τέτοιου τύπου στοιχεία βρίσκονται στο διάστημα $[0, 255]$, όπου το 0 αντιστοιχεί στο μαύρο και το 255 στο λευκό. Σε εικόνες του αριθμητικού τύπου αυτού είναι δυνατή, ωστόσο για την εφαρμογή κάποιας μαθηματικής συνάρτησης σε αυτές θα πρέπει να γίνει πρώτα η μετατροπή τους σε double στο $[0,1]$ διάστημα.
- **uint16:** Ακέραιοι μη προσημασμένοι αριθμοί με μήκος 2 byte. Τέτοιου τύπου στοιχεία βρίσκονται στο διάστημα $[0, 65535]$, όπου το 0 αντιστοιχεί στο μαύρο και το 65535 στο λευκό.
- **logical:** Λογικές τιμές 0 ή 1, μήκους ενός byte.

Παρακάτω βλέπουμε ένα απλό παράδειγμα ανάγνωσης και απεικόνισης μίας εικόνας τύπου Uint8 :

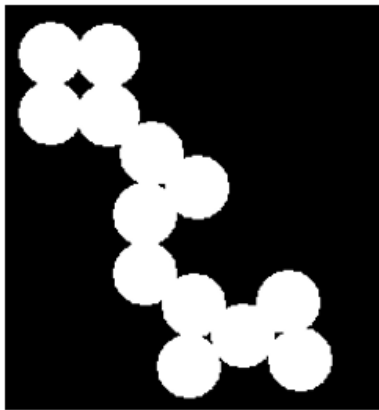


```
>> apeikonisi
Name      Size      Bytes  Class  Attributes
x         291x240    69840  uint8
```

Εικόνα 8.1: Παράδειγμα ανάγνωσης και απεικόνισης μίας εικόνας

Ενώ αναφέραμε τους τύπους των εικονοστοιχείων πρέπει να σημειωθεί ότι οι τύποι εικόνων που υποστηρίζονται από το Matlab είναι οι εξής:

- Εικόνες φωτεινότητας στις οποίες ο αριθμητικός τύπος των στοιχείων τους είναι double, Uint8 ή Uint16.
- Δυαδικές εικόνες, όπου κάθε εικονοστοιχείο είναι logical, παίρνει δηλαδή λογικές τιμές 0 ή 1.



```
>> binaryapeikonisi
Name      Size      Bytes  Class  Attributes
x         256x256   65536  logical
```

Εικόνα 8.2: Παράδειγμα ανάγνωσης και απεικόνισης μίας binary εικόνας

- Έγχρωμες εικόνες (RGB), που αποτελούνται από τρεις πίνακες (έναν για κάθε βασικό χρώμα)



```
Name      Size      Bytes  Class  Attributes
x         1536x2048x3  9437184  uint8
```

Εικόνα 8.3: Παράδειγμα ανάγνωσης και απεικόνισης μίας RGB εικόνας

- Εικόνες χρωματικού δείκτη, που αποτελούνται από ένα πίνακα με τους δείκτες των τιμών φωτεινότητας για κάθε εικονοστοιχείο και το χρωματικό πίνακα με διαστάσεις $m \times 3$ που περιέχει τις τιμές RGB. Οι τιμές του πίνακα δεικτών είναι δείκτες του χρωματικού πίνακα, όπου βρίσκεται η αναλογία των τριών χρωμάτων. Ο αριθμητικός τύπος του πίνακα δεικτών είναι double ή Uint8 στο $[0, 255]$, ενώ του χρωματικού είναι double στο $[0, 1]$.



```
X         256x256   524288  double
map       255x3     6120    double
```

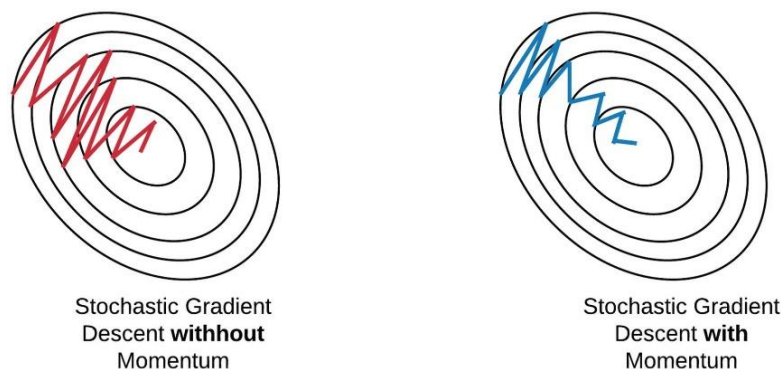
Εικόνα 8.4: Παράδειγμα ανάγνωσης και απεικόνισης μίας εικόνας χρωματικού δείκτη

Τέλος, χρήσιμο είναι να γνωρίζουμε τις βασικές συναρτήσεις μετατροπής εικόνων από έναν τύπο σε άλλον. Κάποιες από αυτές είναι :

- gray2ind - εικόνα φωτεινότητας → εικόνα χρωματικού δείκτη
- im2bw - εικόνα φωτεινότητας → δυαδική
- ind2gray - εικόνα χρωματικού δείκτη → εικόνα φωτεινότητας
- rgb2gray - έγχρωμη εικόνα → εικόνα φωτεινότητας
- rgb2ind - έγχρωμη εικόνα → εικόνα χρωματικού δείκτη

8.3.2 Ορμή (Momentum)

Μια τεχνική που θα χρησιμοποιήσουμε στην προπόνηση των δικτύων μας είναι η ορμή , με της οποίας τη χρήση μπορούμε να επιτύχουμε την γρηγορότερη μάθηση, ενώ ταυτόχρονα μειώνουμε την αστάθεια. Η εφαρμογή ορμής στην διαδικασία της μάθησης βοηθά στην εξομάλυνση της μάθησης κυρίως σε σημεία της συνάρτησης σφάλματος όπου εμφανίζεται έντονη καμπυλότητα ή πολλαπλές μικρού μεγέθους κλίσεις ίδιου προσανατολισμού. Με την ορμή συλλέγεται ένας εκθετικά μειούμενος κινούμενος μέσος όρος από όλες τις κλίσεις που παρήλθαν και αξιολογείται η κατεύθυνση τους με σκοπό τον υπολογισμό του προσανατολισμού της συνάρτησης σφάλματος. Η ορμή μπορεί να πάρει τιμές από 0 έως 1.



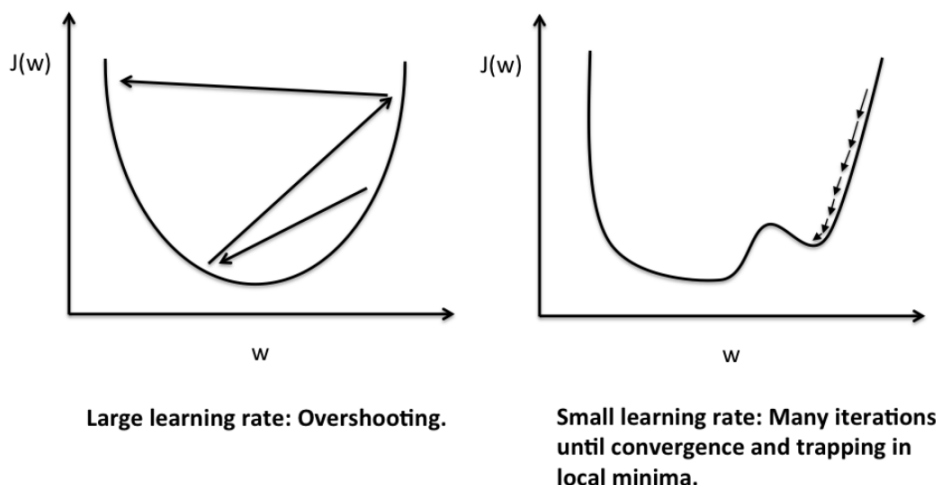
Εικόνα 8.5: Στοχαστική Κατάβαση Κλίσης με και χωρίς ορμή

8.3.3 Ρυθμός Μάθησης (Learning Rate)

Ο ρυθμός εκμάθησης είναι μια διαμορφώσιμη υπερπαράμετρος που χρησιμοποιείται στην εκπαίδευση νευρωνικών δικτύων και έχει μια μικρή θετική τιμή, μεταξύ 0 και 1. Ο ρυθμός εκμάθησης ελέγχει πόσο γρήγορα το μοντέλο προσαρμόζεται στο πρόβλημα. Η αλλαγή της παραμέτρου του ρυθμού εκμάθησης μπορεί να μας βοηθήσει να προσδιορίσουμε εάν κολλάμε στα τοπικά ελάχιστα.

Οι δύο παρακάτω γραφικές παραστάσεις τονίζουν τη σημασία της επιλογής του ρυθμού μάθησης απεικονίζοντας τα δύο πιο κοινά προβλήματα κατά την κλίση κατάβασης:

- Εάν ο ρυθμός εκμάθησης είναι πολύ μεγάλος, η κάθοδος θα υπερβεί τα ελάχιστα και θα αποκλίνει.
- Εάν ο ρυθμός εκμάθησης είναι πολύ μικρός, ο αλγόριθμος θα απαιτήσει πάρα πολλές epochs για να συγκλίνει και μπορεί να παγιδευτεί στα τοπικά ελάχιστα πιο εύκολα.



Εικόνα 8.6: Μεγάλος και μικρός ρυθμός μάθησης

8.3.4 Epoch και Iteration

Μια εποχή (epoch) σε ένα νευρωνικό δίκτυο είναι η εκπαίδευση του νευρωνικού δικτύου με όλα τα δεδομένα εκπαίδευσης για έναν κύκλο, ενώ μια επανάληψη (iteration) περιγράφει πόσες φορές μια παρτίδα δεδομένων πέρασε από τον αλγόριθμο.

8.3.5 Mini-batch

Με τον όρο mini-batch περιγράφουμε ένα σταθερό αριθμό δειγμάτων εκπαίδευσης που είναι μικρότερος από το πραγματικό σύνολο δεδομένων. Έτσι, σε κάθε επανάληψη, εκπαιδεύουμε το δίκτυο σε διαφορετική ομάδα δειγμάτων μέχρι να χρησιμοποιηθούν όλα τα δείγματα του συνόλου δεδομένων.

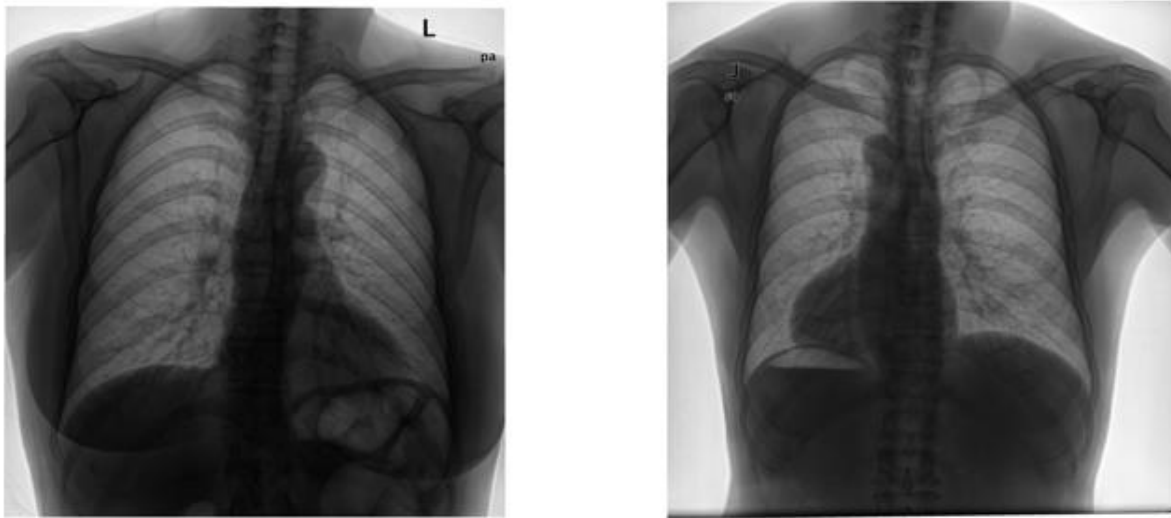
Μεγαλύτερα μεγέθη mini-batch εκτιμούν την κλίση της συνάρτησης σφάλματος με μεγαλύτερη ακρίβεια, κάτι που είναι αναμενόμενο αφού υπολογίζεται η κλίση της συνάρτησης με βάση μεγαλύτερο αριθμό δειγμάτων.

Μικρότερα μεγέθη mini-batch είναι ικανά να προσδώσουν ομαλοποίηση, ενδεχομένως λόγω του γεγονότος ότι προσδίδουν θόρυβο στην διαδικασία της εκμάθησης του δικτύου. Για να πετύχουμε το ελάχιστο δυνατό σφάλμα γενίκευσης θα πρέπει να επιλέξουμε την τιμή 1. Η εκμάθηση όμως με ένα τόσο μικρού μεγέθους mini-batch θα χρειάζεται αντίστοιχα ένα μικρό learning rate με σκοπό να μένει το σύστημα σταθερό καθώς θα εμφανίζονται μεγάλες διακυμάνσεις. Η διάρκεια της εκμάθησης όπως είναι λογικό θα είναι αρκετά μεγάλη βέβαια αν γίνει χρήση μικρού learning rate αλλά και λόγω της παρουσίας των δειγμάτων ένα προς ένα.

8.4 Προ-επεξεργασία Dataset

8.4.1 Dataset και resizing

Για την εκπαίδευση των δικτύων μας χρησιμοποιήσαμε βιοϊατρικές εικόνες και συγκεκριμένα ακτινογραφίες θώρακος. Το dataset δημιουργήθηκε από την εθνική βιβλιοθήκη της Ιατρικής του Μέριλαντ των Ηνωμένων Πολιτειών σε συνεργασία με το πανεπιστημιακό νοσοκομείο της Σεντζέν της Κίνας. Οι ακτινογραφίες θώρακος προέρχονται από εξωτερικά ιατρεία και λήφθηκαν ως μέρος της καθημερινής ρουτίνας εξετάσεων χρησιμοποιώντας συστήματα Ψηφιακής διάγνωσης DR της Philips. Το dataset αποτελείται από 566 εικόνες τύπου PNG και οι διαστάσεις τους ποικίλουν αλλά είναι όλες περίπου 3K x 3K. Παρακάτω βλέπουμε κάποια δείγματα του dataset:



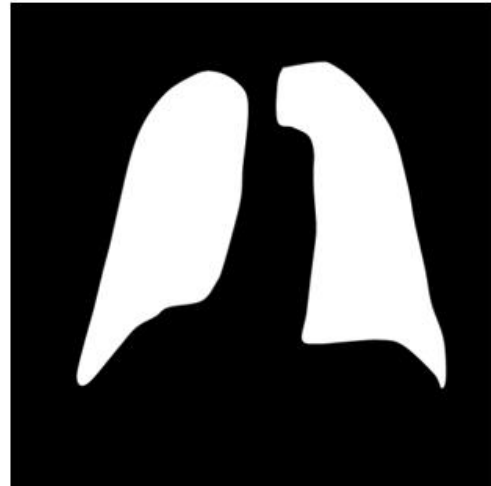
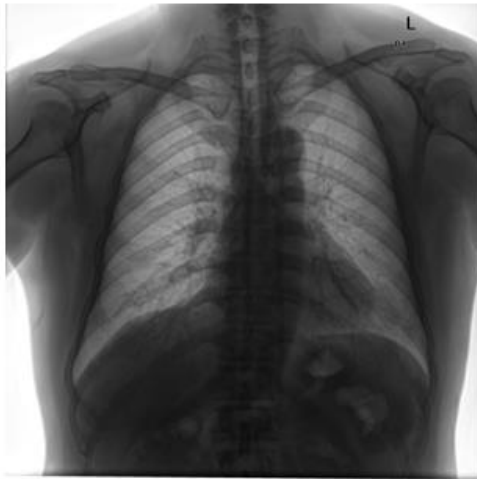
Εικόνα 8.7: Παραδείγματα ακτινογραφιών θώρακος

Πριν εισαχθούν τα δεδομένα μας στα δίκτυα θα πρέπει να προηγηθεί η κανονικοποίησή τους, δηλαδή να έχουν όμοια χαρακτηριστικά και συμβατά με την είσοδο το εκάστοτε δικτύου. Τα συνελκτικά νευρωνικά δίκτυα μαθαίνουν γρηγορότερα όταν οι εικόνες εισόδου αποτελούνται από ένα σχετικά μικρό αριθμό pixels , ενώ σε συγκεκριμένες αρχιτεκτονικές CNN οι εικόνες εισόδου πρέπει να έχουν συγκεκριμένες διαστάσεις , όπως συμβαίνει και στην περίπτωση του Unet. Έτσι λοιπόν αλλάξαμε το μέγεθος όλων των εικόνων του dataset σε 496x496 , διαστάσεις που είναι συμβατές αλλά και αποτελεσματικές με βάση πειράματα που έχουν γίνει πάνω στο Unet.

8.4.1 Labeling του dataset

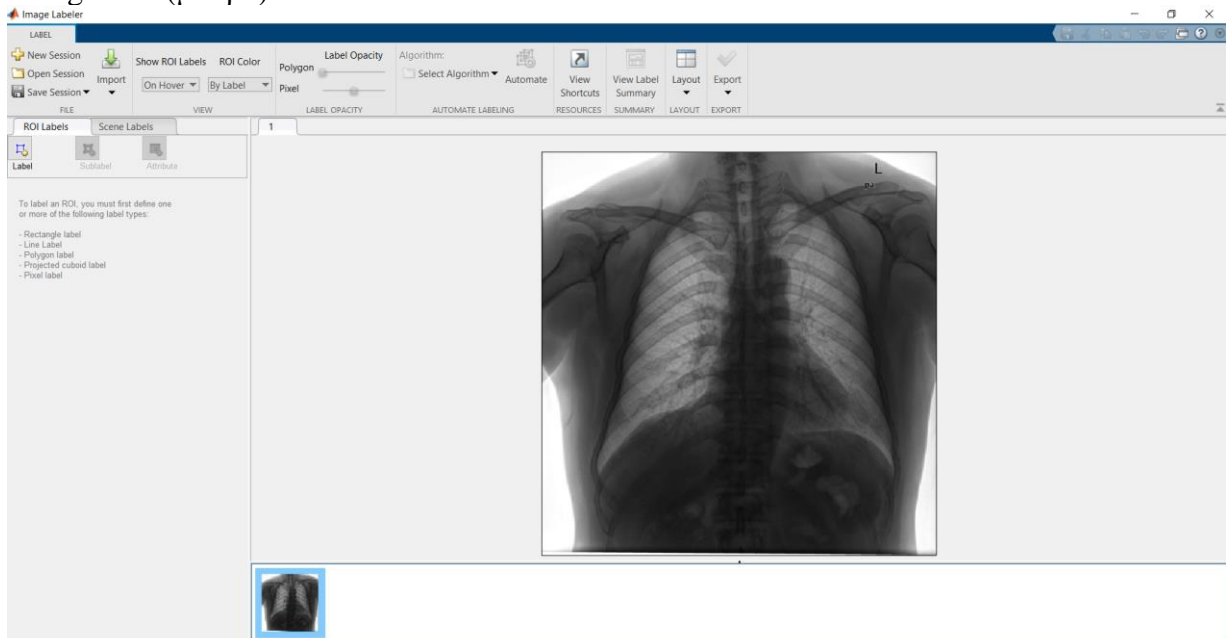
Στη μηχανική μάθηση και την κατάτμηση εικόνας (semantic segmentation) είναι απαραίτητη η προσθήκη μιας ή περισσότερων ουσιαστικών και ενημερωτικών ετικετών για την παροχή πλαισίου ώστε ένα μοντέλο μηχανικής μάθησης να μπορεί να μάθει από αυτά. Στη δική μας υλοποίηση εφαρμόζουμε σημασιολογική κατάτμηση εικόνας (semantic segmentation) και συγκεκριμένα θέλουμε να έχουμε ως αποτέλεσμα την καλύτερη δυνατή αναγνώριση 2 κλάσεων, του πνεύμονα (πρώτο class) και ότι υπάρχει στην ακτινογραφία εκτός από τον πνεύμονα (δεύτερο class). Για να το πετύχουμε αυτό θα πρέπει όλα τα δεδομένα εισόδου να

συνοδεύονται από την αντίστοιχη “μάσκα”, δηλαδή την εικόνα που περιέχει τα 2 labels και που δείχνει το ιδανικό αποτέλεσμα ουσιαστικά της κατάτμησης, βάση του οποίου το δίκτυο μαθαίνει και αργότερα σε περίπτωση που εισαχθούν σε αυτό παρόμοιες εικόνες είναι δυνατή η αυτοματοποιημένη σημασιολογική κατάτμηση των εικόνων. Παρακάτω παρατηρούμε ένα παράδειγμα μάσκας της αντίστοιχης εικόνας:

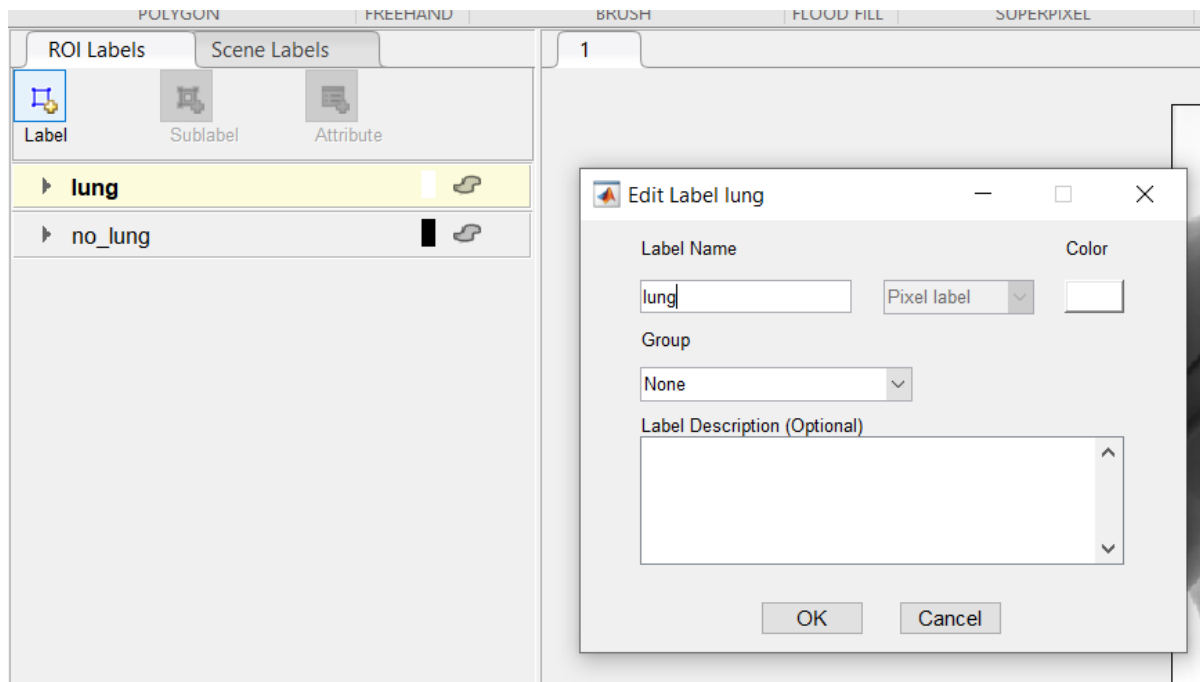


Εικόνα 8.8: Παράδειγμα μάσκας

Η διαδικασία του labeling μπορεί εύκολα να πραγματοποιηθεί με τη βοήθεια της εφαρμογής image labeler του Matlab. Αρχικά εισάγουμε την εικόνα και στη συνέχεια ορίζουμε τα labels. Για την σημασιολογική κατάτμηση επιλέγουμε pixel label και δημιουργούμε 2 labels ένα που αντιστοιχεί όπως προαναφέραμε στον πνεύμονα (άσπρο) και ένα που αντιστοιχεί στο background (μαύρο):

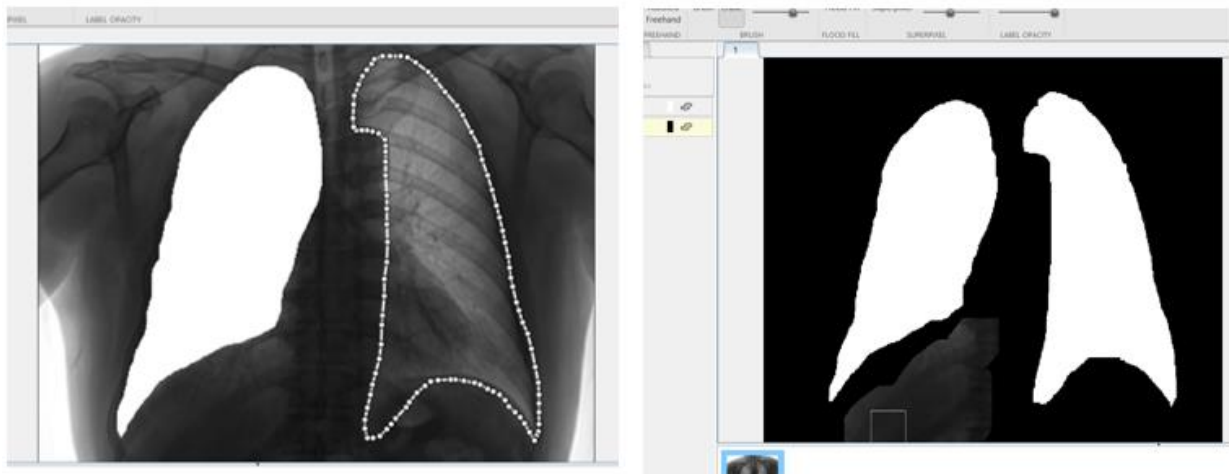


Εικόνα 8.9: Εισαγωγή εικόνας στο pixel labeler



Εικόνα 8.10: Ορισμός των labels

Έπειτα προχωράμε στο μαρκάρισμα της περιοχής την οποία αντιστοιχίζουμε στο αντίστοιχο label, αποθηκεύουμε την μάσκα που δημιουργήσαμε και την κάνουμε export στον αντίστοιχο φάκελο:



Εικόνα 8.11: Δημιουργία μάσκας

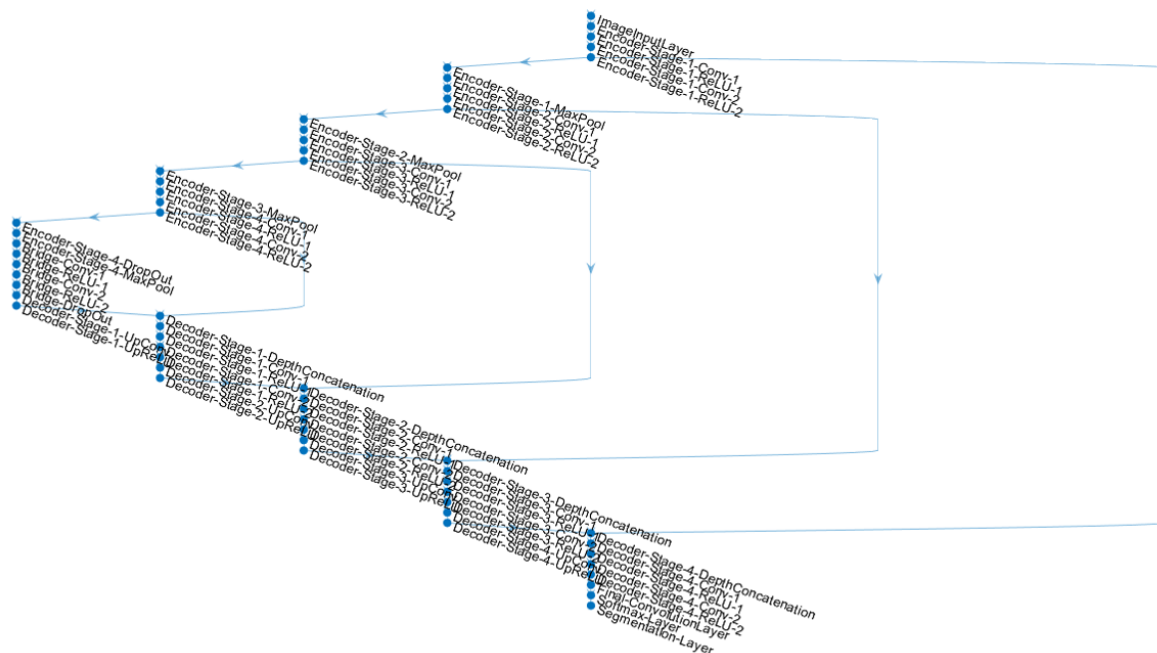
Για την εξοικονόμηση χρόνου θα χρησιμοποιήσουμε ένα έτοιμο dataset που αποτελείται από όλες τις μάσκες των αντίστοιχων ακτινογραφιών, του οποίου η άδεια χρήσης παρατίθεται στη βιβλιογραφία.

8.5 Υλοποίηση

Όπως έχει γίνει αναφορά, σκοπός του πειράματος είναι να εκπαιδύσουμε δύο διαφορετικών αρχιτεκτονικών δίκτυα δίνοντας τους ως δεδομένα τις εικόνες με τις αντίστοιχες μάσκες τους και στη συνέχεια να ελέγξουμε τη λειτουργία του εκάστοτε δικτύου όταν σε αυτό εισάγονται ως δεδομένα παρόμοιες εικόνες με αυτές βάση των οποίων έχει εκπαιδευτεί.

8.5.1 Αρχιτεκτονικές

U-net :



Εικόνα 8.12: Απεικόνιση Unet στο Matlab

Παραπάνω φαίνεται η μορφή του συνελκτικού νευρωνικού δικτύου Unet του οποίου τα 58 στρώματα παρουσιάζουμε πιο αναλυτικά παρακάτω:

58 
layers

0 
warnings

0 
errors

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
1	ImageInputLayer 496x496x1 images with 'zerocenter' normalization	Image Input	496x496x1	-
2	Encoder-Stage-1-Conv-1 64 3x3x1 convolutions with stride [1 1] and padding 'same'	Convolution	496x496x64	Weights 3x3x1x64 Bias 1x1x64
3	Encoder-Stage-1-ReLU-1 ReLU	ReLU	496x496x64	-
4	Encoder-Stage-1-Conv-2 64 3x3x64 convolutions with stride [1 1] and padding 'same'	Convolution	496x496x64	Weights 3x3x64x64 Bias 1x1x64
5	Encoder-Stage-1-ReLU-2 ReLU	ReLU	496x496x64	-
6	Encoder-Stage-1-MaxPool 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	248x248x64	-
7	Encoder-Stage-2-Conv-1 128 3x3x64 convolutions with stride [1 1] and padding 'same'	Convolution	248x248x128	Weights 3x3x64x128 Bias 1x1x128
8	Encoder-Stage-2-ReLU-1 ReLU	ReLU	248x248x128	-
9	Encoder-Stage-2-Conv-2 128 3x3x128 convolutions with stride [1 1] and padding 'same'	Convolution	248x248x128	Weights 3x3x128x128 Bias 1x1x128
10	Encoder-Stage-2-ReLU-2 ReLU	ReLU	248x248x128	-
11	Encoder-Stage-2-MaxPool 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	124x124x128	-
12	Encoder-Stage-3-Conv-1 256 3x3x128 convolutions with stride [1 1] and padding 'same'	Convolution	124x124x256	Weights 3x3x128x256 Bias 1x1x256
13	Encoder-Stage-3-ReLU-1 ReLU	ReLU	124x124x256	-
14	Encoder-Stage-3-Conv-2 256 3x3x256 convolutions with stride [1 1] and padding 'same'	Convolution	124x124x256	Weights 3x3x256x256 Bias 1x1x256
15	Encoder-Stage-3-ReLU-2 ReLU	ReLU	124x124x256	-
16	Encoder-Stage-3-MaxPool 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	62x62x256	-
17	Encoder-Stage-4-Conv-1 512 3x3x256 convolutions with stride [1 1] and padding 'same'	Convolution	62x62x512	Weights 3x3x256x512 Bias 1x1x512
18	Encoder-Stage-4-ReLU-1 ReLU	ReLU	62x62x512	-
19	Encoder-Stage-4-Conv-2 512 3x3x512 convolutions with stride [1 1] and padding 'same'	Convolution	62x62x512	Weights 3x3x512x512 Bias 1x1x512

Εικόνα 8.13: Unet layers 1-19

Ψηφιακή επεξεργασία και κατάτμηση βιοϊατρικών εικόνων με χρήση τεχνητών νευρωνικών δικτύων

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
20	Encoder-Stage-4-ReLU-2 ReLU	ReLU	62x62x512	-
21	Encoder-Stage-4-DropOut 50% dropout	Dropout	62x62x512	-
22	Encoder-Stage-4-MaxPool 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	31x31x512	-
23	Bridge-Conv-1 1024 3x3x512 convolutions with stride [1 1] and padding 'same'	Convolution	31x31x1024	Weights 3x3x512x1024 Bias 1x1x1024
24	Bridge-ReLU-1 ReLU	ReLU	31x31x1024	-
25	Bridge-Conv-2 1024 3x3x1024 convolutions with stride [1 1] and padding 'same'	Convolution	31x31x1024	Weights 3x3x1024x1024 Bias 1x1x1024
26	Bridge-ReLU-2 ReLU	ReLU	31x31x1024	-
27	Bridge-DropOut 50% dropout	Dropout	31x31x1024	-
28	Decoder-Stage-1-UpConv 512 2x2x1024 transposed convolutions with stride [2 2] and cropping [0 0 0 0]	Transposed Convolution	62x62x512	Weights 2x2x512x1024 Bias 1x1x512
29	Decoder-Stage-1-UpReLU ReLU	ReLU	62x62x512	-
30	Decoder-Stage-1-DepthConcatenation Depth concatenation of 2 inputs	Depth concatenation	62x62x1024	-
31	Decoder-Stage-1-Conv-1 512 3x3x1024 convolutions with stride [1 1] and padding 'same'	Convolution	62x62x512	Weights 3x3x1024x512 Bias 1x1x512
32	Decoder-Stage-1-ReLU-1 ReLU	ReLU	62x62x512	-
33	Decoder-Stage-1-Conv-2 512 3x3x512 convolutions with stride [1 1] and padding 'same'	Convolution	62x62x512	Weights 3x3x512x512 Bias 1x1x512
34	Decoder-Stage-1-ReLU-2 ReLU	ReLU	62x62x512	-
35	Decoder-Stage-2-UpConv 256 2x2x512 transposed convolutions with stride [2 2] and cropping [0 0 0 0]	Transposed Convolution	124x124x256	Weights 2x2x256x512 Bias 1x1x256
36	Decoder-Stage-2-UpReLU ReLU	ReLU	124x124x256	-
37	Decoder-Stage-2-DepthConcatenation Depth concatenation of 2 inputs	Depth concatenation	124x124x512	-
38	Decoder-Stage-2-Conv-1 256 3x3x512 convolutions with stride [1 1] and padding 'same'	Convolution	124x124x256	Weights 3x3x512x256 Bias 1x1x256

Εικόνα 8.14: Unet layers 20-38

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
40	Decoder-Stage-2-Conv-2 256 3x3x256 convolutions with stride [1 1] and padding 'same'	Convolution	124x124x256	Weights 3x3x256x256 Bias 1x1x256
41	Decoder-Stage-2-ReLU-2 ReLU	ReLU	124x124x256	-
42	Decoder-Stage-3-UpConv 128 2x2x256 transposed convolutions with stride [2 2] and cropping [0 0 0 0]	Transposed Convolution	248x248x128	Weights 2x2x128x256 Bias 1x1x128
43	Decoder-Stage-3-UpReLU ReLU	ReLU	248x248x128	-
44	Decoder-Stage-3-DepthConcatenation Depth concatenation of 2 inputs	Depth concatenation	248x248x256	-
45	Decoder-Stage-3-Conv-1 128 3x3x256 convolutions with stride [1 1] and padding 'same'	Convolution	248x248x128	Weights 3x3x256x128 Bias 1x1x128
46	Decoder-Stage-3-ReLU-1 ReLU	ReLU	248x248x128	-
47	Decoder-Stage-3-Conv-2 128 3x3x128 convolutions with stride [1 1] and padding 'same'	Convolution	248x248x128	Weights 3x3x128x128 Bias 1x1x128
48	Decoder-Stage-3-ReLU-2 ReLU	ReLU	248x248x128	-
49	Decoder-Stage-4-UpConv 64 2x2x128 transposed convolutions with stride [2 2] and cropping [0 0 0 0]	Transposed Convolution	496x496x64	Weights 2x2x64x128 Bias 1x1x64
50	Decoder-Stage-4-UpReLU ReLU	ReLU	496x496x64	-
51	Decoder-Stage-4-DepthConcatenation Depth concatenation of 2 inputs	Depth concatenation	496x496x128	-
52	Decoder-Stage-4-Conv-1 64 3x3x128 convolutions with stride [1 1] and padding 'same'	Convolution	496x496x64	Weights 3x3x128x64 Bias 1x1x64
53	Decoder-Stage-4-ReLU-1 ReLU	ReLU	496x496x64	-
54	Decoder-Stage-4-Conv-2 64 3x3x64 convolutions with stride [1 1] and padding 'same'	Convolution	496x496x64	Weights 3x3x64x64 Bias 1x1x64
55	Decoder-Stage-4-ReLU-2 ReLU	ReLU	496x496x64	-
56	Final-ConvolutionLayer 2 1x1x64 convolutions with stride [1 1] and padding 'same'	Convolution	496x496x2	Weights 1x1x64x2 Bias 1x1x2
57	Softmax-Layer softmax	Softmax	496x496x2	-
58	Segmentation-Layer Cross-entropy loss with classes 'LUNGS' and 'OTHERS'	Pixel Classification ...	496x496x2	-

Εικόνα 8.15: Unet layers 40-58

- Με μια πρώτη ματιά παρατηρούμε όπως περιμέναμε μία συμμετρική μορφή με ίσα στάδια κωδικοποίησης (encoder stages) και αποκωδικοποίησης (decoder stages). Επίσης, όπως φαίνεται στο input layer όλες οι εικόνες έχουν γίνει resized σε διαστάσεις 496x496 και μεταχειρίζονται ως ένας πίνακας (496x496x1).
- Το μονοπάτι της κωδικοποίησης ακολουθεί την εξής φόρμουλα:

Convolutional layer 1 → Convolutional layer 2 → Max-pool layer

- Στο encoder stage 1 παρατηρούμε βάθος εξόδου x64 από x1 που ήταν, καθώς εφαρμόστηκαν 64 φίλτρα kernel 3x3, με τις διαστάσεις όμως να παραμένουν ίδιες (496x496), αφού χρησιμοποιήθηκε padding = "same", δηλαδή προστέθηκαν επιπλέον εικονοστοιχεία στις άκρες της εικόνας, έτσι ώστε να έχει αυτή ίδιες διαστάσεις με αυτές που είχε στο προηγούμενο στρώμα.
- Κάθε encoder stage κλείνει με ένα max-pool layer, το οποίο υποδιπλασιάζει τις διαστάσεις της εικόνας, εφαρμόζοντας έναν πίνακα 2x2, επιλέγοντας σε αυτόν τις μέγιστες τιμές και αντικαθιστώντας τον πίνακα με αυτές.
- Η διαδικασία αυτή επαναλαμβάνεται τρεις ακόμα φορές με την έξοδό μας τελικά να έχει διαστάσεις 31x31x1024.
- Εδώ ξεκινάει το μονοπάτι της αποκωδικοποίησης, το οποίο ακολουθεί τη φόρμουλα:

Transposed convolution → Concatenation → Convolutional layer 1 → Convolutional layer 2

- Η transposed συνέλιξη είναι μία τεχνική up-sampling που επεκτείνει το μέγεθος των εικόνων. Ουσιαστικά κάνει κάποιο padding (συμπλήρωση) στην αρχική εικόνα και ακολουθείται από μία λειτουργία συνέλιξης.
- Μετά την transposed συνέλιξη, η εικόνα διπλασιάζεται σε μέγεθος (από 124x124x256 σε 248x248x128) και στη συνέχεια αυτή η εικόνα συνενώνεται με την αντίστοιχη εικόνα από τη διαδρομή συστολής, δημιουργώντας μία εικόνα 248x248x256. Αυτό γίνεται για να έχουμε μία πιο ακριβή πρόβλεψη, συνδυάζοντας πληροφορίες από προηγούμενα επίπεδα.
- Ακολουθούν στη συνέχεια δύο επιπλέον συνελίξεις και όλη η διαδικασία επαναλαμβάνεται τρεις ακόμα φορές.
- Τέλος, η εικόνα έρχεται στις επιθυμητές διαστάσεις και τα εικονοστοιχεία αντιστοιχίζονται στην αντίστοιχη κλάση (pixel classification).

Απλό CNN:



Εικόνα 8.16: Απεικόνιση απλού CNN στο Matlab

Παραπάνω φαίνεται η μορφή του απλού συνελκτικού νευρωνικού που κατασκευάσαμε, του οποίου τα 11 στρώματα παρουσιάζουμε πιο αναλυτικά παρακάτω:

Ψηφιακή επεξεργασία και κατάτμηση βιοϊατρικών εικόνων με χρήση τεχνητών νευρωνικών δικτύων

ANALYSIS RESULT				
	Name	Type	Activations	Learnables
1	imageinput 496x496x1 images with 'zerocenter' normalization	Image Input	496x496x1	-
2	conv_1 128 3x3 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	496x496x128	Weights 3x3x1x128 Bias 1x1x128
3	relu_1 ReLU	ReLU	496x496x128	-
4	maxpool 2x2 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	248x248x128	-
5	conv_2 128 3x3 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	248x248x128	Weights 3x3x128x128 Bias 1x1x128
6	relu_2 ReLU	ReLU	248x248x128	-
7	transposed-conv 128 4x4 transposed convolutions with stride [2 2] and cropping [1 1 1 1]	Transposed Convol...	496x496x128	Weights 4x4x128x128 Bias 1x1x128
8	conv_3 128 3x3 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	496x496x128	Weights 3x3x128x128 Bias 1x1x128
9	conv_4 2 1x1 convolutions with stride [1 1] and padding [0 0 0 0]	Convolution	496x496x2	Weights 1x1x128x2 Bias 1x1x2
10	softmax softmax	Softmax	496x496x2	-
11	classoutput Cross-entropy loss	Pixel Classification ...	496x496x2	-

Εικόνα 8.17: Layers απλού CNN

- Στην περίπτωση του απλού συνελκτικού δικτύου, παρατηρούμε μία πολύ απλούστερη αρχιτεκτονική, αποτελούμενη από 11 στρώματα. Εδώ δεν έχουμε συγκεκριμένες προδιαγραφές για τις διαστάσεις των αρχικών δεδομένων, οπότε χρησιμοποιούμε τις ίδιες διαστάσεις με πριν.

- Η φόρμουλα που ακολουθείται εδώ είναι η εξής :

Convolutional layer 1 → Max-pool layer → Convolutional layer 2 → Transposed convolution

→ Convolutional layer 3 → Convolutional layer 4

- Βλέπουμε ότι στην αρχική εικόνα εφαρμόζονται 128 φίλτρα kernel 3x3 στο πρώτο στρώμα συνέλιξης
- Το επόμενο στρώμα είναι ένα max-pool layer, το οποίο υποδιπλασιάζει τις διαστάσεις της εικόνας, εφαρμόζοντας έναν πίνακα 2x2, επιλέγοντας σε αυτόν τις μέγιστες τιμές και αντικαθιστώντας τον πίνακα με αυτές.
- Ακολουθεί το δεύτερο convolutional layer και στη συνέχεια το transposed convolution διπλασιάζει τις διαστάσεις της εικόνας του προηγούμενου στρώματος κρατώντας το βάθος σταθερό
- Ακολουθούν στη συνέχεια δύο επιπλέον συνέλιξεις και τέλος η εικόνα έρχεται στις επιθυμητές διαστάσεις και τα εικονοστοιχεία αντιστοιχίζονται στην αντίστοιχη κλάση (pixel classification).

8.5.2 Εκπαίδευση του Unet

Θα εκπαιδύσουμε το δίκτυο μας δύο φορές χρησιμοποιώντας αρχικά mini-batch size ίσο με 1 και στη δεύτερη περίπτωση mini-batch size ίσο με 3.

Εκπαίδευση με mini-batch size = 1

Τα βήματα που ακολουθήσαμε για την πραγματοποίηση της εκπαίδευσης του δικτύου Unet είναι τα παρακάτω:

Βήμα 1^ο: Γίνεται εισαγωγή ενός μέρους του dataset στο Matlab και συγκεκριμένα 546 εικόνων από τις 566 εικόνες που διαθέτουμε συνολικά , καθώς οι υπόλοιπες 20 εικόνες θα χρησιμοποιηθούν στον έλεγχο των δικτύων ως άγνωστα δεδομένα.

Βήμα 2^ο: Γίνεται εισαγωγή των αντίστοιχων 546 μασκών στο Matlab.

Βήμα 3^ο: Πραγματοποιείται καθορισμός των κλάσεων και των αντίστοιχων τιμών pixel.

Βήμα 4^ο: Φορτώνεται στο Matlab το δίκτυο Unet

Βήμα 5^ο: Ορίζονται οι παράμετροι της εκμάθησης και ειδικότερα:

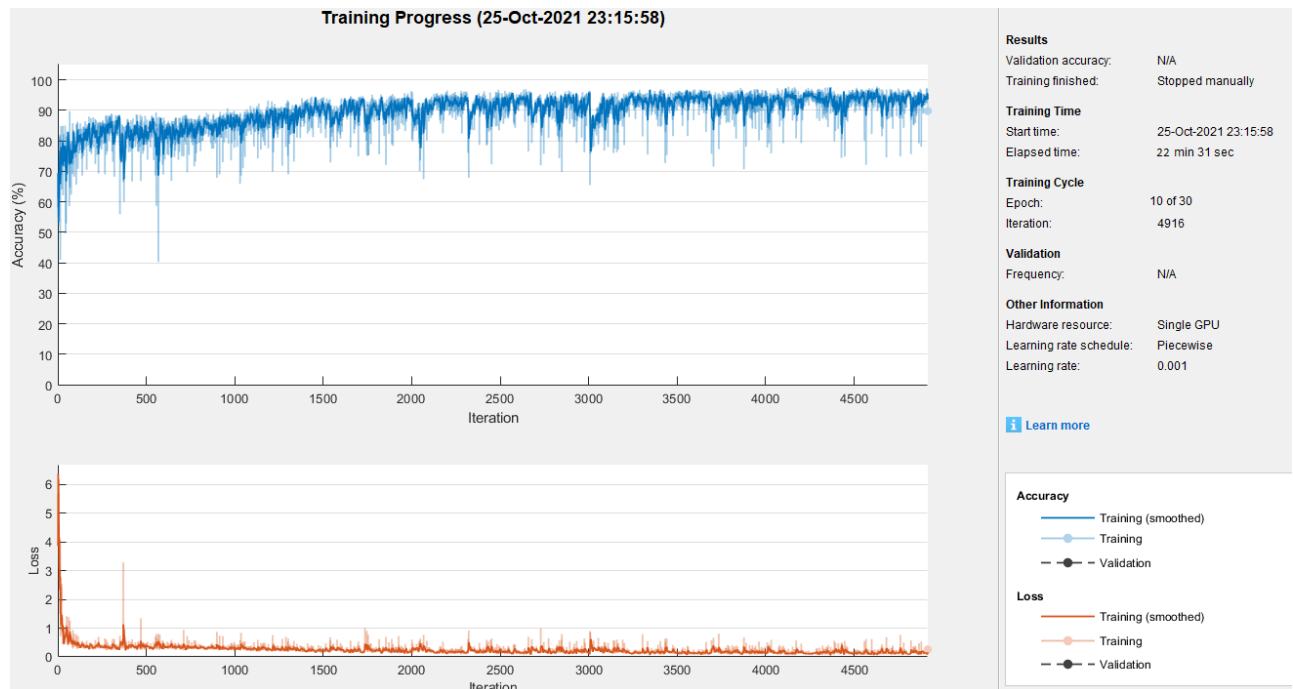
- Ορίζεται αλγόριθμος βελτιστοποίησης (optimizer) “adam”, οποίος έχει αναδειχθεί ως ο καταλληλότερος αλγόριθμος βελτιστοποίησης στις περιπτώσεις εκπαίδευσης CNN και προσαρμόζει τα χαρακτηριστικά του νευρωνικού δικτύου, όπως το ρυθμό μάθησης και τα βάρη. Ως εκ τούτου, βοηθά στη βελτίωση της ακρίβειας και μειώνει τη συνολική απώλεια.
- Ορίζεται ρυθμός μάθησης ίσος με 0.001.
- Επιλέγεται ότι μετά από κάθε 2 epochs ο ρυθμός εκπαίδευσης θα υποπενταπλασιάζεται. Γενικά επιλέγεται αυτή η τεχνική , καθώς έχει παρατηρηθεί ότι έτσι το δίκτυο μαθαίνει πιο ομαλά.
- Ορίζεται μέγιστος αριθμός περασμάτων του dataset (epochs) ίσος με 30.
- Ορίζεται η τιμή 1 για τον παράγοντα mini-batch.

Βήμα 6^ο: Προετοιμάζονται τα δεδομένα εκπαίδευσης .

Βήμα 7^ο: Ξεκινάει η εκπαίδευση του δικτύου με βάση τους παράγοντες που ορίσαμε στο βήμα 5^ο.

Βήμα 8^ο: Τερματίζουμε την εκπαίδευση περίπου στις 10 epochs ,όπως θα κάνουμε σε όλες τις περιπτώσεις εκπαίδευσης για να έχουμε ένα κοινό σημείο αναφοράς και για εξοικονόμηση χρόνου.

Παρακάτω παρουσιάζονται οι καμπύλες της ακρίβειας (accuracy) και του σφάλματος (loss) αντίστοιχα που προκύπτουν κατά την εκπαίδευση του Unet με χρήση mini-batch size 1:



Εικόνα 8.18: Accuracy και loss εκπαίδευσης Unet με mini-batch size=1

- Παρατηρούμε ότι για να εκπαιδύσουμε το δίκτυο για 10 epochs απαιτήθηκαν περίπου 22 λεπτά. Ο χρόνος αυτός αλλάζει ανάλογα με τον υπολογιστή που αναλαμβάνει την εκπαίδευση, αλλά είναι λογικό να απαιτείται σχετικά μεγάλο χρονικό διάστημα για την εκπαίδευση όταν χρησιμοποιούμε mini-batch size ίσο με 1, αφού σε κάθε iteration (βήμα) περνάει από τον αλγόριθμο ένα μόνο δείγμα του dataset. Συνεπώς για την ολοκλήρωση ενός epoch θα πραγματοποιηθούν 546 iterations, όσα τα δείγματά μας.
- Βλέπουμε ότι αρχικά η ακρίβεια (accuracy) είναι αρκετά χαμηλή και αντίστοιχα το σφάλμα (loss) είναι υψηλό. Στο ακόλουθο στιγμιότυπο από το command window φαίνεται ότι η εκπαίδευση ξεκινάει αρκετά ασταθής με ένα προσεγγιστικό μέσο όρο της ακρίβειας της τάξης του 70% και αντίστοιχα με ένα προσεγγιστικό μέσο όρο του σφάλματος της τάξης περίπου ίσο με 30%.

New to MATLAB? See resources for [Getting Started](#).

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:04	77.55%	2.6969	0.0010
1	2	00:00:05	29.53%	11.0773	0.0010
1	4	00:00:05	62.74%	4.4144	0.0010
1	6	00:00:06	74.00%	4.0535	0.0010
1	8	00:00:06	81.63%	2.8593	0.0010
1	10	00:00:07	67.09%	5.0717	0.0010
1	12	00:00:07	80.39%	2.7832	0.0010
1	14	00:00:08	68.58%	4.3512	0.0010
1	16	00:00:08	72.63%	3.4980	0.0010
1	18	00:00:09	72.56%	2.0735	0.0010
1	20	00:00:09	74.01%	0.9950	0.0010
1	22	00:00:09	66.97%	0.9307	0.0010
1	24	00:00:10	86.58%	0.5677	0.0010
1	26	00:00:10	77.25%	1.4107	0.0010
1	28	00:00:11	83.04%	0.7378	0.0010
1	30	00:00:11	79.07%	0.7732	0.0010
1	32	00:00:12	72.76%	0.6154	0.0010
1	34	00:00:12	71.95%	0.8300	0.0010
1	36	00:00:13	72.72%	0.6428	0.0010
1	38	00:00:13	81.76%	0.4829	0.0010
1	40	00:00:13	71.57%	0.6914	0.0010
1	42	00:00:14	46.34%	0.9456	0.0010
1	44	00:00:14	76.13%	0.9702	0.0010
1	46	00:00:15	82.30%	0.7631	0.0010

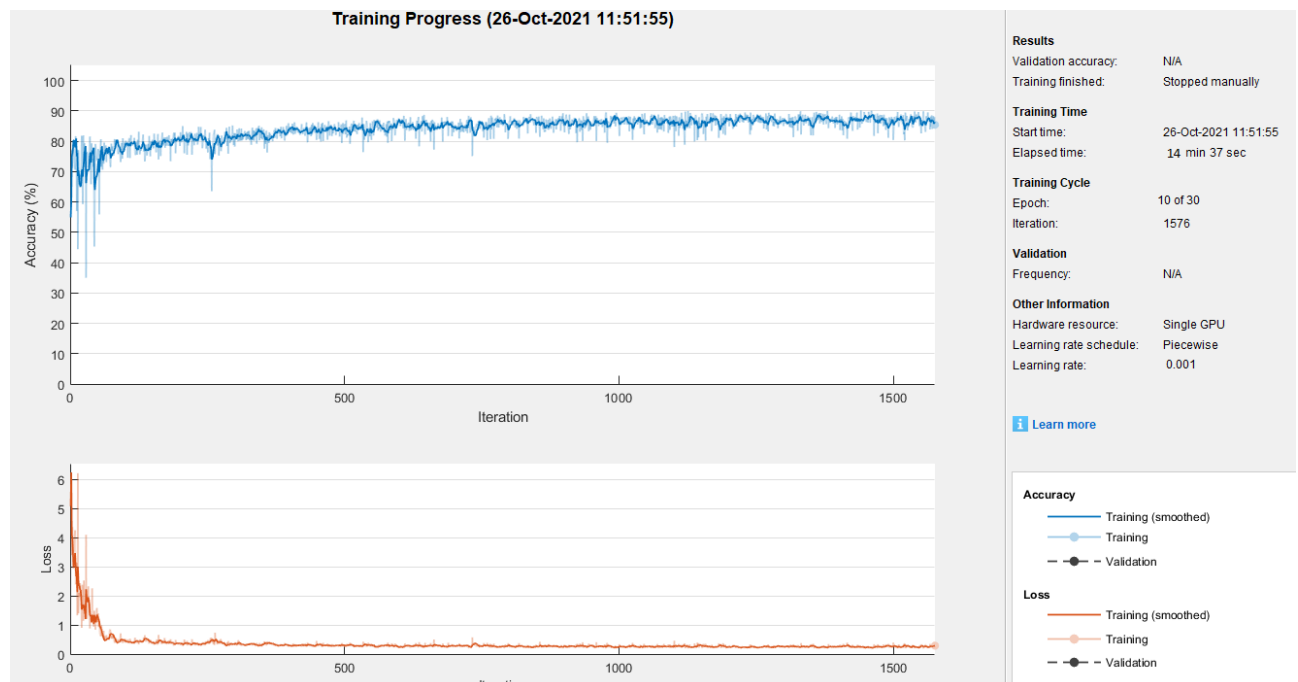
Εικόνα 8.19: Αρχικά iterations εκπαίδευσης Unet με mini-batch size=1

- Μετά από περίπου 2000 iterations, δηλαδή περίπου 4 epochs παρατηρούμε ότι η ακρίβεια σταθεροποιείται πάνω από το 90% και στο τέλος της εκπαίδευσης να προσεγγίζει το 96%. Αντίστοιχα η καμπύλη του σφάλματος παρουσιάζει σταθεροποίηση στα ίδια χρονικά σημεία και στο τέλος της εκπαίδευσης προσεγγίζει τιμές κοντά στο 1-2%.
- Επισημαίνεται ότι η διάρκεια της εκπαίδευσης για κάθε epoch δεν είναι ίδια αφού χρησιμοποιούμε μειούμενο ρυθμό εκπαίδευσης για να προσδώσουμε μεγαλύτερη ακρίβεια και ομαλότητα στο δίκτυο.

Εκπαίδευση με mini-batch size = 3

Τα ίδια βήματα με παραπάνω θα ακολουθηθούν και για την εκπαίδευση του ίδιου δικτύου (Unet) με την μόνη αλλαγή, το διαφορετικό μέγεθος του mini-batch. Στην περίπτωση αυτή λοιπόν ορίζεται η τιμή 3 για τον παράγοντα mini-batch.

Παρακάτω παρουσιάζονται οι καμπύλες της ακρίβειας (accuracy) και του σφάλματος (loss) αντίστοιχα που προκύπτουν κατά την εκπαίδευση του Unet με χρήση mini-batch size 3:



Εικόνα 8.20: Accuracy και loss εκπαίδευσης Unet με mini-batch size=3

- Παρατηρούμε ότι για να εκπαιδύσουμε το δίκτυο για 10 epochs απαιτήθηκαν περίπου 14 λεπτά. Εδώ ουσιαστικά σε κάθε iteration, 3 τυχαία δείγματα περνάνε από τον αλγόριθμο και κατά συνέπεια χρησιμοποιούνται μόνο 182 iterations για την ολοκλήρωση ενός epoch. Ως επακόλουθο, η εκπαίδευση πραγματοποιείται γρηγορότερα, όπως είναι και λογικό.
- Βλέπουμε ότι αρχικά η ακρίβεια (accuracy) είναι και σε αυτή την περίπτωση αρκετά χαμηλή και αντίστοιχα το σφάλμα (loss) είναι υψηλό. Στο ακόλουθο στιγμιότυπο από το command window φαίνεται ότι η εκπαίδευση ξεκινάει με μεγάλες αυξομειώσεις της ακρίβειας, η οποία κυμαίνεται σε παρόμοιες τιμές με την προηγούμενη περίπτωση, όπως και το σφάλμα.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:04	51.53%	5.2255	0.0010
1	2	00:00:04	70.00%	4.7135	0.0010
1	4	00:00:06	55.68%	5.7088	0.0010
1	6	00:00:07	64.09%	3.0874	0.0010
1	8	00:00:08	77.16%	2.0374	0.0010
1	10	00:00:09	66.83%	1.2961	0.0010
1	12	00:00:10	70.96%	1.2819	0.0010
1	14	00:00:11	82.12%	0.7102	0.0010
1	16	00:00:12	74.47%	0.7534	0.0010
1	18	00:00:13	82.85%	0.6450	0.0010
1	20	00:00:14	67.46%	1.1607	0.0010
1	22	00:00:15	78.91%	1.0765	0.0010
1	24	00:00:16	74.47%	1.4732	0.0010
1	26	00:00:18	57.94%	1.2466	0.0010
1	28	00:00:19	79.40%	0.9681	0.0010
1	30	00:00:20	54.16%	2.1392	0.0010
1	32	00:00:21	77.03%	1.2666	0.0010
1	34	00:00:22	76.59%	1.6213	0.0010
1	36	00:00:23	65.48%	1.8365	0.0010
1	38	00:00:24	81.72%	0.8191	0.0010
1	40	00:00:25	56.63%	1.7702	0.0010
1	42	00:00:26	67.95%	0.9074	0.0010
1	44	00:00:27	76.51%	1.0490	0.0010
1	46	00:00:28	80.76%	0.7831	0.0010

Εικόνα 8.21: Αρχικά iterations εκπαίδευσης Unet με mini-batch size=3

- Μετά από περίπου 500 iterations ,δηλαδή περίπου 3 epochs παρατηρούμε ότι η ακρίβεια σταθεροποιείται πάνω από το 80% και στο τέλος της εκπαίδευσης να προσεγγίζει το 85%. Αντίστοιχα η καμπύλη του σφάλματος παρουσιάζει σταθεροποίηση στα ίδια χρονικά σημεία και στο τέλος της εκπαίδευσης προσεγγίζει τιμές κοντά στο 2-3%.

8.5.3 Έλεγχος του Unet με γνωστά δεδομένα

Θα παρατηρήσουμε στο σημείο αυτό, κάποια οπτικά αποτελέσματα κατάτμησης της περιοχής των πνευμόνων, που προέκυψαν από το τεστ του δικτύου Unet σε 20 τυχαίες ακτινογραφίες του ίδιου του training dataset (γνωστά δεδομένα) για κάθε μία από τις δύο περιπτώσεις mini-batch size. Ακόμα θα δείξουμε και ποσοτικά την ακρίβεια του δικτύου.

Αποτελέσματα εκπαίδευσης με mini-batch size = 1

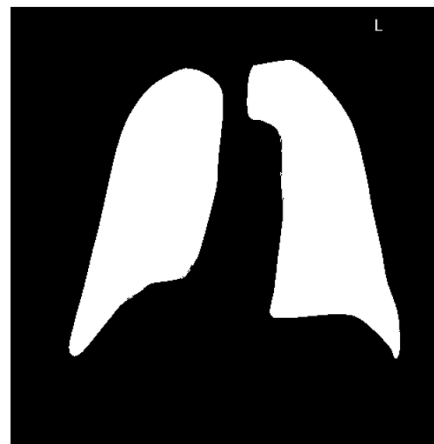
Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα γνωστού δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.22: Γνωστό δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.23: Αποτέλεσμα κατάτμησης



Εικόνα 8.24: Ιδανικό αποτέλεσμα

Για την ποσοτική έκφραση της ακρίβειας της κατάτμησης θα εφαρμόσουμε, για ένα σύνολο 20 αποτελεσμάτων, μία συνάρτηση XOR μεταξύ κάθε εικονοστοιχείου της εικόνας εξόδου και του κάθε αντίστοιχου εικονοστοιχείου της αντίστοιχης μάσκας (ιδανικού αποτελέσματος):

- Γίνεται εισαγωγή στο Matlab των 20 αποτελεσμάτων.
- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((\text{xor}(a, a2) + \text{xor}(b, b2) + \dots + \text{xor}(u, u2)) / 20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = \text{sum}(\text{sum}(z)) = 6626$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων, υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z, 1) * \text{size}(z, 2)) = 0,0269 \text{ ή } 2,69\%$$

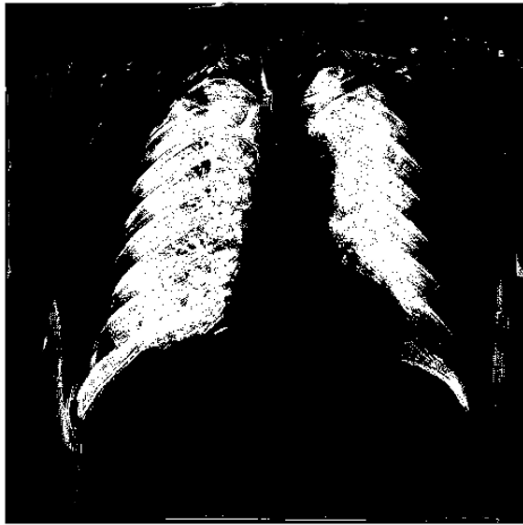
Προκύπτει λοιπόν ότι το δίκτυο Unet, όταν εκπαιδευτεί με παράγοντα mini-batch = 1 και ελεγχτεί σε γνωστά δεδομένα, παρουσιάζει σφάλμα ίσο με 2,69%

Αποτελέσματα εκπαίδευσης με mini-batch size = 3

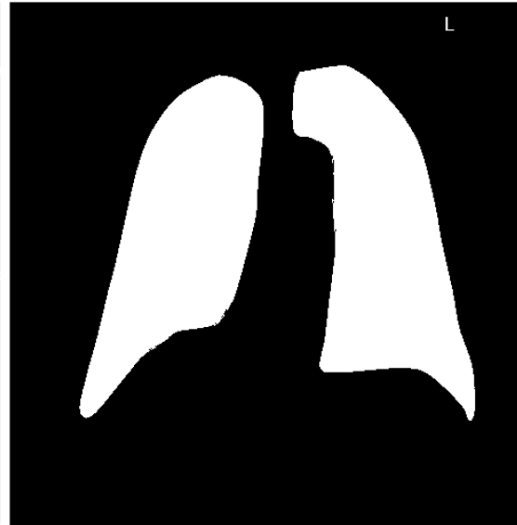
Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα γνωστού δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.25: Γνωστό δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.26: Αποτέλεσμα κατάτμησης



Εικόνα 8.27: Ιδανικό αποτέλεσμα

Παρατηρούμε ότι στην περίπτωση αυτή το αποτέλεσμα της κατάτμησης εμφανίζεται λιγότερο ακριβές.

Για τον αριθμητικό υπολογισμό του σφάλματος , εφαρμόζουμε την ίδια μέθοδο με προηγουμένως:

- Γίνεται εισαγωγή στο Matlab των 20 αντίστοιχων αποτελεσμάτων.
- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((xor(a, a2) + xor(b, b2) + + xor(u, u2)) / 20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = \text{sum}(\text{sum}(z)) = 11707$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων , υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z, 1) * \text{size}(z, 2)) = 0,0476 \text{ ή } 4,76\%$$

Προκύπτει λοιπόν ότι το δίκτυο Unet , όταν εκπαιδευτεί με παράγοντα mini-batch = 3 και ελεγχτεί σε γνωστά δεδομένα , παρουσιάζει σφάλμα ίσο με 4,76%

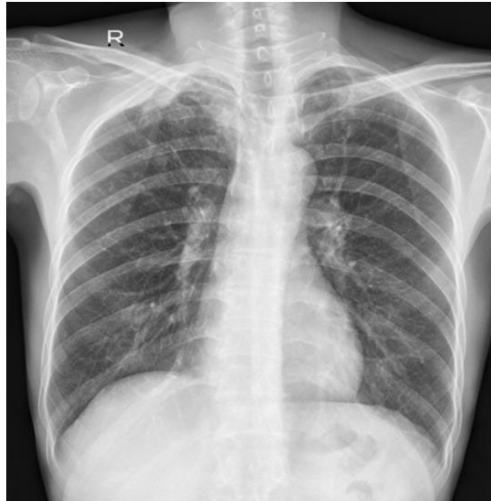
8.5.4 Έλεγχος του Unet με άγνωστα δεδομένα

Θα παραθέσουμε στο σημείο αυτό, κάποια οπτικά αποτελέσματα κατάτμησης της περιοχής των πνευμόνων, που προέκυψαν από το τεστ του δικτύου Unet σε 20 τυχαίες ακτινογραφίες που δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση του δικτύου (άγνωστα δεδομένα) για κάθε

μία από τις δύο περιπτώσεις mini-batch size. Ακόμα θα δείξουμε και ποσοτικά την ακρίβεια του δικτύου.

Αποτελέσματα εκπαίδευσης με mini-batch size = 1

Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα άγνωστου δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.28: Άγνωστο δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.29: Αποτέλεσμα κατάτμησης



Εικόνα 8.30: Ιδανικό αποτέλεσμα

Παρακάτω υπολογίζεται, με την γνωστή φόρμουλα, το σφάλμα που προκύπτει από την κατάτμηση:

- Γίνεται εισαγωγή στο Matlab των 20 αποτελεσμάτων της κατάτμησης άγνωστων δεδομένων.

- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((\text{xor}(a, a2) + \text{xor}(b, b2) + \dots + \text{xor}(u, u2)) / 20) ;$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = \text{sum}(\text{sum}(z)) = 12599$$

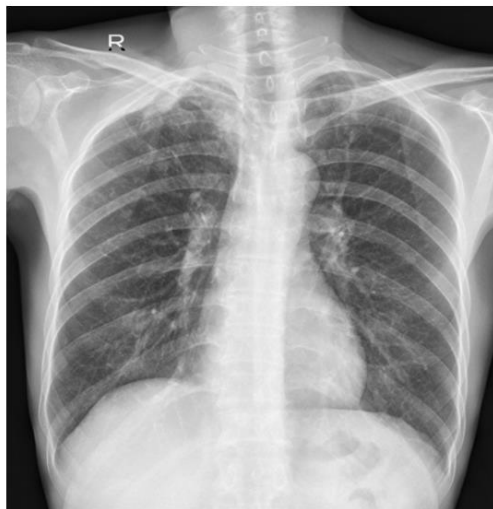
- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων, υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z, 1) * \text{size}(z, 2)) = 0,0512 \text{ ή } 5,12\%$$

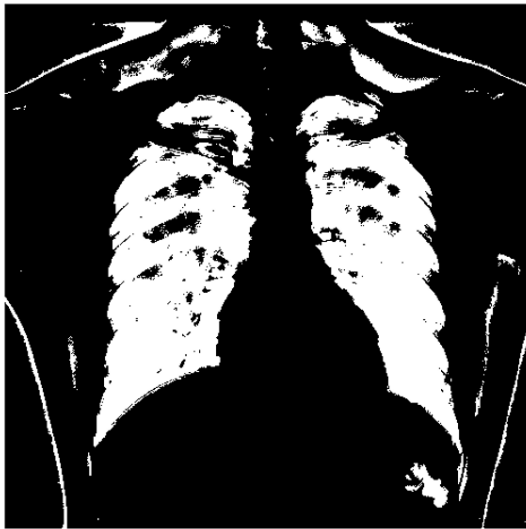
Προκύπτει λοιπόν ότι το δίκτυο Unet, όταν εκπαιδευτεί με παράγοντα mini-batch = 1 και ελεγχτεί σε άγνωστα δεδομένα, παρουσιάζει σφάλμα ίσο με 5,12%

Αποτελέσματα εκπαίδευσης με mini-batch size = 3

Παρουσιάζεται παρακάτω το ίδιο χαρακτηριστικό παράδειγμα άγνωστου δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.31: Άγνωστο δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.32 Αποτέλεσμα κατάτμησης



Εικόνα 8.33: Ιδανικό Αποτέλεσμα

Ακολουθεί ο υπολογισμός του σφάλματος :

- Γίνεται εισαγωγή στο Matlab των 20 αποτελεσμάτων της κατάτμησης άγνωστων δεδομένων.
- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((xor(a,a2) + xor(b,b2) + + xor(u,u2))/20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = \text{sum}(\text{sum}(z)) = 16723$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων, υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z,1) * \text{size}(z,2)) = 0,068 \text{ ή } 6,8\%$$

Προκύπτει λοιπόν ότι το δίκτυο Unet, όταν εκπαιδευτεί με παράγοντα mini-batch = 3 και ελεγχτεί σε άγνωστα δεδομένα, παρουσιάζει σφάλμα ίσο με 6,8%

Γίνεται αντιληπτό ότι με την αύξηση του mini-batch size, η κατάτμηση χειροτερεύει. Επίσης όπως είναι λογικό, έχουμε μεγαλύτερο σφάλμα κατά την κατάτμηση άγνωστων δεδομένων.

8.5.5 Εκπαίδευση του απλού CNN

Θα εκπαιδεύσουμε το δίκτυο μας δύο φορές χρησιμοποιώντας αρχικά mini-batch size ίσο με 1 και στη δεύτερη περίπτωση mini-batch size ίσο με 3, όπως πράξαμε και για το δίκτυο Unet.

Εκπαίδευση με mini-batch size = 1

Τα βήματα που ακολουθήσαμε για την πραγματοποίηση της εκπαίδευσης του απλού CNN είναι τα παρακάτω:

Βήμα 1^ο: Γίνεται εισαγωγή ενός μέρους του dataset στο Matlab και συγκεκριμένα 546 εικόνων από τις 566 εικόνες που διαθέτουμε συνολικά, καθώς οι υπόλοιπες 20 εικόνες θα χρησιμοποιηθούν στον έλεγχο των δικτύων ως άγνωστα δεδομένα.

Βήμα 2^ο: Γίνεται εισαγωγή των αντίστοιχων 546 μασκών στο Matlab.

Βήμα 3^ο: Πραγματοποιείται καθορισμός των κλάσεων και των αντίστοιχων τιμών pixel.

Βήμα 4^ο: Εδώ, σε αντίθεση με την περίπτωση του Unet, εισάγουμε την επιθυμητή αρχιτεκτονική του δικτύου μας στο Matlab, με τον αντίστοιχο κώδικα:

```
38 - numFilters = 128;
39 - filterSize = 3;
40 - numClasses = 2;
41 - layers = [
42     imageInputLayer([r c 1])
43     convolution2dLayer(filterSize,numFilters,'Padding',1)
44     reluLayer()|
45     maxPooling2dLayer(2,'Stride',2)
46     convolution2dLayer(filterSize,numFilters,'Padding',1)
47     reluLayer()
48     transposedConv2dLayer(4,numFilters,'Stride',2,'Cropping',1);
49     convolution2dLayer(filterSize,numFilters,'Padding',1)
50     convolution2dLayer(1,numClasses);
51     softmaxLayer()
52     pixelClassificationLayer()
53 ];
```

Εικόνα 8.34: Εισαγωγή αρχιτεκτονικής απλού CNN στο Matlab

Βήμα 5^ο: Ορίζονται οι παράμετροι της εκμάθησης και ειδικότερα:

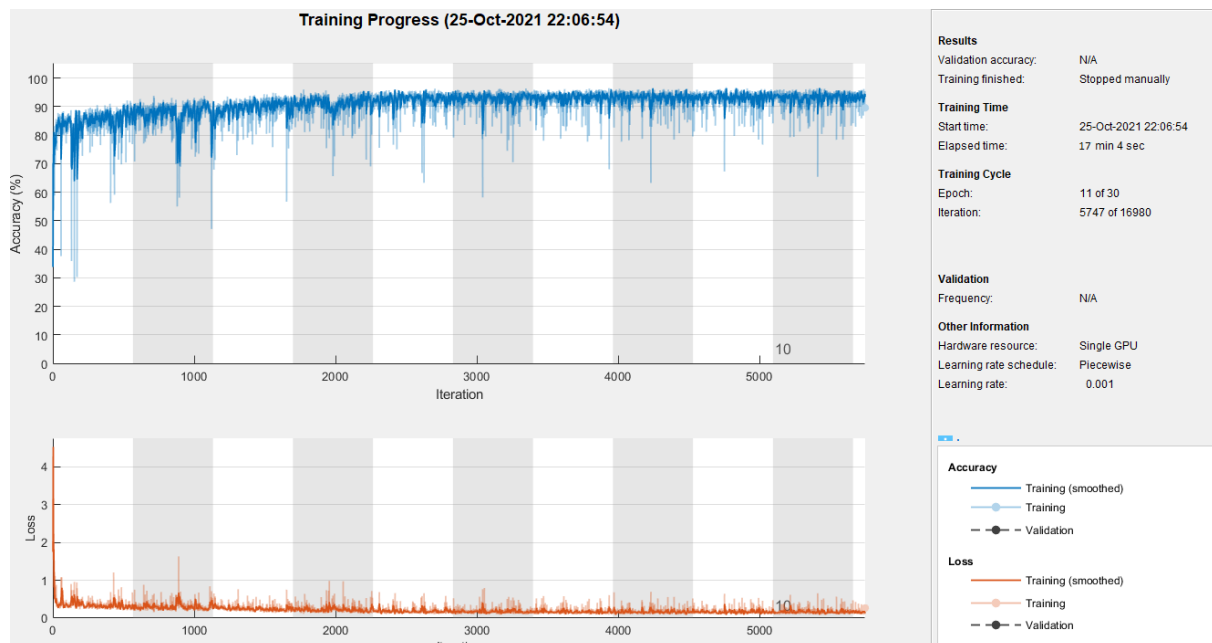
- Ορίζεται αλγόριθμος βελτιστοποίησης (optimizer) “adam”.
- Ορίζεται ρυθμός μάθησης ίσος με 0.001.
- Επιλέγεται ότι μετά από κάθε 2 epochs ο ρυθμός εκπαίδευσης θα υποπενταπλασιάζεται. Γενικά επιλέγεται αυτή η τεχνική, καθώς έχει παρατηρηθεί ότι έτσι το δίκτυο μαθαίνει πιο ομαλά.
- Ορίζεται μέγιστος αριθμός περασμάτων του dataset (epochs) ίσος με 30.
- Ορίζεται η τιμή 1 για τον παράγοντα mini-batch.

Βήμα 6^ο: Προετοιμάζονται τα δεδομένα εκπαίδευσης .

Βήμα 7^ο: Ξεκινάει η εκπαίδευση του δικτύου με βάση τους παράγοντες που ορίσαμε στο βήμα 5^ο.

Βήμα 8^ο : Τερματίζουμε την εκπαίδευση περίπου στις 10 epochs ,όπως κάνουμε σε όλες τις περιπτώσεις εκπαίδευσης για να έχουμε ένα κοινό σημείο αναφοράς και για εξοικονόμηση χρόνου.

Παρακάτω παρουσιάζονται οι καμπύλες της ακρίβειας (accuracy) και του σφάλματος (loss) αντίστοιχα που προκύπτουν κατά την εκπαίδευση του απλού CNN με χρήση mini-batch size 1:



Εικόνα 8.35: Accuracy και loss εκπαίδευσης απλού CNN με mini-batch size=1

- Παρατηρούμε ότι για να εκπαιδύσουμε το δίκτυο για 10 epochs απαιτήθηκαν περίπου 17 λεπτά. Ο χρόνος εκπαίδευσης σε αυτή την περίπτωση είναι ελαφρώς μειωμένος σε σχέση με την περίπτωση του Unet, κάτι που είναι λογικό, καθώς η αρχιτεκτονική που χρησιμοποιούμε εδώ είναι σημαντικά απλούστερη. Ωστόσο η διαφορά στην χρονική διάρκεια εκπαίδευσης δεν είναι μεγάλη.
- Μετά από περίπου 2500 iterations, δηλαδή περίπου 5 epochs παρατηρούμε ότι η ακρίβεια σταθεροποιείται πάνω από το 90% και στο τέλος της εκπαίδευσης να προσεγγίζει το 95%. Αντίστοιχα η καμπύλη του σφάλματος παρουσιάζει σταθεροποίηση στα ίδια χρονικά σημεία και στο τέλος της εκπαίδευσης προσεγγίζει τιμές κοντά στο 2-3%.
- Βλέπουμε ότι αρχικά η ακρίβεια (accuracy) είναι σχετικά χαμηλή και αντίστοιχα το σφάλμα (loss) είναι υψηλό. Στο ακόλουθο στιγμιότυπο από το command window φαίνεται ότι η εκπαίδευση ξεκινάει με ένα προσεγγιστικό μέσο όρο της ακρίβειας της τάξης του 65% και αντίστοιχα με ένα προσεγγιστικό μέσο όρο του σφάλματος της τάξης περίπου ίσο με 20%.

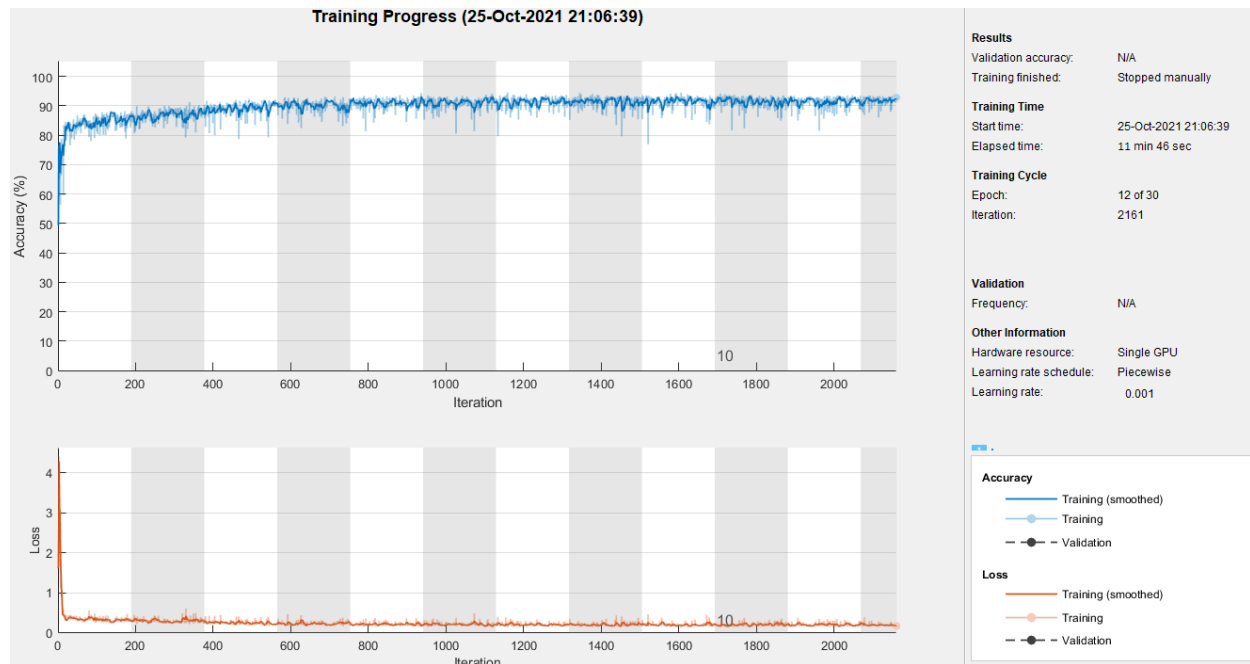
Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:12	50.49%	0.9823	0.0010
1	2	00:00:13	59.12%	2.3389	0.0010
1	3	00:00:13	75.58%	2.9726	0.0010
1	4	00:00:14	70.60%	3.5743	0.0010
1	5	00:00:14	62.73%	2.3027	0.0010
1	6	00:00:14	61.13%	1.3516	0.0010
1	7	00:00:14	66.17%	0.9379	0.0010
1	8	00:00:14	77.96%	0.7288	0.0010
1	9	00:00:15	51.77%	1.3692	0.0010
1	10	00:00:15	73.41%	1.6467	0.0010
1	11	00:00:15	80.67%	0.9569	0.0010
1	12	00:00:15	80.52%	0.5822	0.0010
1	13	00:00:15	79.27%	0.7672	0.0010
1	14	00:00:15	77.38%	0.5218	0.0010
1	15	00:00:15	76.19%	0.5360	0.0010
1	16	00:00:15	86.77%	0.3049	0.0010
1	17	00:00:16	77.57%	0.4994	0.0010
1	18	00:00:16	81.09%	0.4362	0.0010
1	19	00:00:16	83.22%	0.4011	0.0010
1	20	00:00:16	83.38%	0.4626	0.0010
1	21	00:00:16	78.83%	0.5260	0.0010
1	22	00:00:16	77.78%	0.4560	0.0010
1	23	00:00:16	77.73%	0.4280	0.0010

Εικόνα 8.36: Αρχικά iterations εκπαίδευσης CNN με mini-batch size=1

Εκπαίδευση με mini-batch size = 3

Τα ίδια βήματα με παραπάνω θα ακολουθηθούν και για την εκπαίδευση του ίδιου δικτύου (CNN) με την μόνη αλλαγή, το διαφορετικό μέγεθος του mini-batch. Στην περίπτωση αυτή λοιπόν ορίζεται η τιμή 3 για τον παράγοντα mini-batch.

Παρακάτω παρουσιάζονται οι καμπύλες της ακρίβειας (accuracy) και του σφάλματος (loss) αντίστοιχα που προκύπτουν κατά την εκπαίδευση του απλού CNN με χρήση mini-batch size 3:



Εικόνα 8.37: Accuracy και loss εκπαίδευσης απλού CNN με mini-batch size=3

- Παρατηρούμε ότι για να εκπαιδύσουμε το δίκτυο για παρόμοιο αριθμό epochs απαιτήθηκαν περίπου 11 λεπτά. Και εδώ παρατηρούμε μειωμένο χρόνο εκπαίδευση σε σχέση με το Unet.
- Μετά από περίπου 1000 iterations, δηλαδή περίπου 5 epochs παρατηρούμε ότι η ακρίβεια σταθεροποιείται στο 90% και στο τέλος της εκπαίδευσης να προσεγγίζει το 93%. Αντίστοιχα η καμπύλη του σφάλματος παρουσιάζει σταθεροποίηση στα ίδια χρονικά σημεία και στο τέλος της εκπαίδευσης προσεγγίζει τιμές κοντά στο 3%.
- Βλέπουμε ότι αρχικά η ακρίβεια (accuracy) είναι σχετικά χαμηλή και αντίστοιχα το σφάλμα (loss) είναι υψηλό. Στο ακόλουθο στιγμιότυπο από το command window φαίνεται ότι η εκπαίδευση ξεκινάει με ένα προσεγγιστικό μέσο όρο της ακρίβειας της τάξης του 65% και αντίστοιχα με ένα προσεγγιστικό μέσο όρο του σφάλματος της τάξης περίπου ίσο με 25%.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:03	59.75%	2.4132	0.0010
1	2	00:00:04	50.00%	4.2158	0.0010
1	3	00:00:04	79.84%	2.2306	0.0010
1	4	00:00:05	75.99%	2.9098	0.0010
1	5	00:00:05	73.07%	2.0616	0.0010
1	6	00:00:05	66.98%	2.0490	0.0010
1	7	00:00:05	60.29%	0.8110	0.0010
1	8	00:00:06	53.52%	0.8508	0.0010
1	9	00:00:06	47.60%	0.7471	0.0010
1	10	00:00:06	65.01%	0.8808	0.0010
1	11	00:00:07	73.61%	0.6171	0.0010
1	12	00:00:07	73.45%	0.5733	0.0010
1	13	00:00:07	79.15%	0.5088	0.0010
1	14	00:00:08	69.50%	0.6368	0.0010
1	15	00:00:08	80.08%	0.3743	0.0010
1	16	00:00:08	76.04%	0.4259	0.0010
1	17	00:00:09	74.69%	0.4590	0.0010
1	18	00:00:09	79.45%	0.3941	0.0010
1	19	00:00:09	74.33%	0.4168	0.0010
1	20	00:00:09	77.98%	0.3845	0.0010
1	21	00:00:10	80.99%	0.3554	0.0010
1	22	00:00:10	76.86%	0.4337	0.0010
1	23	00:00:10	71.20%	0.4896	0.0010
1	24	00:00:11	80.23%	0.3855	0.0010

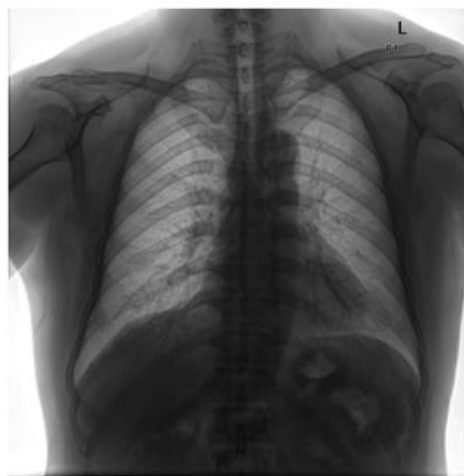
Εικόνα 8.38: Αρχικά iterations εκπαίδευσης CNN με mini-batch size=3

8.5.6 Έλεγχος του απλού CNN με γνωστά δεδομένα

Θα παρατηρήσουμε στο σημείο αυτό, κάποια οπτικά αποτελέσματα κατάτμησης της περιοχής των πνευμόνων, που προέκυψαν από το τεστ του απλού CNN σε 20 τυχαίες ακτινογραφίες του ίδιου του training dataset (γνωστά δεδομένα) για κάθε μία από τις δύο περιπτώσεις mini-batch size. Ακόμα θα δείξουμε και ποσοτικά την ακρίβεια του δικτύου.

Αποτελέσματα εκπαίδευσης με mini-batch size = 1

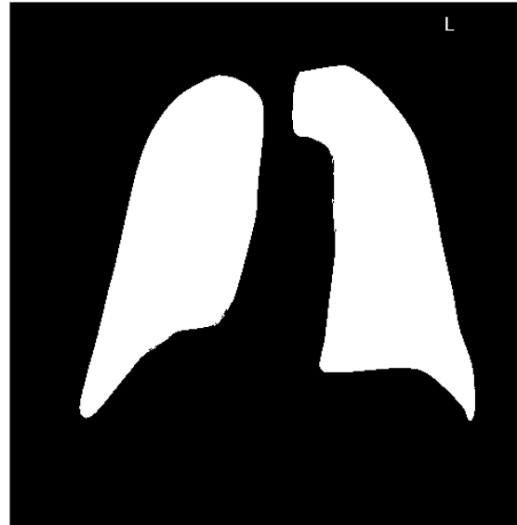
Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα γνωστού δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.39: Γνωστό δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.40: Αποτέλεσμα κατάτμησης



Εικόνα 8.41: Ιδανικό Αποτέλεσμα

Παρακάτω υπολογίζεται, με την γνωστή φόρμουλα, το σφάλμα που προκύπτει από την κατάτμηση:

Γίνεται εισαγωγή στο Matlab των 20 αποτελεσμάτων της κατάτμησης γνωστών δεδομένων.

- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((\text{xor}(a,a2) + \text{xor}(b,b2) + \dots + \text{xor}(u,u2))/20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = \text{sum}(\text{sum}(z)) = 16948$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων, υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z,1) * \text{size}(z,2)) = 0,0689 \text{ ή } 6,89\%$$

Προκύπτει λοιπόν ότι το απλό CNN , όταν εκπαιδευτεί με παράγοντα mini-batch = 1 και ελεγχτεί σε γνωστά δεδομένα , παρουσιάζει σφάλμα ίσο με 6,89%

Αποτελέσματα εκπαίδευσης με mini-batch size = 3

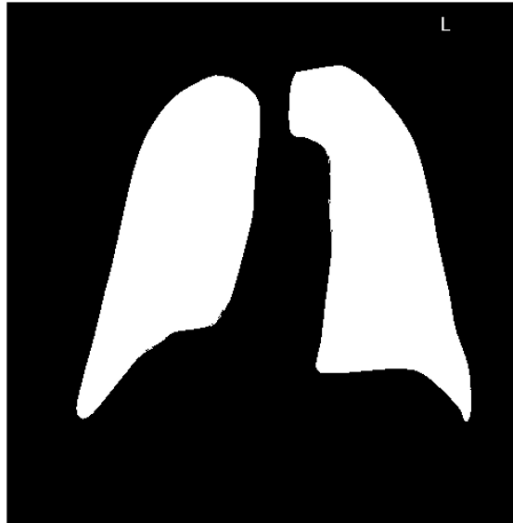
Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα γνωστού δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.42: Γνωστό δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.43: Αποτέλεσμα κατάτμησης



Εικόνα 8.44: Ιδανικό Αποτέλεσμα

Παρατηρούμε ότι στην περίπτωση αυτή το αποτέλεσμα της κατάτμησης εμφανίζεται λιγότερο ακριβές συγκριτικά με το αποτέλεσμα για mini-batch size = 1.

Για τον αριθμητικό υπολογισμό του σφάλματος, εφαρμόζουμε την ίδια μέθοδο με προηγούμενος:

- Γίνεται εισαγωγή στο Matlab των 20 αντίστοιχων αποτελεσμάτων.
- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((xor(a, a2) + xor(b, b2) + + xor(u, u2)) / 20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = sum(sum(z)) = 18319$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων, υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z,1) * \text{size}(z,2)) = 0,0745 \text{ ή } 7,45\%$$

Προκύπτει λοιπόν ότι το απλό CNN , όταν εκπαιδευτεί με παράγοντα mini-batch = 3 και ελεγχτεί σε γνωστά δεδομένα , παρουσιάζει σφάλμα ίσο με 7,45%

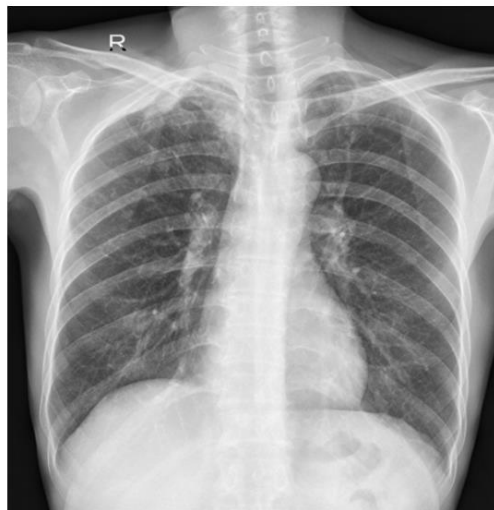
Είναι φανερό ότι και στις δύο περιπτώσεις mini-batch size , το απλό CNN προσφέρει μικρότερης ακρίβειας κατάτμηση.

8.5.7 Έλεγχος του απλού CNN με άγνωστα δεδομένα

Θα παραθέσουμε στο σημείο αυτό, κάποια οπτικά αποτελέσματα κατάτμησης της περιοχής των πνευμόνων, που προέκυψαν από το τεστ του απλού CNN σε 20 τυχαίες ακτινογραφίες που δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση του δικτύου (άγνωστα δεδομένα) για κάθε μία από τις δύο περιπτώσεις mini-batch size. Ακόμα θα δείξουμε και ποσοτικά την ακρίβεια του δικτύου.

Αποτελέσματα εκπαίδευσης με mini-batch size = 1

Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα άγνωστου δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.45: Άγνωστο δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.46: Αποτέλεσμα κατάτμησης



Εικόνα 8.47: Ιδανικό Αποτέλεσμα

Παρακάτω υπολογίζεται, με την γνωστή φόρμουλα, το σφάλμα που προκύπτει από την κατάτμηση:

- Γίνεται εισαγωγή στο Matlab των 20 αποτελεσμάτων της κατάτμησης άγνωστων δεδομένων.
- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((\text{xor}(a, a2) + \text{xor}(b, b2) + \dots + \text{xor}(u, u2)) / 20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = \text{sum}(\text{sum}(z)) = 21146$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων, υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z, 1) * \text{size}(z, 2)) = 0,086 \text{ ή } 8,6\%$$

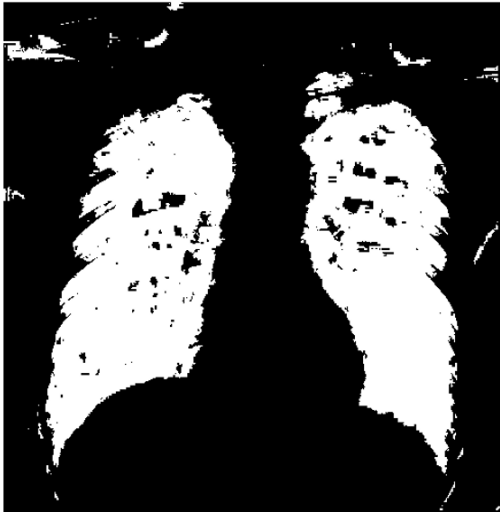
Προκύπτει λοιπόν ότι το απλό CNN, όταν εκπαιδευτεί με παράγοντα mini-batch = 1 και ελεγχτεί σε άγνωστα δεδομένα, παρουσιάζει σφάλμα ίσο με 8,6%

Αποτελέσματα εκπαίδευσης με mini-batch size = 3

Παρουσιάζεται παρακάτω ένα χαρακτηριστικό παράδειγμα άγνωστου δείγματος, συνοδευόμενο από την έξοδο του δικτύου (κατάτμηση πνευμόνων) και το ιδανικό αντίστοιχο αποτέλεσμα (μάσκα):



Εικόνα 8.48: Άγνωστο δείγμα ακτινογραφίας πνεύμονα



Εικόνα 8.49: Αποτέλεσμα κατάτμησης



Εικόνα 8.50: Ιδανικό Αποτέλεσμα

Παρακάτω υπολογίζεται, με την γνωστή φόρμουλα, το σφάλμα που προκύπτει από την κατάτμηση:

- Γίνεται εισαγωγή στο Matlab των 20 αποτελεσμάτων της κατάτμησης άγνωστων δεδομένων.
- Υπολογίζεται ο μέσος όρος Z του αθροίσματος όλων των διαδοχικών λογικών συναρτήσεων XOR.

$$z = ((xor(a, a2) + xor(b, b2) + + xor(u, u2)) / 20);$$

- Υπολογίζεται ο μέσος όρος του συνολικού αριθμού των λάθος εικονοστοιχείων:

$$E = sum(sum(z)) = 21826$$

- Βάσει του μεγέθους των εικόνων ($496 \times 496 = 246016$ pixels) και του μέσου όρου του συνολικού αριθμού των λάθος εικονοστοιχείων , υπολογίζεται το μέσο σφάλμα επί τις εκατό:

$$r = E / (\text{size}(z,1) * \text{size}(z,2)) = 0,0887 \text{ ή } 8,87\%$$

Προκύπτει λοιπόν ότι το απλό CNN , όταν εκπαιδευτεί με παράγοντα mini-batch = 3 και ελεγχτεί σε άγνωστα δεδομένα , παρουσιάζει σφάλμα ίσο με 8,87%

9. Σύγκριση αποτελεσμάτων και συμπεράσματα

9.1 Σύγκριση αποτελεσμάτων

Παρακάτω παρουσιάζεται ένας συγκεντρωτικός πίνακας των σφαλμάτων που προκύπτουν κατά τον έλεγχο των δύο δικτύων με γνωστά και άγνωστα δεδομένα σε συνδυασμό με δύο διαφορετικές τιμές του παράγοντα mini-batch size που ορίστηκε κατά την εκπαίδευση του κάθε δικτύου:

Δίκτυο	Σφάλμα(%)			
	Γνωστά δεδομένα		Άγνωστα δεδομένα	
	Mini-batch =1	Mini-batch =3	Mini-batch =1	Mini-batch =3
Unet	2,69%	4,76%	5,12%	6,80%
Απλό CNN	6,89%	7,45%	8,60%	8,87%

Εικόνα 9.1: Πίνακας σφαλμάτων

Είναι εμφανές, ότι και τα δύο δίκτυα προσφέρουν καλύτερα αποτελέσματα κατάτμησης όταν ελέγχονται με γνωστά δεδομένα, δηλαδή όταν για τον έλεγχο επεξεργάζονται εικόνες που έχουν ήδη χρησιμοποιηθεί για την εκπαίδευσή τους, κάτι που είναι απόλυτα λογικό. Ακόμα παρατηρούμε, ότι και τα δύο νευρωνικά δίκτυα παρουσιάζουν μεγαλύτερο σφάλμα, όταν εκπαιδεύονται με mini-batch size ίσο με 3. Αυτό συμβαίνει, καθώς με την τεχνική αυτή, σε κάθε επανάληψη (iteration), ο αλγόριθμος χρησιμοποιεί 3 δείγματα του συνόλου των δειγμάτων, πρώτου ανανεωθούν οι παράμετροι, σε αντίθεση με την περίπτωση, όπου το mini-batch size ισούται με 1, όπου οι παράμετροι του συστήματος ανανεώνονται μετά από κάθε ένα δείγμα. Ουσιαστικά με την επιλογή mini-batch size = 3, να μεν μειώνουμε σημαντικά την διάρκεια εκπαίδευσης των δικτύων, αλλά λαμβάνουμε αποτελέσματα μικρότερης ακρίβειας. Επίσης, όπως είδαμε και κατά την διαδικασία της εκπαίδευσης, η καμπύλη “accuracy” σταθεροποιείται νωρίτερα, όταν γίνεται χρήση μικρότερου mini-batch size.

Αναφορικά με την σύγκριση των δύο δικτύων, παρατηρούμε ότι το Unet παρουσιάζει μικρότερο σφάλμα από το απλό CNN σε όλες τις περιπτώσεις, γεγονός που οφείλεται στην περιπλοκότερη και ιδιαίτερη αρχιτεκτονική του. Συγκεκριμένα η αρχιτεκτονική του Unet επιτρέπει τον συνδυασμό των χαρτών χαρακτηριστικών μεταξύ του contraction path (encoder path) και του expansion path (decoder path), δίνοντας πιο ακριβείς σημασιολογικές κατατμήσεις. Ωστόσο, μία τόσο περίπλοκη αρχιτεκτονική απαιτεί και μεγαλύτερη χρονική διάρκεια εκπαίδευσης σε σχέση με την χρήση μίας πολύ πιο απλής αρχιτεκτονικής CNN, όπως η δεύτερη αρχιτεκτονική που χρησιμοποιήσαμε.

9.2 Συμπεράσματα

Στην παρούσα διπλωματική εργασία διαδραματίστηκε έρευνα στον κλάδο των τεχνητών νευρωνικών δικτύων και της ψηφιακής επεξεργασίας εικόνων, με σκοπό την πραγματοποίηση της κατάτμησης βιοϊατρικών εικόνων, με τη βοήθεια των δικτύων αυτών. Στο πεδίο των τεχνητών νευρωνικών δικτύων, πραγματοποιήθηκε μελέτη της μορφής, της λειτουργίας και των εφαρμογών διαφορετικών τύπων ΤΝΔ, με εκτενέστερη αναφορά στα συνελκτικά νευρωνικά δίκτυα. Στα πλαίσια της ψηφιακής επεξεργασίας της εικόνας, μελετήθηκαν οι βασικοί μετασχηματισμοί που καθιστούν την κωδικοποίηση των εικόνων πιο αποτελεσματική, αλλά παρατέθηκαν ακόμα διάφορες τεχνικές βελτίωσης και αποκατάστασης των ψηφιακών εικόνων. Όσον αφορά το πειραματικό μέρος της εργασίας, μελετήθηκαν κάποιοι θεμελιώδεις παράγοντες που επηρεάζουν την εκπαίδευση των τεχνητών νευρωνικών δικτύων και αναλύθηκαν δύο διαφορετικές αρχιτεκτονικές CNN, ενός απλού CNN που κατασκευάσαμε και του Unet, τα οποία δίκτυα εκπαιδεύτηκαν στο περιβάλλον του Matlab, ώστε να μπορέσουν στη συνέχεια να πραγματοποιήσουν αυτοματοποιημένη κατάτμηση (διαχωρισμό) της περιοχής των πνευμόνων από ένα σύνολο ακτινογραφιών θώρακος. Ολοκληρώνοντας το πείραμα, καταλήξαμε σε συγκεκριμένες τιμές σφάλματος για κάθε δίκτυο και για διαφορετικές τιμές του παράγοντα mini-batch size, επιβεβαιώνοντας την υπεροχή του δικτύου Unet σε εφαρμογές κατάτμησης βιοϊατρικών εικόνων.

Ωστόσο, είναι προφανές, ότι για να επιτευχθούν τα μέγιστα δυνατά αποτελέσματα κατάτμησης απαιτείται να διεξαχθεί περαιτέρω έρευνα και να διατεθεί μεγαλύτερη υπολογιστική δύναμη. Ο τομέας των τεχνητών νευρωνικών δικτύων είναι ιδιαίτερα ευρύς, με πληθώρα αρχιτεκτονικών που μπορούν να μετατραπούν, ώστε να ικανοποιούν τις απαιτήσεις του συστήματος σε μεγαλύτερο βαθμό. Παραδείγματος χάριν, υπάρχει η δυνατότητα αύξησης του αριθμού των συνελκτικών στρωμάτων, χρήση διαφορετικών συναρτήσεων ενεργοποίησης, εφαρμογή περισσότερων φίλτρων kernel κ.α. Εκτός του πλήθους αρχιτεκτονικών των ΤΝΔ, όπως έχουμε προαναφέρει ποικίλουν και οι παράγοντες της εκπαίδευσης που μπορούν να μεταβληθούν με αμέτρητους συνδυασμούς. Ενδεικτικά, μπορεί να επιλεγεί μεγαλύτερος ή μικρότερος ρυθμός μάθησης, διαφορετικοί αλγόριθμοι βελτιστοποίησης, περισσότερα epochs κ.α. Είναι φανερό λοιπόν, ότι όσο διεξάγονται περισσότερες έρευνες και πειράματα, οι δυνατότητες των νευρωνικών δικτύων στο πεδίο της κατάτμησης της εικόνας είναι απεριόριστες.

Βιβλιογραφία - Αναφορές - Διαδικτυακές Πηγές

- 1) Kamnitsas, K., Ledig, C., Newcombe, V. F., Simpson, J. P., Kane, A. D., Menon, D. K., & Rueckert, D. (2014). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *IEEE Transactions on Medical Imaging*, 34(9), 1993-2004. <https://doi.org/10.1109/TMI.2013.2284099>
- 2) Candemir, S., Jaeger, S., Palaniappan, K., Musco, J. P., Singh, R. K., Xue, Z., Karargyris, A., Antani, S., Thoma, G., & McDonald, C. J. (2014). Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE Transactions on Medical Imaging*, 33(2), 577-590. <https://doi.org/10.1109/TMI.2013.2290491>
- 3) Creative Commons. (2013). Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0). Retrieved from <https://creativecommons.org/licenses/by-nc-sa/4.0/>
- 4) MathWorks. (n.d.). Marker-controlled watershed segmentation. Retrieved from <https://uk.mathworks.com/help/images/marker-controlled-watershed-segmentation.html>
- 5) Wikipedia contributors. (2023, February 17). Image segmentation. In Wikipedia. Retrieved February 22, 2023, from https://en.wikipedia.org/wiki/Image_segmentation
- 6) UpGrad. (2018, February 27). Image segmentation techniques. [Blog post]. Retrieved from <https://www.upgrad.com/blog/image-segmentation-techniques/>
- 7) MathWorks. (2016). Color-Based Segmentation Using K-Means Clustering. Retrieved from https://ch.mathworks.com/content/dam/mathworks/tagteam/Objects/c/88360_93001v00_Color-Based_Seg_K-Means_Clustering_2016.pdf
- 8) MathWorks. (n.d.). Low-Light Image Enhancement. Retrieved from <https://ch.mathworks.com/help/images/low-light-image-enhancement.html>
- 9) MathWorks. (n.d.). Image Enhancement. MathWorks. <https://ch.mathworks.com/discovery/image-enhancement.html>
- 10) MathWorks. (n.d.). Image Enhancement and Restoration. Retrieved from https://uk.mathworks.com/help/images/image-enhancement-and-restoration.html?s_tid=CRUX_lftnav
- 11) The AI Summer. (n.d.). UNet Architectures. Retrieved from <https://theaisummer.com/unet-architectures/>
- 12) MathWorks. (n.d.). Image type conversions. Retrieved from https://uk.mathworks.com/help/images/image-type-conversions.html?searchHighlight=image%20type%20conversion&s_tid=srchtitle
- 13) MathWorks. (n.d.). Im2bw (Image Processing Toolbox). Retrieved from <https://uk.mathworks.com/help/images/ref/im2bw.html#d123e115840>

- 14) Matlab. (n.d.). otsuthresh. Retrieved from <https://uk.mathworks.com/help/images/ref/otsuthresh.html>
- 15) MathWorks. (n.d.). Image Processing and Computer Vision: Image Transform. Retrieved from <https://uk.mathworks.com/discovery/image-transform.html>.
- 16) MathWorks. (n.d.). Image Transforms. Retrieved from https://uk.mathworks.com/help/images/image-transforms.html?s_tid=CRUX_lftnav
- 17) Matlab. (n.d.). Evaluate semantic segmentation result. MathWorks. Retrieved from https://uk.mathworks.com/help/vision/ref/evaluatesemanticsegmentation.html?searchHighlight=semantic%20segmentation%20result%20evaluation&s_tid=srchtitle
- 18) MathWorks. (n.d.). Color-Based Segmentation Using K-Means Clustering. Retrieved from <https://uk.mathworks.com/help/images/color-based-segmentation-using-k-means-clustering.html>
- 19) StackOverflow. (2011, March 12). How can I invert a binary image in MATLAB? StackOverflow. <https://stackoverflow.com/questions/5277005/how-can-i-invert-a-binary-image-in-matlab>
- 20) "Artificial neural network." Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 10 February 2023. Web. 22 February 2023. https://en.wikipedia.org/wiki/Artificial_neural_network
- 21) IBM. (n.d.). Convolutional neural networks. IBM Cloud Education. <https://www.ibm.com/cloud/learn/convolutional-neural-networks>
- 22) ΕΛΛΗΝΑΣ, Ι. Ν. (2010). ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ ΚΑΙ ΒΙΝΤΕΟ - ΑΠΟ ΤΗ ΘΕΩΡΙΑ ΣΤΗΝ ΠΡΑΞΗ (1η έκδοση). ΙΔΙΩΤΙΚΗ.
- 23) Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (pp. 234-241). Springer International Publishing. doi: 10.1007/978-3-319-24574-4_28.
- 24) Vadakkanmarveetil, A. (2021, March 7). Image Dataset Labeling/Annotation. Analytics Vidhya. <https://medium.com/analytics-vidhya/image-dataset-labeling-annotation-bec3390eda2d>.
- 25) MathWorks. (n.d.). Label Pixels for Semantic Segmentation. Retrieved February 22, 2023, from https://uk.mathworks.com/help/vision/ug/label-pixels-for-semantic-segmentation.html?searchHighlight=image%20labelling%20for%20semantic%20segmentation&s_tid=srchtitle
- 26) MathWorks. (n.d.). trainingOptions. Retrieved February 22, 2023, from <https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html>

- 27) Analytics Vidhya. (2019, November 25). 4 tricks to improve deep learning model performance. Retrieved from <https://www.analyticsvidhya.com/blog/2019/11/4-tricks-improve-deep-learning-model-performance/>
- 28) MathWorks. (n.d.). Marker-Controlled Watershed Segmentation. Retrieved February 23, 2023, from <https://uk.mathworks.com/help/images/marker-controlled-watershed-segmentation.html>.
- 29) Jain, A. K. (1997). Fundamentals of digital image processing. Prentice Hall. ISBN: 0-13-336165-9.
- 30) Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing. Prentice Hall.

Παραρτήματα

Κώδικας προσαρμογής διαστάσεων εικόνων και μασκών του dataset:

```
clear; close all; clc; warning off;
%% Loading data directory
dataDir = uigetdir();
imDir=fullfile(dataDir, 'images');
pxDir=fullfile(dataDir, 'mask');
%% Loading images
imds_img = imageDatastore(imDir);
%% resizing images
num_img=length(imds_img.Files);
for i=1:num_img
    I = readimage(imds_img,i);
    if length(size(I))>2
        I=rgb2gray(I);
    end
    I=imresize(I, [226 226]);
    im_name=num2str(i);
    formt='.png';
    full_name=strcat(im_name, formt);
    imwrite(I, full_name);

    fprintf('~');
end
%% Loading mask/labels
imds_px = imageDatastore(pxDir);
%% resizing mask/labels
num_img=length(imds_px.Files);
for i=1:num_img
    I = readimage(imds_px,i);
    if length(size(I))>2
        I=rgb2gray(I);
    end
    I=imresize(I, [226 226]);
    im_name=num2str(i);
    formt='.png';
    full_name=strcat(im_name, formt);
    imwrite(I, full_name);

    fprintf('~');
end
```

Κώδικας εκπαίδευσης Unet με mini-batch size = 1

```
clear; close all; clc; warning off;
%% Loading Data into Matlab
```

```
dataDir = uigetdir(); %provide data directory
imDir=fullfile(dataDir,'images');
pxDir=fullfile(dataDir,'mask');
%% Loading images in datastore
imds = imageDatastore(imDir);
%% Visualizing random loaded image
I = readimage(imds,1);
[r c]=size(I);
figure
imshow(I)
%% Specifying class and corresponding pixel values
classNames = ["LUNGS","OTHERS"];
pixelLabelID = [255 0];

%% Loading Masks
pxds = pixelLabelDatastore(pxDir,classNames,pixelLabelID);
%% Example: Mask over related image
C = readimage(pxds,1);
B = labeloverlay(I,C);
figure
imshow(B)
y= rgb2gray(B);
figure
imshow(y)
J = im2bw( y , 0.361 )
figure
imshow(J)
G = not(J);
figure
imshow(G)
%% Loading U-Net from Matlab
imageSize = [r c];
numClasses = 2;
lgraph = unetLayers(imageSize, numClasses);

%% Training options
options = trainingOptions('adam', ...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',2,...
    'LearnRateDropFactor',0.2,...
    'InitialLearnRate',0,001, ...
    'L2Regularization',0.005, ...
    'MaxEpochs',30, ...
    'MiniBatchSize',1, ...
    'Shuffle','every-epoch', ...
    'VerboseFrequency',2,...
    'Plots','training-progress',...
    'ValidationPatience', 4);
```



```
% Preparing Training Data
ds = combine(imds,pxds);
%% Start Training
net = trainNetwork(ds,lgraph,options)
```

Κώδικας εκπαίδευσης Unet με mini-batch size = 3

```
clear; close all; clc; warning off;
%% Loading Data into Matlab
dataDir = uigetdir(); %provide data directory
imDir=fullfile(dataDir,'images');
pxDir=fullfile(dataDir,'mask');
%% Loading images in datastore
imds = imageDatastore(imDir);
%% Visualizing random loaded image
I = readimage(imds,1);
[r c]=size(I);
figure
imshow(I)
%% Specifying class and corresponding pixel values
classNames = ["LUNGS","OTHERS"];
pixelLabelID = [255 0];

%% Loading Masks
pxds = pixelLabelDatastore(pxDir,classNames,pixelLabelID);
%% Example: Mask over related image
C = readimage(pxds,1);
B = labeloverlay(I,C);
figure
imshow(B)
y= rgb2gray(B);
figure
imshow(y)
J = im2bw( y , 0.361 )
figure
imshow(J)
G = not(J);
figure
imshow(G)
%% Loading U-Net from Matlab
imageSize = [r c];
numClasses = 2;
lgraph = unetLayers(imageSize, numClasses);

%% Training options
options = trainingOptions('adam', ...
    'LearnRateSchedule','piecewise',...
```

```
'LearnRateDropPeriod',2,...
'LearnRateDropFactor',0.2,...
'InitialLearnRate',0,001, ...
'L2Regularization',0.005, ...
'MaxEpochs',30, ...
'MiniBatchSize',3, ...
'Shuffle','every-epoch', ...
'VerboseFrequency',2,...
'Plots','training-progress',...
'ValidationPatience', 4);
%% Preparing Training Data
ds = combine(imds,pxds);
%% Start Training
net = trainNetwork(ds,lgraph,options)
```

Κώδικας εκπαίδευσης απλού CNN με mini-batch size = 1

```
clear; close all; clc; warning off;
%% Loading Data into Matlab

%NOTE: For this script you can load images from folder
"sample1" and "Sample2"

dataDir = uigetdir(); %provide data directory
imDir=fullfile(dataDir,'images');
pxDir=fullfile(dataDir,'mask');
%% Loading images in datastore
imds = imageDatastore(imDir);
%% Visualizing random loaded image
I = readimage(imds,1);
[r c]=size(I);
figure
imshow(I)
%% Specifying class and corresponding pixel values
classNames = ["LUNGS","OTHERS"];
pixelLabelID = [255 0];

%% Loading Masks
pxds = pixelLabelDatastore(pxDir,classNames,pixelLabelID);
%% Example: Mask over related image
C = readimage(pxds,1);
B = labeloverlay(I,C);
figure
imshow(B)
y= rgb2gray(B);
figure
imshow(y)
J = im2bw( y , 0.361 )
```

```
figure
imshow(J)
G = not(J);
figure
imshow(G)

%% Network Architecture
numFilters = 128;
filterSize = 3;
numClasses = 2;
layers = [
    imageInputLayer([r c 1])
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    reluLayer()
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    reluLayer()

    transposedConv2dLayer(4,numFilters,'Stride',2,'Cropping',1);
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    convolution2dLayer(1,numClasses);
    softmaxLayer()
    pixelClassificationLayer()
];

%% Training Options
options = trainingOptions('adam', ...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',2,...
    'LearnRateDropFactor',0.2,...
    'InitialLearnRate',0.001, ...
    'MaxEpochs',30, ...
    'MiniBatchSize',1, ...
    'Shuffle','every-epoch', ...
    'VerboseFrequency',1,...
    'Plots','training-progress',...
    'ValidationPatience', 4);
%'Momentum',0.9, ...
%% Preparing Training Data
trainingData = pixelLabelImageDatstore(imds,pxds);
%% Start Training
net = trainNetwork(trainingData,layers,options);
```

Κώδικας εκπαίδευσης απλού CNN με mini-batch size = 3

```
clear; close all; clc; warning off;
%% Loading Data into Matlab
```

```
%NOTE: For this script you can load images from folder
"sample1" and "Sample2"

dataDir = uigetdir(); %provide data directory
imDir=fullfile(dataDir,'images');
pxDir=fullfile(dataDir,'mask');
%% Loading images in datastore
imds = imageDatastore(imDir);
%% Visualizing random loaded image
I = readimage(imds,1);
[r c]=size(I);
figure
imshow(I)
%% Specifying class and corresponding pixel values
classNames = ["LUNGS","OTHERS"];
pixelLabelID = [255 0];

%% Loading Masks
pxds = pixelLabelDatastore(pxDir,classNames,pixelLabelID);
%% Example: Mask over related image
C = readimage(pxds,1);
B = labeloverlay(I,C);
figure
imshow(B)
y= rgb2gray(B);
figure
imshow(y)
J = im2bw( y , 0.361 )
figure
imshow(J)
G = not(J);
figure
imshow(G)

%% Network Architecture
numFilters = 128;
filterSize = 3;
numClasses = 2;
layers = [
    imageInputLayer([r c 1])
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    reluLayer()
    maxPooling2dLayer(2,'Stride',2)
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    reluLayer()

    transposedConv2dLayer(4,numFilters,'Stride',2,'Cropping',1);
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    convolution2dLayer(1,numClasses);
```

```
softmaxLayer()
pixelClassificationLayer()
];

%% Training Options
options = trainingOptions('adam', ...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',2,...
    'LearnRateDropFactor',0.2,...
    'InitialLearnRate',0.001, ...
    'MaxEpochs',30, ...
    'MiniBatchSize',3, ...
    'Shuffle','every-epoch', ...
    'VerboseFrequency',1,...
    'Plots','training-progress',...
    'ValidationPatience', 4);
%'Momentum',0.9, ...
%% Preparing Training Data
trainingData = pixelLabelImageDatastore(imds,pxds);
%% Start Training
net = trainNetwork(trainingData, layers, options);
```

Κώδικας ελέγχου των δικτύων

```
%% Testing Trained Network
for i=1:20

    testImage=readimage(imds,i);
    C = semanticseg(testImage,net);
    B = labeloverlay(testImage,C);
    imshow(B)
    y= rgb2gray(B);
    figure(4)
    imshow(y)
    J = im2bw( y , 0.361 )
    figure(5)
    imshow(J)
    G = not(J);
    figure(6)
    imshow(G)

    disp('Press enter to see next result')
    pause;
    close all
end
```

Ψηφιακή επεξεργασία και κατάτμηση βιοϊατρικών εικόνων με χρήση τεχνητών νευρωνικών δικτύων