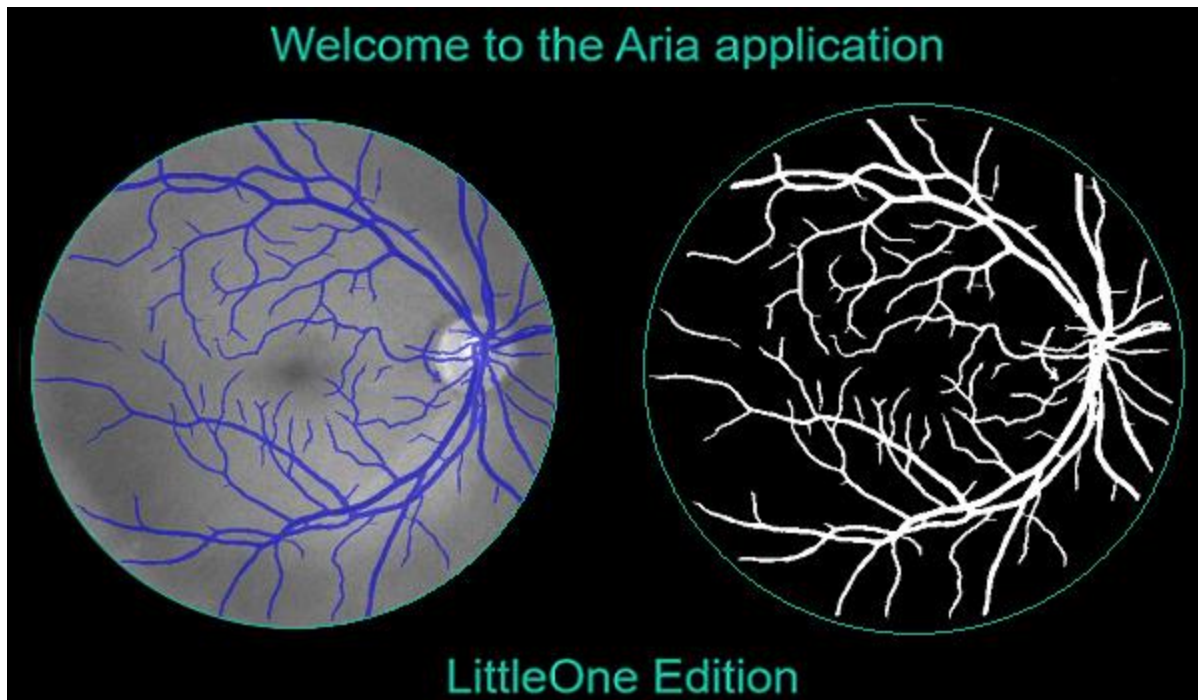




ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ARIA APP | LITTLE ONE EDITION, Η ΉΞΥΠΙΝΗ ΕΦΑΡΜΟΓΗ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΜΟΝΤΕΛΩΝ UNET ΣΕ ΕΡΓΑΣΙΕΣ ΚΑΤΑΤΜΗΣΗΣ ΤΩΝ ΑΓΓΕΙΩΝ ΤΟΥ ΑΜΦΙΒΛΗΣΤΡΟΕΙΔΟΥΣ

του μεταπτυχιακού φοιτητή

Ντάνιελ Ρεζ

Επιβλέπων
Παντελής Ασβεστάς, καθηγητής

Τριμελής Επιτροπή

Παντελεήμων Ασβεστάς Καθηγητής	Δημήτριος Γκλώτσος Καθηγητής	Σπυρίδων Κωστόπουλος Αναπλ. Καθηγητής
Ψηφιακή Υπογραφή	Ψηφιακή Υπογραφή	Ψηφιακή Υπογραφή

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο υπογράφων **PEZ NTANIEL** του **ΑΧΜΑΝΤ** με αριθμό μητρώου **2015**
Φοιτητής του Τμήματος **Μηχανικών Βιοϊατρικής** της Σχολής Μηχανικών
του Πανεπιστημίου Δυτικής Αττικής

Δηλώνω υπεύθυνα ότι:

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

Ημερομηνία

Ο Δηλών

1^η Σεπτεμβρίου, 2023

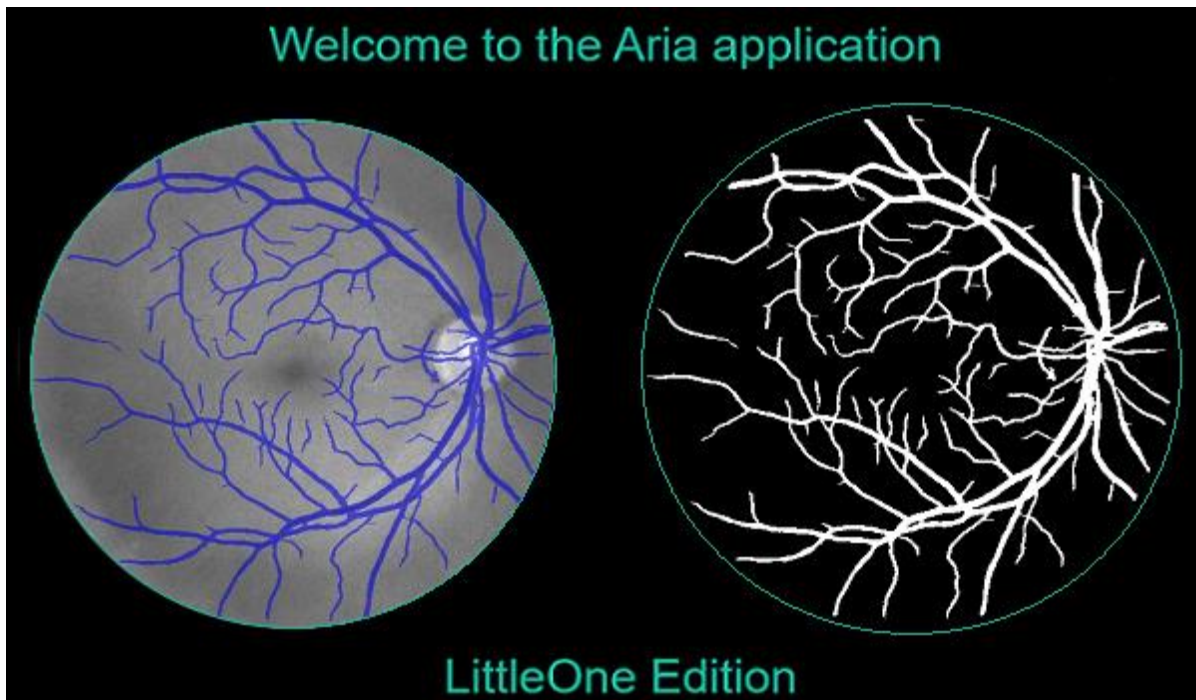




UNIVERSITY OF WEST ATTICA

FACULTY OF ENGINEERING

DEPARTMENT OF BIOMEDICAL ENGINEERING



DIPLOMA THESIS

ARIA APP | LITTLE ONE EDITION, THE SMART APP FOR BUILDING AND
EVALUATING U-NET MODELS ON RETINAL VESSEL SEGMENTATION TASKS

by MSc student

Daniel Rez

Supervisor

Pantelis Asvestas, professor

Three-member Thesis Committee

Panteleimon Asvestas Professor	Dimitrios Glotsos Professor	Spiros Kostopoulos Assoc. Professor
Digital Signature	Digital Signature	Digital Signature

Περίληψη

Η ακριβής κατάτμηση των αγγείων του αμφιβληστροειδούς αποτελεί ζωτικό βήμα για πολλές ιατρικές εφαρμογές, καθώς χρησιμοποιείται ευρέως για την παρακολούθηση της εξέλιξης της υγείας και την αξιολόγηση διαφόρων οφθαλμολογικών ασθενειών. Ωστόσο, η παραδοσιακή χειροκίνητη κατάτμηση των αγγείων από εκπαιδευμένους ειδικούς απαιτεί πολύ χρόνο και είναι μια επαναλαμβανόμενη διαδικασία.

Στην προσπάθεια να αυτοματοποιηθεί η διαδικασία αυτή, έχουν προταθεί πολλές προσεγγίσεις τα τελευταία χρόνια. Μία από τις πιο προηγμένες είναι η χρήση νευρωνικών δικτύων και της βαθιάς μάθησης. Στην ακόλουθη διπλωματική εργασία, εμπνευσμένοι από τα κατορθώματα της τεχνητής νοημοσύνης σε απαιτητικά προβλήματα computer vision, ταξινόμησης εικόνων, κ.α., παρουσιάζεται η εφαρμογή Aria App, LittleOne edition η οποία παρέχει ένα φιλικό προς τον χρήστη περιβάλλον, στο οποίο μπορεί να γίνει η προ-επεξεργασία των εικόνων, η δημιουργία, εκπαίδευση και αξιολόγηση ενός δικτύου. Μέσω της εφαρμογής, η αντιμετώπιση της πρόκλησης του εντοπισμού των αιμοφόρων αγγείων, γίνεται μέσω του δικτύου U-Net. Η ιδιαιτερότητα της αρχιτεκτονικής U-Net με το χαρακτηριστικό σχήμα "U", εννοεί την προκειμένη εφαρμογή, καθώς σε πρώτο επίπεδο «αναγνωρίζει» τι είναι ένα αγγείο, ενώ σε δεύτερο επίπεδο, «μαθαίνει» να το χαρτογραφεί, εκτελώντας με αυτό το τρόπο μια σημασιολογική κατάτμηση σε μια εικόνα.

Για την αξιολόγηση και την εκπαίδευση του δικτύου, χρησιμοποιήθηκε το αποθετήριο εικόνων DRIVE, το οποίο αποτελείται από 40 εικόνες.

Τα αποτελέσματα των πειραμάτων μέσω της εφαρμογής με μόλις 2 ώρες συνολικής εκπαίδευσης είναι συγκρίσιμα με σχετικές έρευνες των τελευταίων 5 ετών που εστιάζουν μόνο στην αξιολόγηση του δικτύου U-Net και εμφανίζονται ως εξής:

Acc 0.949 | Sens 0.932 | Spec 0.951

Λέξεις Κλειδιά

Κατάτμηση, αιμοφόρα αγγεία, οφθαλμός, αμφιβληστροειδής, U-NET, νευρωνικά δίκτυα, βαθιά μάθηση, μηχανική μάθηση, ιατρική εικόνα

Abstract

The precise segmentation of the vessels of the retina is a crucial step for various medical applications, as it is widely used for monitoring disease progression and evaluating various ophthalmological conditions. However, the traditional manual vessel segmentation by trained experts is a time-consuming and repetitive process.

In an effort to automate this process, numerous approaches have been proposed in recent years. One of the most advanced, involves the use of neural networks and deep learning. In the following dissertation, inspired by the achievements of artificial intelligence in demanding computer vision tasks, image classification, and more, we present the Aria App, LittleOne edition. This application provides a user-friendly environment where image pre-processing, network creation, training, and evaluation can be performed. Through this application, addressing the challenge of identifying blood vessels is accomplished using the U-Net network. The unique architecture of U-Net, characterized by its 'U' shape, favours this particular application. In the initial stage, it 'recognizes' what constitutes a vessel, while in the subsequent stage, it 'learns' to map it, thus performing semantic segmentation on an image.

For the evaluation and training of the network, the DRIVE image repository is used, which consists of 40 images in total.

The results of the experiments performed through the app with just 2 hours of total training are comparable to related research papers in the last 5 years which focus on the evaluation of U-Net only and they appear as:

Acc 0.949 | Sens 0.932 | Spec 0.951

Keywords

Segmentation, retinal vessels, eye, retina, U-Net, neural networks, deep learning, machine learning, medical imaging

Ευχαριστίες

Με την συγγραφή της παρούσας διπλωματικής εργασίας, η φοίτηση μου στο μεταπτυχιακό πρόγραμμα «Προηγμένα Συστήματα και Μέθοδοι στην Βιοϊατρική Τεχνολογία» έρχεται στο τέλος της και κλείνει έναν κύκλο φοιτητικών χρόνων στο πανεπιστήμιο δυτικής αττικής, όπου και σε συνδυασμό με την ολοκλήρωση της στρατιωτικής θητείας μου στον Ε.Σ, μου δίνεται πλέον η δυνατότητα να εξελιχθώ από φοιτητής σε ολοκληρωμένος πολίτης, με πολλά εφόδια, εμπειρίες και προοπτικές εξέλιξης στον τομέα της βιοϊατρικής τεχνολογίας.

Θα ήθελα να ευχαριστήσω από καρδιάς όλο το Εργαστήριο Ιατρικής Εικόνας και Σήματος (ΕΙΣΕ) (medisp.uniwa.gr) του τμήματος Μηχανικών Βιοϊατρικής Τεχνολογίας, για την τεράστια βοήθεια, γνώσεις και εφόδια που παρέχουν όλα αυτά τα χρόνια τόσο σε προπτυχιακό αλλά και σε μεταπτυχιακό επίπεδο, ειδικότερα σε τομείς της ιατρικής, βίο-στατιστικής, προγραμματισμού, μηχανικής & βαθιάς μηχανικής μάθησης σε περιβάλλον MATLAB και Python. Γνώσεις οι οποίες είναι απαραίτητες για τους ταχύτατους ρυθμούς της έρευνας, ανάπτυξης και αναβάθμισης των υπάρχοντων τεχνολογιών στο γενικότερο σύνολο, με κύριο όμως ρόλο, την επιστήμη της Ιατρικής.

Επιπροσθέτως, θα ήθελα να ευχαριστήσω ιδιαίτερα:

- Τον καθηγητή και επιβλέπων της παρούσας διπλωματικής εργασίας, κύριο Παντελή Ασβεστά, για την εμπιστοσύνη του στις ικανότητες μου, την προσοχή και τη καθοδήγηση του για την εργασία αυτή.
- Την Περσίλα Κουμάτη ([linkedin.com/persila-koumati](https://www.linkedin.com/persila-koumati)), τελειόφοιτη του τμήματος Ιατρικής του Πανεπιστημίου Θεσσαλίας. Για την συνεισφορά της στην αναγνώριση, κατανόηση και επεξεργασία των εικόνων του αμφιβληστροειδούς, τις εξαιρετικές ιδέες που συζητήθηκαν κατά έναρξη της εκπόνησης της παρούσας διπλωματικής εργασίας και για τη δοκιμή της εφαρμογής σε διάφορα στάδια της υλοποίησής της.
- Τον πολύ καλό μου φίλο Μανώλη Δ, για την απέραντη υποστήριξη του κατά τη διάρκεια της εκπόνησης της παρούσας διπλωματικής.

Acknowledgements

With the completion of this thesis, my enrolment in the postgraduate program "Advanced Systems and Methods in Biomedical Technology" comes to an end, marking the culmination of a cycle of academic years at the University of West Attica. This period, combined with my fulfilled military duties, affords me the opportunity to transition from an academic student to a well-rounded individual, equipped with a wealth of knowledge, experiences, and prospects for advancement in the field of biomedical technology.

I would like to wholeheartedly thank the entire Medical Imaging and Signal Processing Laboratory (MEDISP) (medisp.uniwa.gr) of the Department of Biomedical Engineering for their immense support, expertise, and resources provided throughout these years, both at the undergraduate and postgraduate levels. Their contributions span various domains, including medicine, biostatistics, programming, as well as machine learning and deep learning using MATLAB and Python. These competencies are essential for the rapid pace of research, development, and enhancement of existing technologies in the broader context, with a central role in the realm of Medicine.

Additionally, I would like to express my sincere gratitude to:

- Professor and supervisor of this thesis, Mr. Pantelis Asvestas, for his confidence in my abilities, his attention, and guidance throughout this work.
- MD Persila Koumati ([linkedin.com/persila-koumati](https://www.linkedin.com/persila-koumati)), a graduate of the Medical Department at the University of Thessaly. Her contribution in image recognition and comprehension, and processing of retinal images, as well as the great ideas discussed at the inception of this thesis, have been invaluable, along with her application testing at various stages of implementation has been invaluable.
- My dear friend Manolis D, for his unwavering support throughout the completion of this thesis.

Περιεχόμενα

Περίληψη.....	9
Λέξεις Κλειδιά.....	9
Abstract.....	11
Keywords.....	11
Ευχαριστίες.....	13
Acknowledgements.....	15
Περιεχόμενα.....	17
Κατάλογος Συντομογραφιών.....	20
Κατάλογος Σχημάτων και Εικόνων.....	22
Εισαγωγή.....	28
Κεφάλαιο 1: Θεωρητικό υπόβαθρο.....	31
1.1 Αντικείμενο Ενδιαφέροντος - Ετικετοποίηση.....	31
1.2 Αρχιτεκτονικές Βαθιών Συνελικτικών Δικτύων.....	32
1.3 Στόχοι – Προσδοκίες.....	33
1.4 Κλινική ανατομία του οφθαλμού.....	34
1.5 Απεικόνιση – Λήψη εικόνας του βυθού.....	37
1.6 Μηχανική Μάθηση - Machine Learning.....	38
1.7 Βαθιά Μάθηση – Deep Learning.....	39
1.8 Συνελικτικά Νευρωνικά Δίκτυα – Convolutional Neural Networks (CNN).....	41
Κεφάλαιο 2: Δίκτυο U-Net.....	46
2.1 Αρχιτεκτονική του δικτύου U-Net.....	46
2.2 Το U-Net σε εφαρμογές ιατρικού τομέα.....	47
2.3 Αρχιτεκτονική U-Net.....	48
Κεφάλαιο 3: Προετοιμασία δεδομένων.....	52
3.1 Σημασία της επεξεργασίας των δεδομένων.....	52
3.2 Επεξεργασία εικόνων – Image Preprocessing.....	53
3.2.1 Μετατροπή του χρώματος της εικόνας.....	54
3.2.2 Τεμαχισμός εικόνας – Image Patch Extraction.....	54
3.2.3 Κανονικοποίηση – Standardization.....	56

3.2.4	Εξισορρόπηση Ιστογράμματος – CLAHE.....	57
3.2.5	Επαύξηση των δεδομένων – Data Augmentation	58
Κεφάλαιο 4: Εφαρμογή Aria.....		60
4.1	Παρουσίαση εφαρμογής Aria App LittleOne Edition	60
4.2	Τεχνολογική επισκόπηση της Aria App	61
4.3	Overview: Aria App – Tabs	62
Κεφάλαιο 5: Aria App - Frontend.....		66
5.1	Aria App: Frontend – Homepage Tab	66
5.2	Aria App: Frontend – Updates Tab	70
5.3	Aria App: Frontend – Image Processing Tab	72
5.4	Aria App: Frontend – Training Tab	79
5.5	Aria App: Frontend – Training Progress Tab	85
5.6	Aria App: Frontend – Evaluation Tab	86
5.7	Aria App: Frontend – Επιπλέον δυνατότητες.....	90
Κεφάλαιο 6: Aria App - Backend.....		94
6.1	Aria App: Backend – Επεξήγηση του backend των βημάτων	94
6.2	Aria App: Backend – Training Tab	102
6.3	Aria App: Backend – Training Progress Tab	114
6.4	Aria App: Backend – Evaluation Tab	115
Κεφάλαιο 7: Αποτελέσματα Aria App		124
7.1	Πειραματικό Στάδιο – Aria App in Action (AAA).....	124
7.2	Αξιολόγηση με Weighted pixelClassificationLayer.....	147
7.3	Αξιολόγηση με DicePixelClassificationLayer	153
7.4	Αξιολόγηση με Focal Loss Layer	159
7.5	Αξιολόγηση με Weighted pixelClassifationLayer (2)	165
Κεφάλαιο 8: Τελικά συμπεράσματα.....		172
8.1	Συμπεράσματα	172
8.2	Απλότητα και επιτάχυνση διαδικασιών.....	172
8.3	Μελλοντικές Επεκτάσεις	173
Βιβλιογραφία και Παραπομπές		175

Κατάλογος Συντομογραφιών

App = Application

CNN = Convolutional Neural Network

UNET = Unified Network

CAD = Computer-Aided Diagnosis ή Computer-Aided Detection

DSS = Decision Support System

AI = Artificial Intelligence

ML = Machine Learning

DL = Deep Learning

UI = User Interface

UX = User Experience

Κατάλογος Σχημάτων και Εικόνων

- Εικόνα 1: Η ανατομία του αμφιβληστροειδούς
- Εικόνα 2: DRIVE dataset δείγματα / υπό-εικόνες μεγέθους 128 x 128
- Εικόνα 3: Η αρχική οθόνη “homepage” της Aria App, LittleOne edition
- Εικόνα 4: Ανατομία του οφθαλμού / maniateas.gr
- Εικόνα 5: Αντίληψη των εικόνων από το οπτικό σύστημα / Διάλεξη EIE (eclass.uniwa.gr)
- Εικόνα 6: Ανατομία του αμφιβληστροειδούς / ucl.ac.uk
- Εικόνα 7: Εξοπλισμός – Ψηφιακή κάμερα απεικόνισης του βυθού / today.uconn.edu
- Εικόνα 8: Διαφορές – συσχετίσεις μεταξύ των AI, ML, DL / wearebrain.com
- Εικόνα 9: Παράδειγμα ενός συνελκτικού δικτύου (CNN) / analyticsvidhya.com
- Εικόνα 10: Παράδειγμα του επιπέδου συνέλιξης / researchgate.net
- Εικόνα 11: Παράδειγμα υποδειγματοληψίας / towardsdatascience.com
- Εικόνα 12: Παράδειγμα των διάφορων επιπέδων χαρακτηριστικών / semiconductorforu.com
- Εικόνα 13: Παράδειγμα διαδικασίας χαρτογράφησης χαρακτηριστικών / mdpi.com
- Εικόνα 14: Παράδειγμα activation output map με GRAD CAM / researchgate.net
- Εικόνα 15: Παράδειγμα activation – feature mapping (έξοδος) / arxiv.org
- Εικόνα 16: Απεικόνιση της αρχιτεκτονικής του δικτύου U-Net / lmb.informatik.uni-freiburg.de
- Εικόνα 17: Παράδειγμα πινάκων μιας RGB εικόνας / researchgate.net
- Εικόνα 18: Homepage – Αρχική καρτέλα της εφαρμογής Aria App, LittleOne Edition
- Εικόνα 19: Updates – Καρτέλα ενημερώσεων
- Εικόνα 20: Image Processing – Οθόνη προεπεξεργασίας
- Εικόνα 21: Training – Καρτέλα εκπαίδευσης μοντέλου
- Εικόνα 22: Training Progress – Καρτέλα προόδου του μοντέλου
- Εικόνα 23: Evaluation – Καρτέλα αξιολόγησης μοντέλου
- Εικόνα 24: Homepage – Αρχική καρτέλα
- Εικόνα 25: Homepage – Τίτλος αρχικής καρτέλας
- Εικόνα 26: Background (Παρασκήνιο)
- Εικόνα 27: Homepage – Φόρμα σύνδεσης
- Εικόνα 28: Homepage – Loading screen
- Εικόνα 29: Homepage – Εμφανές πεδίο του κωδικού
- Εικόνα 30: Homepage – Μη εμφανές πεδίου του κωδικού
- Εικόνα 31: Homepage – Καρτέλα ενημερώσεων
- Εικόνα 32: Updates – Τίτλος καρτέλας ενημερώσεων
- Εικόνα 33: Updates – Πεδίο text των ενημερώσεων
- Εικόνα 34: Image Processing – Καρτέλα προ-επεξεργασίας
- Εικόνα 35: Image Processing – Βήμα 1 της προ-επεξεργασίας
- Εικόνα 36: Image Processing – Βήμα 2 της προ-επεξεργασίας
- Εικόνα 37: Image Processing – Επιλογές του βήματος 2
- Εικόνα 38: Image Processing – Βήμα 3 της προ-επεξεργασίας
- Εικόνα 39: Image Processing – Placeholder – Πεδίο εικόνας
- Εικόνα 40: Image Processing – Δείκτης τωρινής εικόνας προς παρουσίαση
- Εικόνα 41: Image Processing – Κουμπιά προβολής δείγματος και εφαρμογής επιλογών στο dataset
- Εικόνα 42: Image Processing – Βήμα 4 της προ-επεξεργασίας
- Εικόνα 43: Image Processing – Τελευταία επιλογή προ-επεξεργασίας πριν την εκπαίδευση
- Εικόνα 44: Training – Καρτέλα παραμετροποίησης δικτύου προς εκπαίδευση
- Εικόνα 45: Training – Βήμα 1 της καρτέλας εκπαίδευσης

Εικόνα 46: Training – Βήμα 2 της καρτέλας εκπαίδευσης
Εικόνα 47: Training – Επιλογές του βήματος 2 για την παραμετροποίηση της εκπαίδευσης
Εικόνα 48: Training – Βήμα 3 της καρτέλας εκπαίδευσης
Εικόνα 49: Training – Επιλογή δικτύου και βάθος του encoder side
Εικόνα 50: Training – Επιλογές – παράμετροι του ImageInputLayer
Εικόνα 51: Training – Στάδιο επιλογής final output layer
Εικόνα 52: Training – Λίστα επιλογών final output layers
Εικόνα 53: Training – Επιπλέον πλαίσιο παραμετροποίησης Dropout layer και δημιουργίας δικτύου
Εικόνα 54: Training Progress – Πεδίο προβολής της προόδου εκπαίδευσης του δικτύου
Εικόνα 55: Evaluation – Καρτέλα αξιολόγησης δικτύων Aria Network, et al
Εικόνα 56: Evaluation – Πεδίο προβολής της αξιολόγησης του δικτύου για εικόνες ατομικά και συλλογικά
Εικόνα 57: Evaluation – Πεδία προβολής τελικού αποτελέσματος κατάτμησης και επιπλέον ρυθμίσεις
Εικόνα 58: Evaluation – Επιλογή δικτύου προς αξιολόγηση και επιλογή ενεργοποίησης μηχανισμού PK
Εικόνα 59: Image Processing – Καρτέλα προ-επεξεργασίας εικόνων
Εικόνα 60: Image Processing – Ξεχωριστό παράθυρο παρουσίασης Original Image
Εικόνα 61: Image Processing – Ξεχωριστό παράθυρο παρουσίασης Processed Image
Εικόνα 62: Evaluation – Καρτέλα αξιολόγησης δικτύου Aria Network, et al
Εικόνα 63: Image Processing – Βήμα 1 της προ-επεξεργασίας, κουμπιά επιλογής φακέλων
Εικόνα 64: Image Processing – Βήμα 2 της προ-επεξεργασίας, τεχνικές/μέθοδοι προ-επεξεργασίας
Εικόνα 65: Image Processing – Βήμα 3 της καρτέλας προ-επεξεργασίας
Εικόνα 66: Image Processing – Βήμα 4 της καρτέλας προ-επεξεργασίας
Εικόνα 67: Training – Βήμα 1 της καρτέλας εκπαίδευσης
Εικόνα 68: Training – Βήμα 2 και Βήμα 3 της καρτέλας εκπαίδευσης
Εικόνα 69: Training – Στάδιο επιλογής final output layer
Εικόνα 70: Training – Παράδειγμα εμφάνισης classWeights μέσω της επιλογής Weighted final output layer
Εικόνα 71: Training – Επιλογή dicePixelClassificationLayer ως final output layer
Εικόνα 72: Training – επιλογή FocalLossLayer ως final output layer
Εικόνα 73: Training – Επιπλέον πλαίσιο παραμετροποίησης Dropout layer και δημιουργίας δικτύου
Εικόνα 74: Training – Ξεχωριστό παράθυρο ανάλυσης της αρχιτεκτονικής του δικτύου
Εικόνα 75: Training – Ξεχωριστό παράθυρο παρουσίασης προόδου εκπαίδευσης
Εικόνα 76: Training Progress – Αναλυτική μορφή παρουσίασης προόδου εκπαίδευσης
Εικόνα 77: Evaluation – Καρτέλα αξιολόγησης δικτύου με παραδείγματα
Εικόνα 78: Evaluation – Πλαίσιο προβολής αριθμητικών αποτελεσμάτων της απόδοσης του δικτύου
Εικόνα 79: Evaluation – Ξεχωριστό παράθυρο οπτικοποίησης αποτελεσμάτων απόδοσης
Εικόνα 80: Homepage – Φόρμα σύνδεσης χρήστη
Εικόνα 81: Homepage – Εκκίνηση επαλήθευσης των διαπιστευτηρίων του χρήστη (login)
Εικόνα 82: Homepage – Στάδιο επαλήθευσης (loading)
Εικόνα 83: Updates – Καρτέλα νέων ενημερώσεων
Εικόνα 84: Image Processing - Βήμα 1 της καρτέλας προ-επεξεργασίας
Εικόνα 85: Image Processing – Προβολή δείγματος από το φάκελο εικόνων προς εκπαίδευση
Εικόνα 86: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Εικόνα 87: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Εικόνα 88: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Εικόνα 89: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Εικόνα 90: Image Processing – Εφαρμογή όλων των επιλεγμένων τεχνικών στο dataset

Εικόνα 91: Image Processing – Εμφάνιση μηνύματος επιτυχούς εφαρμογής των επιλογών

Εικόνα 92: Image Processing – Επιλογή διαμόρφωσης διαστάσεων και εξαγωγής patches

Εικόνα 93: Image Processing – Επιλογή resizing / patching και εισαγωγή τιμών στα πεδία

Εικόνα 94: Image Processing – Εφαρμογή και μετάβαση στη καρτέλα εκπαίδευσης

Εικόνα 95: Image Processing – Εμφάνιση μηνύματος επιτυχούς αλλαγής διαστάσεων

Εικόνα 96: Image Processing – Εμφάνιση ειδοποίησης επιτυχούς εξαγωγής των patches

Εικόνα 97: Windows OS – Temporary Directory – Φάκελοι επεξεργασμένων εικόνων

Εικόνα 98: Windows OS – Temporary Directory – Προβολή επεξεργασμένων εικόνων

Εικόνα 99: Windows OS – Temporary Directory – Προβολή δειγμάτων εξαγωγής patches

Εικόνα 100: Image Processing – Εφαρμογή και μετάβαση στη καρτέλα εκπαίδευσης

Εικόνα 101: Training – Παράμετροι Data Augmentation

Εικόνα 102: Training - Παράδειγμα τιμών στις παραμέτρους του Data Augmentation

Εικόνα 103: Training – Παράδειγμα εφαρμογής data augmentation στα patches

Εικόνα 104: Training – Παράδειγμα εφαρμογής data augmentation στα labels των patches

Εικόνα 105: Training – Βήμα 2 και βήμα 3 της καρτέλας Training

Εικόνα 106: Training – Παράδειγμα τιμών στα βήματα 2 και 3 της καρτέλας εκπαίδευσης

Εικόνα 107: Training – Πρόσθετη ρύθμιση του Dropout layer στα στάδια encoder και bridge

Εικόνα 108: Training – Κουμπί δημιουργίας και ανάλυσης του δικτύου

Εικόνα 109: Training – Αναλυτική παρουσίαση των layers του δικτύου

Εικόνα 110: Training – Κουμπί εκκίνησης εκπαίδευσης του δικτύου που δημιουργήθηκε

Εικόνα 111: Training – Ξεχωριστό παράθυρο παρουσίασης προόδου εκπαίδευσης

Εικόνα 112: Training – Ξεχωριστό παράθυρο παρουσίασης αποτελέσματος εκπαίδευσης

Εικόνα 113: Training Progress – Αναλυτική μορφή παρουσίασης της προόδου εκπαίδευσης

Εικόνα 114: Evaluation – Πεδία προβολής τελικού αποτελέσματος κατάτμησης και ρυθμίσεις

Εικόνα 115: Evaluation – Επιλογή δικτύου για αξιολόγηση

Εικόνα 116: Evaluation – Κουμπί εκκίνησης αξιολόγησης του δικτύου

Εικόνα 117: Evaluation – Προβολή των αποτελεσμάτων του δικτύου Aria Network

Εικόνα 118: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Εικόνα 119: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Εικόνα 120: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης

Εικόνα 121: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνα 122: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

Εικόνα 123: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

Εικόνα 124: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

Εικόνα 125: Evaluation – Επιλογή Weighted pixelClassificationLayer ως final output layer

Εικόνα 126: Evaluation – Προβολή των αποτελεσμάτων του δικτύου Aria Network

Εικόνα 127: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Εικόνα 128: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Εικόνα 129: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης

Εικόνα 130: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνα 131: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

Εικόνα 132: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

Εικόνα 133: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

Εικόνα 134: Evaluation – Επιλογή DicePixelClassificationLayer ως final output layer
Εικόνα 135: Evaluation – Προβολή των αποτελεσμάτων του δικτύου Aria Network
Εικόνα 136: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης
Εικόνα 137: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης
Εικόνα 138: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης
Εικόνα 139: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη
Εικόνα 140: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη
Εικόνα 141: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων
Εικόνα 142: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU
Εικόνα 143: Evaluation – Επιλογή FocalLossLayer ως final output layer
Εικόνα 144: Evaluation – Προβολή των αποτελεσμάτων του δικτύου Aria Network
Εικόνα 145: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης
Εικόνα 146: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης
Εικόνα 147: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης
Εικόνα 148: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη
Εικόνα 149: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη
Εικόνα 150: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων
Εικόνα 151: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU
Εικόνα 152: Evaluation – Επιλογή Weighted pixelClassificationLayer ως final output layer
Εικόνα 153: Evaluation – Προβολή των αποτελεσμάτων του δικτύου Aria Network
Εικόνα 154: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης
Εικόνα 155: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης
Εικόνα 156: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη
Εικόνα 157: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη
Εικόνα 158: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη
Εικόνα 159: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων
Εικόνα 160: Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

Παράδειγμα 6.1: Παράδειγμα κώδικα επιλογής directory path

Παράδειγμα 6.2: Παράδειγμα κώδικα πρόσθεσης τεχνικών σε λίστα με σειρά προτεραιότητας

Παράδειγμα 6.7: Παράδειγμα κώδικα εφαρμογής των τεχνικών μέσα απο τη λίστα με σειρά προτεραιότητας

Παράδειγμα 6.8: Παράδειγμα κώδικα για την υλοποίηση data augmentation

Παράδειγμα 6.9: Παράδειγμα κώδικα για την υλοποίηση των ρυθμίσεων εκπαίδευσης

Παράδειγμα 7.1: Παράδειγμα κώδικα για την αντικατάσταση του ImageInputLayer

Παράδειγμα 7.2: Παράδειγμα κώδικα για την υλοποίηση pixelClassificationLayer

Παράδειγμα 7.3: Παράδειγμα κώδικα για την υλοποίηση Weighted pixelClassificationLayer

Παράδειγμα 7.4: Παράδειγμα κώδικα για την υλοποίηση DicePixelClassificationLayer

Παράδειγμα 7.5: Παράδειγμα κώδικα για την υλοποίηση FocalLossLayer

Παράδειγμα 7.6: Παράδειγμα κώδικα για την υλοποίηση και ρύθμιση DropOut layer

Παράδειγμα 7.7: Παράδειγμα κώδικα για την εκπαίδευση του δικτύου

Παράδειγμα 7.8: Παράδειγμα κώδικα για την μεταφορά της προόδου στη καρτέλα Training Progress

Παράδειγμα 7.9: Παράδειγμα κώδικα για τον υπολογισμό του μέσου όρου

Παράδειγμα 8.1: Παράδειγμα κώδικα για την συγκράτηση της εικόνας για την ελάχιστη/μέγιστη τιμή ανά δείκτη

Τύπος 3.1: Κανονικοποίηση

Τύπος 3.2: Τυποποίηση

Τύπος 6.3: rgb2gray() formula

Τύπος 6.4: Z-Score normalization

Τύπος 6.5: min-max normalization

Τύπος 6.6: 0-255 range

Τύπος 8.2: Accuracy

Τύπος 8.3: Sensitivity

Τύπος 8.4: Specificity

Τύπος 8.5: Precision

Τύπος 8.6: F1 Score

Τύπος 8.7: Intersection over Union

Εισαγωγή

Η ιατρική απεικόνιση αποτελεί μια σημαντική μέθοδο για τη δημιουργία γραφικών αναπαραστάσεων, προσφέροντας εικόνες διαφορετικών περιοχών του ανθρώπινου σώματος που αποτελούν κρίσιμο εργαλείο για την ανίχνευση και την παρακολούθηση παθήσεων. Ποικίλες τεχνολογίες χρησιμοποιούνται στην ιατρική απεικόνιση για να αποκτηθούν πληροφορίες για διάφορα σημεία του σώματος. Για παράδειγμα, η ακτινογραφία είναι κυρίως κατάλληλη για την απεικόνιση οστικών δομών, ενώ η απεικόνιση μαγνητικού συντονισμού παρέχει υψηλής ανάλυσης εικόνες του εγκεφάλου και των εσωτερικών ιστών [1].

Στο κομμάτι αυτό εμπλέκονται και τα συστήματα υποστήριξης αποφάσεων (DSS). Η υποβοηθούμενη διάγνωση από υπολογιστή (CAD) είναι ένα ενδεικτικό παράδειγμα. Τα CAD συστήματα επεξεργάζονται ψηφιακές ιατρικές εικόνες και εφαρμόζουν πολυδιάστατη ανάλυση, επισημαίνοντας προβληματικές περιοχές που απαιτούν περαιτέρω προσοχή. Αυτό αναδεικνύει τη σημαντική συνεισφορά των DSS στο να υποβοηθούν τους ιατρούς στην ταχύτατη διάγνωση και στο decision-making process.

Επιπλέον, τα συστήματα DSS σε συνδυασμό με την τεχνητή νοημοσύνη και την όραση του υπολογιστή, έχουν αναπτυχθεί για την αυτόματη ανίχνευση σοβαρών ασθενειών, όπως όγκοι (στον εγκέφαλο, το μαστό, τους πνεύμονες κ.λπ.) και καρδιαγγειακές παθήσεις, βοηθώντας έτσι στην έγκαιρη διάγνωση και αντιμετώπισή τους.

Συνολικά, η συνεργασία μεταξύ ιατρικής απεικόνισης και συστημάτων υποστήριξης αποφάσεων παρέχει ένα ισχυρό εργαλείο για την αντιμετώπιση των παθήσεων και τη λήψη αποφάσεων, βελτιώνοντας την ποιότητα της φροντίδας των ασθενών [2].

Ο αμφιβληστροειδής αντιπροσωπεύει ένα πολυστρωματικό σχηματισμό στο εσωτερικό του ματιού, υπεύθυνος για τη δημιουργία των εικόνων που αντιλαμβανόμαστε μέσω της όρασής μας, μετατρέποντας τις αντανάκλασεις του φωτός σε νευρικά σήματα, τα οποία στη συνέχεια επεξεργάζεται ο οπτικός φλοιός του εγκεφάλου μας, δημιουργώντας έτσι το πεδίο της όρασης. Τα αιμοφόρα αγγεία του αμφιβληστροειδούς αποτελούν μια κυρίαρχη δομή που μπορεί να παρακολουθηθεί και να αναλυθεί μέσω εξειδικευμένων καμερών βυθού. Η μορφολογία αυτών των αγγείων παίζει σημαντικό ρόλο στην αξιολόγηση διαφόρων

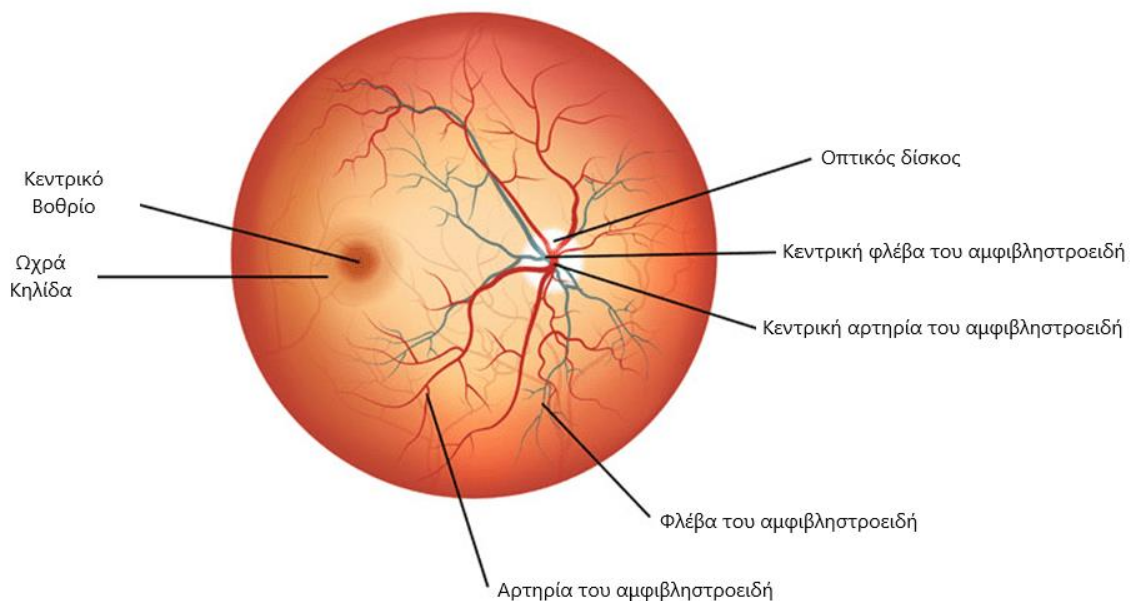
παθήσεων του αμφιβληστροειδούς, όπως η διαβητική αμφιβληστροειδοπάθεια, η υπέρταση και η αρτηριοσκλήρωση [3] [4].

Ωστόσο, η χειροκίνητη ανάλυση αυτών των εικόνων είναι χρονοβόρα και επιρρεπής σε ανθρώπινα λάθη, ειδικά όταν οι δομή – θέση των αγγείων είναι πολύπλοκη. Στο στάδιο αυτό έρχεται το αυτοματοποιημένο λογισμικό που βοηθά τους ιατρούς να επεξεργαστούν τις εικόνες αμφιβληστροειδούς και να διαχωρίσουν τα αγγεία.

Μια αξιόπιστη αυτοματοποιημένη μέθοδος για τη κατάτμηση των αγγείων αμφιβληστροειδούς είναι κρίσιμη για την ακριβή διάγνωση και τη βελτίωση της παρακολούθησης των παθήσεων του αμφιβληστροειδούς [5].

Η ανατομία του ανθρώπινου οφθαλμού

Ο αμφιβληστροειδής



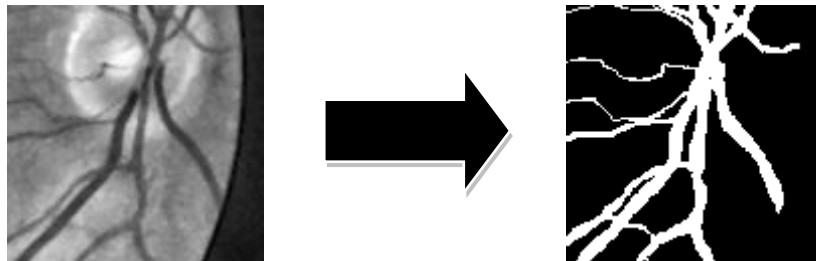
Εικόνα 1: Η ανατομία του αμφιβληστροειδή / [Πηγή](#)

Κεφάλαιο 1: Θεωρητικό υπόβαθρο

Το κρίσιμο βήμα που προηγείται της εξαγωγής αυτών των μορφολογικών χαρακτηριστικών των αγγείων από τις εικόνες του βυθού του ματιού εμπλέκει την κατάτμηση των αγγείων. Αυτή η διαδικασία περιλαμβάνει τη σημασιολογική ετικετοποίηση “labeling” των αγγείων και του φόντου.

1.1 Αντικείμενο Ενδιαφέροντος - Ετικετοποίηση

Ο κύριος στόχος της κατάτμησης των αγγείων στο τομέα του αμφιβληστροειδούς είναι να αντιστοιχίσει μία από τις δύο ετικέτες, δηλαδή, αγγείο “vessel” (1) και φόντο “background” (0), σε κάθε ξεχωριστό pixel εντός της εικόνας [6]. Ως εκ τούτου, αναφέρεται ως ένα πρόβλημα δυαδικής ταξινόμησης. Σε αυτή την εργασία, χρησιμοποιούνται προηγμένες τεχνικές βαθιάς μάθησης, με ένα ενδεικτικό παράδειγμα να είναι τα βαθιά συνελκτικά νευρωνικά δίκτυα (CNNs). Τα CNNs διαθέτουν τη δυνατότητα να αναλύσουν αυτόνομα μια πληθώρα συγκεκριμένων χαρακτηριστικών απευθείας από τα δεδομένα εισόδου (input) [7] [8]. Η εφαρμογή τους αναδεικνύει βελτιώσεις σε πολλά στάδια σε σύγκριση με συστήματα που βασίζονται στη χειροκίνητη εξαγωγή χαρακτηριστικών στον τομέα της οπτικής αναγνώρισης [9].



Εικόνα 2: DRIVE dataset δείγματα / υπό-εικόνες μεγέθους 128 x 128

Στην προτεινόμενη μέθοδο για την κατάτμηση των αγγείων του αμφιβληστροειδούς, ακολουθούνται δύο κύρια στάδια πριν από την κατάτμηση μέσω CNNs [10]: Η αρχική προ-επεξεργασία “preprocessing” της εικόνας γίνεται με μεθόδους:

- Κανονικοποίησης εικόνας “Image Normalization”
- Μετάφραση εικόνας σε κλίμακα του γκρι “Grayscale”
- Εφαρμογή προσαρμοστική εξίσωσης ιστογράμματος περιορισμένης αντίθεσης “CLAHE”
- Αλλαγής μεγέθους της εικόνας “Image Resizing”
- Προαιρετικά: φιλτραρίσματος εικόνας με γκαουσιανά και ενδιάμεσης τιμής φίλτρα “Gaussian Filtering” “Median Filtering”

Μετά την προ-επεξεργασία, εκτελείται η διαδικασία εξαγωγής πλήθους υπό-εικόνων “patches” από κάθε εικόνα του αμφιβληστροειδούς και η επιβολή παραμορφώσεων “augmentation” για την παραγωγή πρόσθετων εικόνων για την εκπαίδευση του δικτύου η οποία γίνεται είτε σε λειτουργία αποθήκευσης αυτών ή σε πραγματικό χρόνο, κατά τη διάρκεια της εκπαίδευσης του δικτύου. Αυτή η προσθήκη δεδομένων είναι ζωτικής σημασίας λόγω του περιορισμένου αριθμού των χειροκίνητα επισημασμένων εικόνων αμφιβληστροειδούς από επαγγελματίες υγείας, για το συγκεκριμένο σετ δεδομένων “dataset”.

Τελικώς, η απόδοση του συστήματος αξιολογείται χρησιμοποιώντας το σύνολο δεδομένων DRIVE [11] με τους δείκτες μέτρησης - αξιολόγησης να περιλαμβάνουν την accuracy, sensitivity, specificity, precision, f1 score, IoU και AUC.

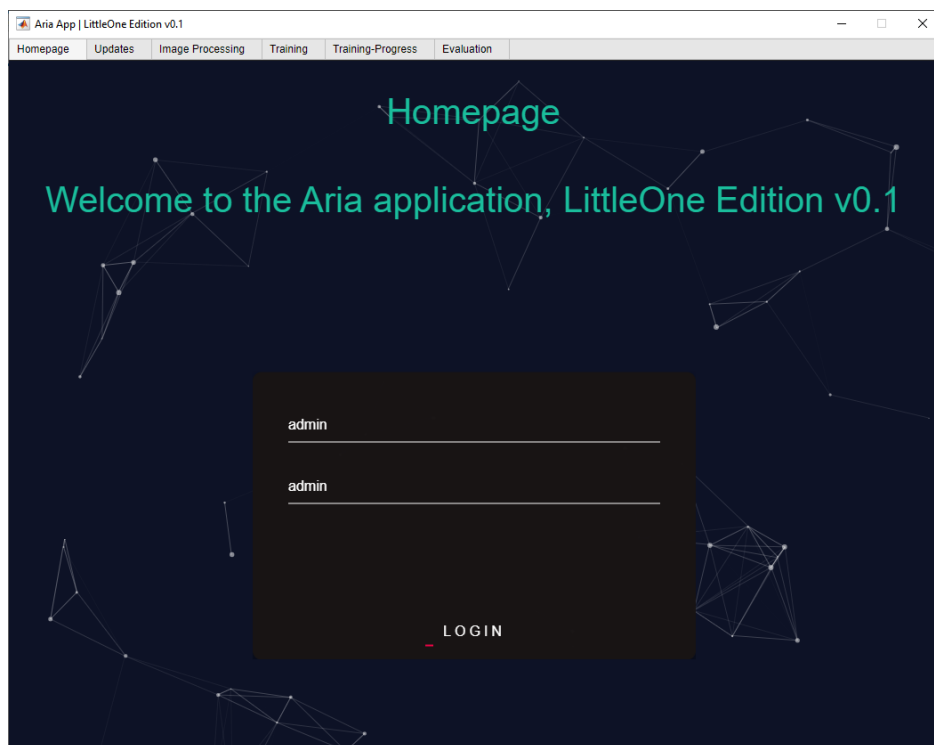
1.2 Αρχιτεκτονικές Βαθιών Συνελικτικών Δικτύων

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs) παρουσιάζουν έντονο ενδιαφέρον λόγω της δυνατότητάς τους να αντλούν αυτόματα πολύπλοκα επίπεδα χαρακτηριστικών από τα πρωτογενή δεδομένα εισόδου. Η εφαρμογή των CNNs έχει εκτείνει τα πεδία της ταξινόμησης, της αναγνώρισης και της κατάτμησης εικόνων. Το επίπεδο χρησιμότητας τους σε σχέση με τις εικόνες απορρέει από την ικανότητά τους να κατανοήσουν περίπλοκες μη αλλοιώσιμες παραμέτρους όπως η κλίμακα και η περιστροφή [12]. Με την εφαρμογή ισχυρών τεχνικών βαθιάς εκμάθησης όπως τα Συνελικτικά Νευρωνικά Δίκτυα (CNNs), τα Εξαιρετικά Συνελικτικά Δίκτυα (FCNs) και πρόσφατα του δικτύου U-Net, έχει επιτευχθεί εξαιρετική απόδοση στην αντιμετώπιση του προβλήματος της κατάτμησης των αιμοφόρων αγγείων του αμφιβληστροειδούς. Παρόλα αυτά, εξακολουθούν να υπάρχουν διάφοροι παράγοντες που δυσκολεύουν την επίτευξη αυτού του στόχου. Μεταξύ αυτών των παραγόντων περιλαμβάνονται η παρουσία μεταβλητής κλίμακας της αντίθεσης της συνολικής εικόνας αλλά και ανά τμήματα, της μεταβλητής έντασης των εικονοστοιχείων μεταξύ διαφορετικών σετ εικόνων, οι ενδο-αγγειακές διαφορές στο πάχος των αγγείων στα διάφορα σημεία της δομής, η παρουσία του οπτικού δίσκου και άλλων παραμορφώσεων, καθώς και η περιορισμένη ποσότητα αξιόπιστων δεδομένων [13].

1.3 Στόχοι – Προσδοκίες

Με την παρούσα εργασία, θα καταβάλουμε μια προσπάθεια στην λύση της χρονοβόρας διαδικασίας της κατάτμησης των αιμοφόρων αγγείων του αμφιβληστροειδούς με τη χρήση μιας απλής, ολοκληρωμένης και έξυπνης εφαρμογής με το όνομα **Aria App | LittleOne Edition** που παρέχει απλουστευμένες τεχνικές επεξεργασίας (ιατρικής) εικόνες, και βασίζεται (προς το παρόν) σε αρχιτεκτονική UNET για την υποβοήθηση τόσο στο κομμάτι της κατάτμησης αλλά και στην καταγραφή δεικτών μέτρησης – αξιολόγησης των αποτελεσμάτων.

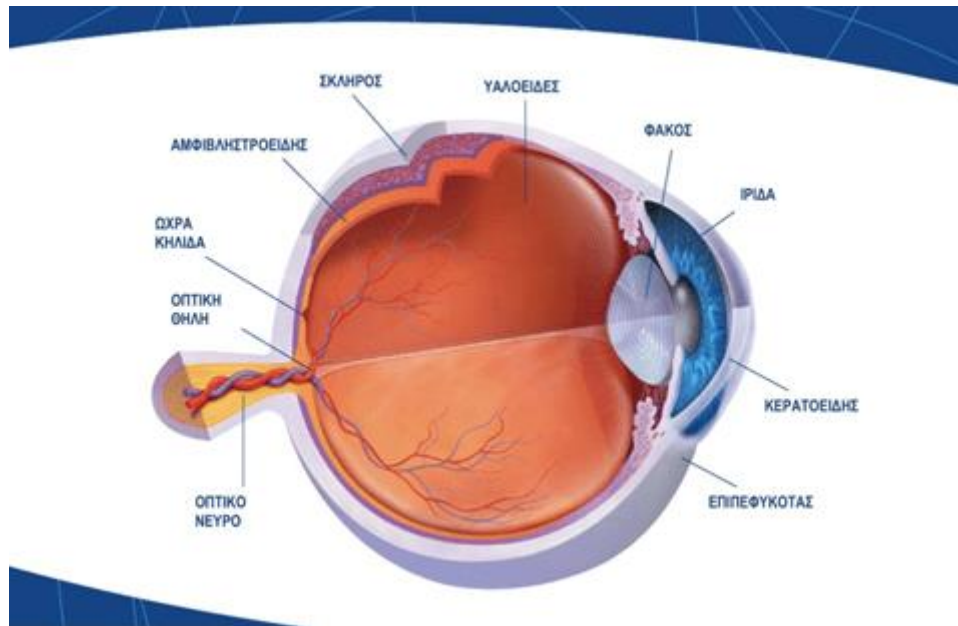
Η ιδέα πίσω από την δημιουργία της εφαρμογής **Aria** προέρχεται και βασίζεται κατά κύριο λόγο στην δυσκολία χρήσης «έξυπνων» επιλογών, που απαιτούν τη γνώση προγραμματισμού και μηχανικής μάθησης. Η χρήση της **Aria** μπορεί να γίνει από κάθε φοιτητή, ερευνητή, ακαδημαϊκό, επαγγελματία υγείας ή μη, ο οποίος ενδιαφέρεται να εφαρμόσει και να δοκιμάσει διάφορες μεθόδους, τεχνικές και συλλογές εικόνων “datasets” από ερευνητικά άρθρα, με βάση την προϋπάρχουσα και αρχική εκδοχή της αρχιτεκτονική δικτύου UNET, χωρίς να απαιτείται ιδιαίτερη τεχνική γνώση στους τομείς της επεξεργασίας (ιατρικής) εικόνες, μηχανικής μάθησης και προγραμματισμού.



Εικόνα 3: Η αρχική οθόνη “homepage” της Aria App, LittleOne edition.

1.4 Κλινική ανατομία του οφθαλμού

Η συνήθως άσπρη σφαίρα του ματιού είναι γνωστή ως η οπτική κυψέλη, η οποία κατέχει ένα διάφανο μέτωπο μέρος που λέγεται κερατοειδής επικάλυψη. Κάτω από το κερατοειδή, υπάρχει η ίριδα, που προσαρμόζει την ποσότητα του φωτός που εισέρχεται στο μάτι, και ο φακός, που επικεντρώνει το φως στο πίσω μέρος του ματιού, όπως παρουσιάζονται στην εικόνα 4 όπου αναπαρίσταται η ανατομία του ματιού.



Εικόνα 4: Ανατομία του οφθαλμού / [Πηγή](#)

Το πίσω μέρος του ματιού, γνωστό ως ρετίνα, είναι το σημείο όπου βρίσκονται τα φωτοευαίσθητα κύτταρα σε μια στρώση πυκνού ιστού, που ονομάζεται αμφιβληστροειδής ιστός. Αυτή η ιστική στρώση, γνωστή ως ο φωτοευαίσθητος χιτώνας, συγκρατείται στο εσωτερικό στρώμα του οφθαλμού, που είναι ο εσωτερικός χιτώνας, με το λεπτό επιθήλιο που αποκαλείται χρωστική αμφιβληστροειδούς στο κέντρο.

Ο εσωτερικός χιτώνας, μια λεπτή μεμβράνη πλούσια σε αιμοφόρες αγγεία, βρίσκεται μεταξύ του αμφιβληστροειδούς και του οπτικού χιτώνα και είναι υπεύθυνος για την παροχή αίματος στον αμφιβληστροειδή. Το εσωτερικό μέρος του ματιού ονομάζεται υαλοειδές σώμα και είναι γεμάτο με διαυγές γέλη που ονομάζεται υαλοειδές υγρό.

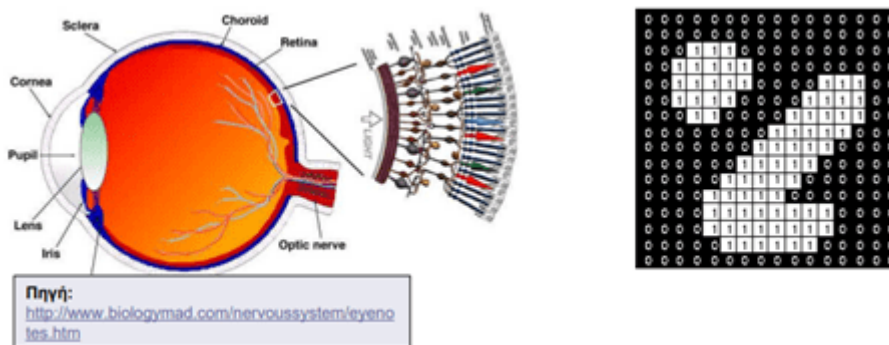
Ο αμφιβληστροειδής αντιπροσωπεύει τον φωτοευαίσθητο χιτώνα του ματιού, παίζοντας το ρόλο ενός φιλμ σε μια φωτογραφική μηχανή. Αποτελεί το πιο σημαντικό ανατομικό μέρος του ματιού, μιας και είναι το αντικείμενο ενδιαφέροντος στο πλαίσιο αυτής της εργασίας. Διάφορες χημικές και ηλεκτρικές διεργασίες λαμβάνουν χώρα εντός του αμφιβληστροειδούς, ενεργοποιούμενες από τα οπτικά στοιχεία του ματιού καθώς εστιάζουν σε μια λαμπερή διέγερση, η οποία είναι μια εικόνα. Τα ηλεκτρικά σήματα που προκύπτουν από αυτήν την ακολουθία γεγονότων μεταφέρονται στον εγκέφαλο μέσω νευρικών ινών, όπου μεταφράζονται σε οπτικές αποτυπώσεις (εικόνες). Ο αμφιβληστροειδής αποτελείται από ποικίλα κύτταρα που εκτελούν διάφορες λειτουργίες, συμπεριλαμβανομένης της μετατροπής της φωτεινής ενέργειας, της προεπεξεργασίας οπτικών πληροφοριών και της μετάδοσης νευρικών σημάτων.

Πως αντιλαμβάνεται τις εικόνες το οπτικό σύστημα

Αισθητήρια κύτταρα

- Κωνία: Ανίχνευση χρώματος (6-7 εκατομμύρια)
- Ραβδία: Ανίχνευση φωτονίων (90-120 εκατομμύρια), περισσότερο ευαίσθητα από τα κωνία, περιφερική όραση, νυκτερινή όραση

•Ψηφιακοί ανιχνευτές!!! 0 ή 1

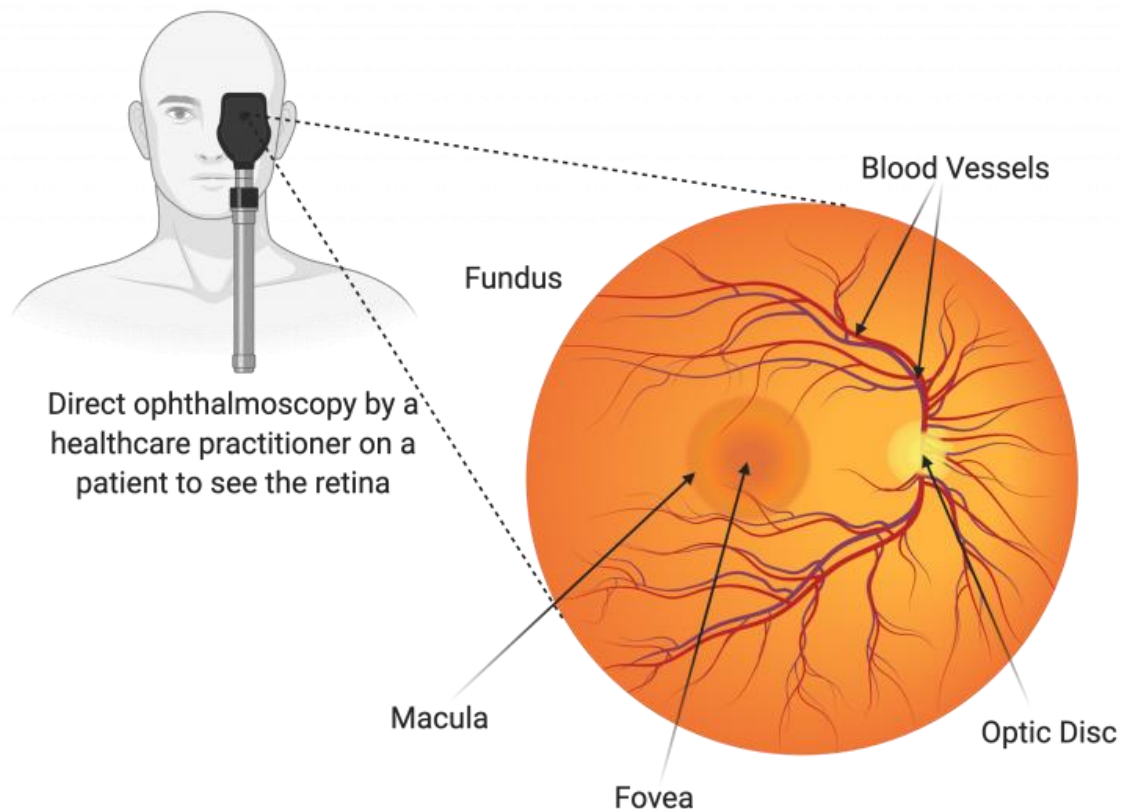


Εικόνα 5: Αντίληψη των εικόνων από το οπτικό σύστημα / Μάθημα ΕΙΕ ([Πηγή](#))

Το στρώμα του αμφιβληστροειδούς αποτελείται από δέκα διαφορετικές στοίβες κυττάρων με περίπλοκες συνδέσεις μεταξύ τους. Δύο σημαντικές πληθυσμιακές ομάδες φωτοδεκτών, τα ραβδία και τα κωνία, βρίσκονται εδώ. Τα ραβδία είναι υπεύθυνα για τη νυκτερινή όραση, ενώ τα κωνία προσφέρουν την ευκρινή όραση.

Η δομή του αμφιβληστροειδούς, όπως παρουσιάζεται στη παρακάτω εικόνα 6, είναι εντοπισμένη στο εσωτερικό μέρος του οπίσθιου τμήματος του ματιού και αναδεικνύεται από μια λεπτή, διαφανή πλάκα ιστού που προέρχεται από το νευροεκτόδερμα.

What does our eye look like when the doctor has a look?



Εικόνα 6: Η ανατομία του αμφιβληστροειδούς / [Πηγή](#)

Αυτή η πλάκα περιλαμβάνει τους αισθητήριους νευρώνες, οι οποίοι αποτελούν την αρχή της οπτικής διαδρομής. Πολλοί νευρώνες συνθέτουν τον νευρικό αμφιβληστροειδή, ο οποίος είναι οργανωμένος σε εννέα στρώματα και αποτελεί τον βασικό παράγοντα για την παραγωγή και μετάδοση ηλεκτρικών παλμών. Οι παλμοί αυτοί μεταφέρονται μέσω νευρικών ινών που συνιστούν τον οπτικό νεύρο στον εγκεφαλικό λοβό, όπου βρίσκονται τα ανώτερα οπτικά κέντρα που αναλαμβάνουν την ανάλυση των εικόνων. Είναι σημαντικό να σημειωθεί ότι οι εικόνες αντικατοπτρίζονται ανάποδα στον αμφιβληστροειδή, με το κάτω ημισφαίριο να αντιπροσωπεύει το άνω οπτικό πεδίο και το ρινικό ημισφαίριο να αντιπροσωπεύει το κροταφικό οπτικό πεδίο [14].

1.5 Απεικόνιση – Λήψη εικόνας του βυθού



Εικόνα 7: Εξοπλισμός – Ψηφιακή κάμερα απεικόνισης του βυθού / [Πηγή](#)

Η απεικόνιση του βυθού χρήζει εξειδικευμένη κάμερα βυθού για την καταγραφή του αμφιβληστροειδούς. Οι ψηφιακές κάμερες απεικόνισης του βυθού, όπως αυτή στην Εικόνα 7, μπορεί να χρησιμοποιούν μια σειρά από ειδικά φίλτρα και βαφές για να γίνει η σύλληψη της κύριας δομής του ματιού με επιτυχία, η οποία αποτελείται από τον οπτικό δίσκο, την ωχρά κηλίδα, τον κεντρικό και περιφερειακό αμφιβληστροειδή.

Οι κατηγορίες αποτύπωσης του βυθού έχουν διαχωριστεί σε διαφορετικούς τρόπους, π.χ.:

- Έγχρωμη αποτύπωση
- Με ή χωρίς κόκκινο φίλτρο
- Αγγειογραφία

Ο αμφιβληστροειδής φωτίζεται με λευκό φως και η εξέταση του γίνεται έγχρωμα σε έγχρωμες φωτογραφίες. Σε περιπτώσεις που οι φωτογραφίες αποτυπώνονται χωρίς κόκκινο, γίνεται χρήση ενός ειδικού πράσινου χρώματος φίλτρου, με μήκος κύματος 540-570 nm του οποίου ο σκοπός

είναι η αποκάλυψη επιφανειακής βλάβης και φυσικά την αφαίρεση του κόκκινου φωτός από το τελικό αποτέλεσμα.

Στη διαδικασία της αγγειογραφίας, πραγματοποιείται έγχυση φθορίζουσας χρωστικής στην κυκλοφορία του αίματος. Η οποία φωτίζει σε διαφορετικά μήκη κύματος, και σε περίπτωση ύπαρξης διαβητικής αμφιβληστροειδοπάθειας, βοηθάει στην ευκρινέστερη παρουσίαση της [15].

1.6 Μηχανική Μάθηση - Machine Learning

Η μηχανική μάθηση, ο ακρογωνιαίος λίθος της τεχνητής νοημοσύνης, εντοπίζει τις ρίζες της στα μέσα του 20ου αιώνα με αξιοσημείωτες συνεισφορές από πρωτοπόρους που έθεσαν τις βάσεις για την ανάπτυξή της. Η έννοια της μηχανικής μάθησης άρχισε να διαμορφώνεται το 1943 όταν οι Warren McCulloch και Walter Pitts παρουσίασαν το πρώτο μοντέλο τεχνητού νευρωνικού δικτύου, εμπνευσμένο από τις αλληλεπιδράσεις των νευρώνων του ανθρώπινου εγκεφάλου [16]. Στα τέλη της δεκαετίας του 1950, ο Arthur Samuel επινόησε τον όρο «μηχανική μάθηση» και πρωτοστάτησε στην έννοια της αυτοβελτίωσης στις μηχανές μέσω της μάθησης από δεδομένα [17]. Η δουλειά του Samuel επικεντρώθηκε στην ανάπτυξη ενός προγράμματος που θα μπορούσε να παίξει πούλια και να βελτιώσει το παιχνίδι του με την πάροδο του χρόνου, σηματοδοτώντας μια από τις πρώτες περιπτώσεις ενός αλγορίθμου μηχανικής μάθησης.

Ένα σημαντικό ορόσημο στη μηχανική μάθηση ήρθε το 1957 όταν ο Frank Rosenblatt έφερε στο φως της πληροφορικής, το perceptron, έναν τύπο τεχνητού νευρωνικού δικτύου σχεδιασμένο για εργασίες αναγνώρισης προτύπων και δυαδικής ταξινόμησης [18]. Αυτό έθεσε τα θεμέλια για μελλοντική έρευνα νευρωνικών δικτύων. Στις δεκαετίες του 1960 και του 1970, η μηχανική μάθηση γνώρισε σημαντικές προόδους με τη μορφή αλγορίθμων δέντρων αποφάσεων “decision trees”, όπως παραδειγματίζεται από το ID3 (Iterative Dichotomiser 3) που εισήχθη από τον Ross Quinlan το 1986 [19].

Η μηχανική μάθηση περιλαμβάνει διάφορα παραδείγματα. Η εποπτευόμενη μάθηση “supervised learning”, μια θεμελιώδης προσέγγιση, περιλαμβάνει την εκπαίδευση ενός μοντέλου σε δεδομένα με ετικέτα για να κάνει προβλέψεις ή ταξινομήσεις. Η μάθηση χωρίς επίβλεψη “unsupervised learning”, από την άλλη πλευρά, εστιάζει στον εντοπισμό προτύπων και δομών μέσα σε δεδομένα χωρίς ετικέτα. Αξιοσημείωτες

τεχνικές χωρίς επίβλεψη περιλαμβάνουν αλγόριθμους ομαδοποίησης όπως το k-means, που εισήχθη από τον Stuart Lloyd το 1957 [20].

Η ενισχυτική μάθηση “reinforcement learning”, που εισήχθη από τον Richard S. Sutton στη δεκαετία του 1980 [21], περιλαμβάνει την εκπαίδευση των παραγόντων να λαμβάνουν διαδοχικές αποφάσεις αλληλεπιδρώντας με ένα περιβάλλον.

Αυτές οι πρώιμες εξελίξεις άνοιξαν το δρόμο για τη μεγάλη ποικιλία τεχνικών μηχανικής μάθησης και αλγορίθμων που έχουμε σήμερα. Η εξέλιξη του πεδίου καθοδηγήθηκε από την πρόοδο στην υπολογιστική ισχύ, τη διαθεσιμότητα δεδομένων και την αλγοριθμική καινοτομία. Από μηχανές γραμμικής παλινδρόμησης και υποστήριξης διανυσμάτων έως τυχαία δάση και ενίσχυση κλίσης, η μηχανική μάθηση έχει εξελιχθεί σε μια ευέλικτη εργαλειοθήκη για την επίλυση σύνθετων προβλημάτων σε διάφορους τομείς, όπως αυτόν που καλύπτει η διπλωματική, τον τομέα της Ιατρικής.

1.7 Βαθιά Μάθηση – Deep Learning

Η βαθιά μάθηση, ένα υποσύνολο της μηχανικής μάθησης και της τεχνητής νοημοσύνης, έχει φέρει επανάσταση σε διάφορους τομείς δίνοντας τη δυνατότητα στους υπολογιστές να μαθαίνουν και να λαμβάνουν αποφάσεις παρόμοια με τις λειτουργίες του ανθρώπινου εγκεφάλου. Στα μέσα της δεκαετίας του 2000 η βαθιά μάθηση απέκτησε σημαντική έλξη, κυρίως λόγω της διαθεσιμότητας τεράστιων συνόλων δεδομένων και της προόδου στην υπολογιστική ισχύ.

Ο όρος «βαθιά μάθηση» διαδόθηκε από τον Geoffrey Hinton και τους συναδέλφους του στις αρχές της δεκαετίας του 2000, αν και η ιδέα μπορεί να αποδοθεί σε παλαιότερη εργασία του Frank Rosenblatt στα τέλη της δεκαετίας του 1950 με τη δημιουργία του perceptron που περιγράψαμε προηγουμένως, μιας και υπήρξε θεμελιώδους μονάδα των νευρωνικών δικτύων. Στην ουσία, σημαντικό ρόλο έπαιξε η πρωτοποριακή έρευνα του Hinton για την εκπαίδευση των βαθιών νευρωνικών δικτύων, ιδιαίτερα η εργασία του για την backpropagation, που πυροδότησε τεράστιο ενδιαφέρον στο πεδίο της μηχανικής-βαθιάς μάθησης και του AI [22].

Το 2006, ο Hinton και η ομάδα του δημοσίευσαν μια εργασία σχετικά με την εκμάθηση ενός δικτύου χωρίς επίβλεψη, σε επίπεδο χαρακτηριστικών, καταδεικνύοντας τη δύναμη των αλγορίθμων βαθιάς μάθησης στην αυτόματη εκμάθηση ιεραρχικής αναπαράστασης δεδομένων.

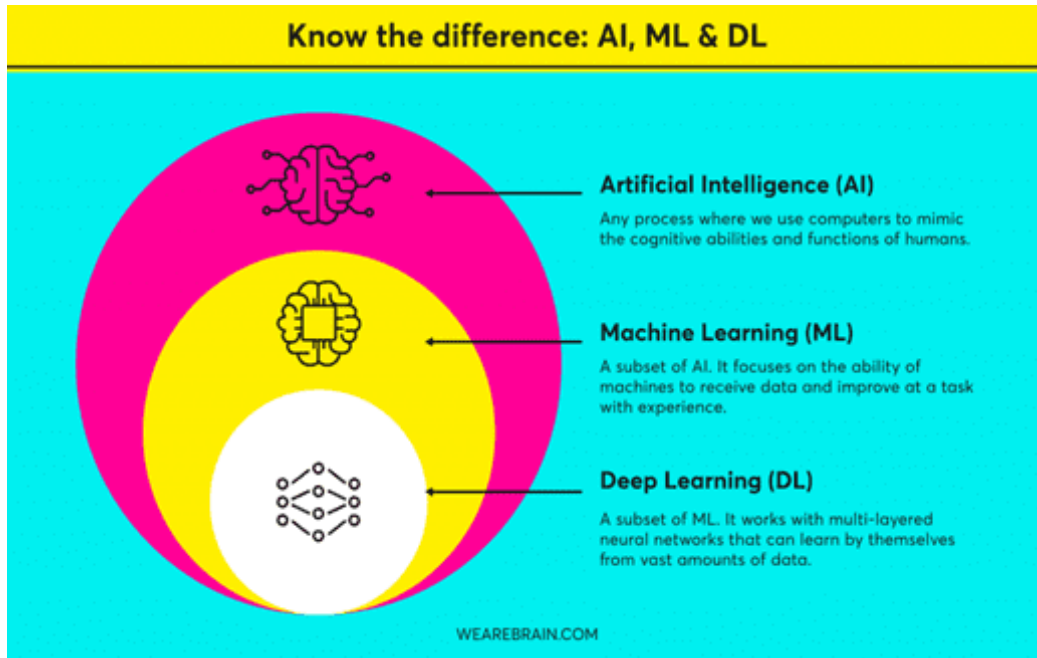
Αυτό σηματοδότησε ένα σημείο καμπής, προκαλώντας ένα ακόμη κύμα ενδιαφέροντος στην έρευνα και τις εφαρμογές που έχει το πεδίο αυτό. Η εισαγωγή των συνελκτικών νευρωνικών δικτύων “convolutional neural networks” (CNN) από τον Yann LeCun, et al, έφερε επανάσταση στην ανάλυση εικόνας και οπτικών δεδομένων, ενώ η ανάπτυξη επαναλαμβανόμενων νευρωνικών δικτύων (RNNs) από τους Sepp Hochreiter και Jürgen Schmidhuber προώθησε σημαντικά τη διαδοχική επεξεργασία δεδομένων [23] [24].

Οι τεχνικές βαθιάς μάθησης έχουν εφαρμοστεί από τότε σε διάφορους τομείς, συμπεριλαμβανομένης της όρασης του υπολογιστή “computer vision”, της επεξεργασίας φυσικής γλώσσας “natural language processing” και της υγειονομικής περίθαλψης. Τα συνελκτικά νευρωνικά δίκτυα έχουν επιτύχει αξιοσημείωτη επιτυχία σε εργασίες ταξινόμησης εικόνων και ανίχνευσης αντικειμένων, με επιτεύγματα ορόσημο όπως το AlexNet [25] το 2012 και μεταγενέστερα μοντέλα όπως το VGG [26], το ResNet [27]. Τα επαναλαμβανόμενα νευρωνικά δίκτυα και οι παραλλαγές τους, όπως τα δίκτυα Long-Short-Term Memory (LSTM), έχουν βελτιώσει σημαντικά τη μοντελοποίηση ακολουθίας, επιτρέποντας εφαρμογές όπως η μετάφραση γλώσσας, η ανάλυση συναισθημάτων και η αναγνώριση ομιλίας [28].

Στην υγειονομική περίθαλψη, η βαθιά μάθηση έχει επιδείξει τις δυνατότητές της στη διάγνωση ιατρικών εικόνων (όπως ακτινογραφίες και μαγνητική τομογραφία), ανακάλυψη φαρμάκων και εξατομικευμένη θεραπεία. Για παράδειγμα, το 2016, ένας αλγόριθμος βαθιάς μάθησης ξεπέρασε τους ανθρώπινους δερματολόγους στον εντοπισμό του καρκίνου του δέρματος από εικόνες. Η ικανότητα του Deep Learning να μαθαίνει αυτόματα χαρακτηριστικά από δεδομένα έχει οδηγήσει σε ανακαλύψεις και στον τομέα των γονιδίων [29].

Συνοπτικά, ο χώρος της βαθιάς μάθησης από τα πρώτα θεμέλιά της, που τροφοδοτείται από πρωτοπόρους όπως η Rosenblatt και η Hinton, έως τις σύγχρονες εφαρμογές της, έχει μεταμορφώσει το τοπίο της τεχνητής νοημοσύνης.

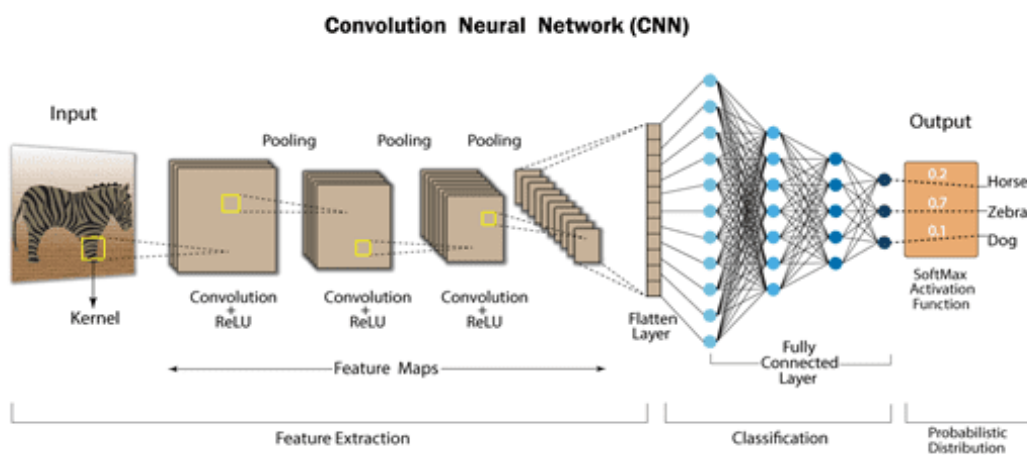
Η έλευση της βαθιάς μάθησης άνοιξε το δρόμο για άνευ προηγουμένου προόδους σε διάφορους τομείς, αξιοποιώντας πολύπλοκες αρχιτεκτονικές νευρωνικών δικτύων για να αποκαλύψει μοτίβα, πρότυπα και ιδέες από δεδομένα που προηγουμένως ήταν αδιανόητα.



Εικόνα 8: Διαφορές – συσχετίσεις μεταξύ των AI, ML, DL / Πηγή

1.8 Συνελικτικά Νευρωνικά Δίκτυα – Convolutional Neural Networks (CNN)

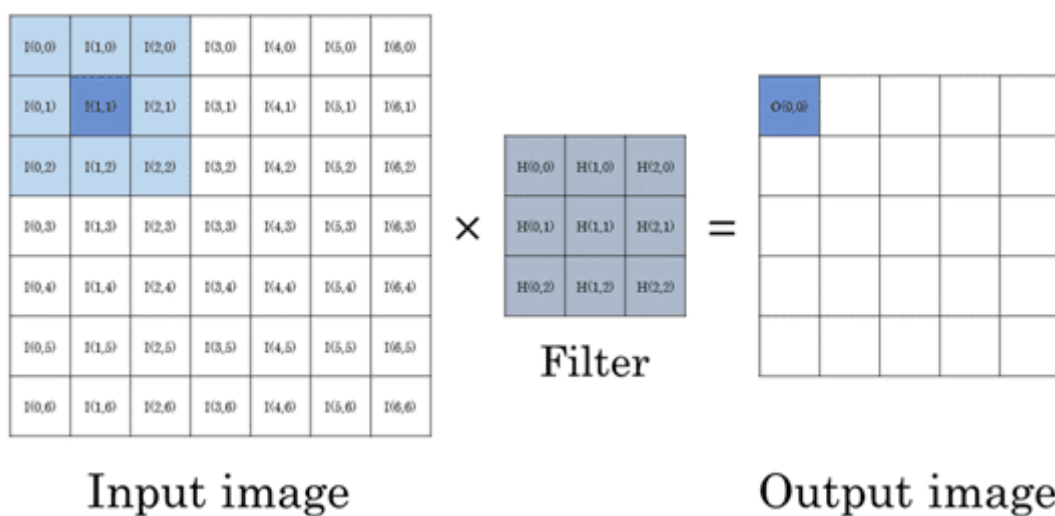
Τα Συνελικτικά Νευρωνικά Δίκτυα (CNN) αποτελούν το αποκορύφωμα της αρχιτεκτονικής καινοτομίας στον τομέα της τεχνητής νοημοσύνης, αναδιαμορφώνοντας το τοπίο της ανάλυσης εικόνας μέσω περίπλοκων μηχανισμών εξαγωγής χαρακτηριστικών, χαρτογράφησης χαρακτηριστικών και εκμάθησης μέσω ιεραρχικής αναπαράστασης.



Εικόνα 9: Παράδειγμα ενός συνελικτικού δικτύου (CNN) / Πηγή

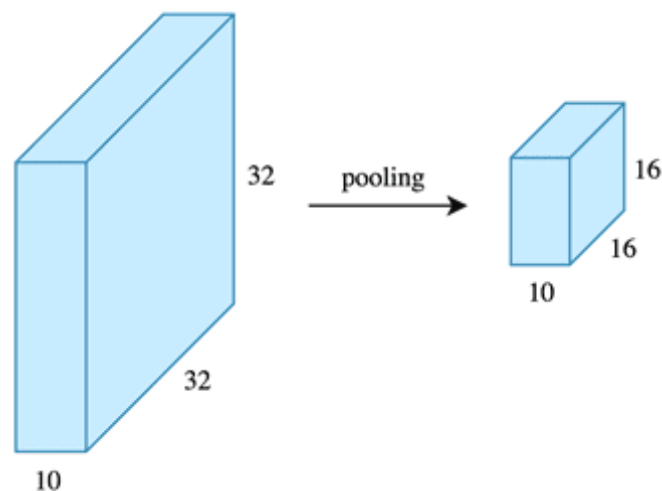
Στον πυρήνα των CNN βρίσκεται μια βασική αρχή: η συνέλιξη. Τα συνελκτικά στρώματα ή επίπεδα “convolutional layers” είναι από τα θεμέλια που παρέχουν την ικανότητα στο δίκτυο να διακρίνει περίπλοκα μοτίβα μέσα στις εικόνες. Αυτά τα επίπεδα χρησιμοποιούν φίλτρα που μπορούν να μάθουν, παρόμοια με πεδία λήψης, για να συνεκκλίνουν στην εικόνα εισόδου. Αυτή η διαδικασία, γνωστή ως εξαγωγή χαρακτηριστικών “feature extraction”, επιτρέπει στο δίκτυο να ανιχνεύει στοιχειώδη χαρακτηριστικά όπως ακμές, υφές και απλά σχήματα. Η λειτουργία συνέλιξης μεγεθύνει την ικανότητα του δικτύου να γενικεύει και να αναγνωρίζει πιο πολύπλοκες δομές καθώς ξεδιπλώνονται τα επόμενα επίπεδα [30].

Η χαρτογράφηση χαρακτηριστικών “feature mapping” είναι ένα άλλο χαρακτηριστικό των CNN, που εκδηλώνεται με τη μορφή χαρτών χαρακτηριστικών “feature maps” ή χαρτών ενεργοποίησης “activation maps”. Κάθε φίλτρο περιπλέκεται σε ολόκληρη την εικόνα, δημιουργώντας έναν χάρτη χαρακτηριστικών που τονίζει την παρουσία ενός συγκεκριμένου χαρακτηριστικού. Αυτοί οι χάρτες χαρακτηριστικών σχηματίζουν συλλογικά ένα περίπλοκο ‘μωσαϊκό’ που αποτυπώνει την ουσία της εικόνας εισόδου. Επιπλέον, η έννοια της συγκέντρωσης, που συνήθως επιτυγχάνεται μέσω λειτουργιών όπως η max-pooling, ενθαρρύνει τη χωρική αναλλοίωση και τη μείωση διαστάσεων. Τα επίπεδα συγκέντρωσης αποστάζουν σημαντικές πληροφορίες, επιτρέποντας στο δίκτυο να εστιάζει στα πιο σχετικά χαρακτηριστικά, ενώ απορρίπτει περιττές λεπτομέρειες.



Εικόνα 10: Παράδειγμα του επιπέδου συνέλιξης / Πηγή

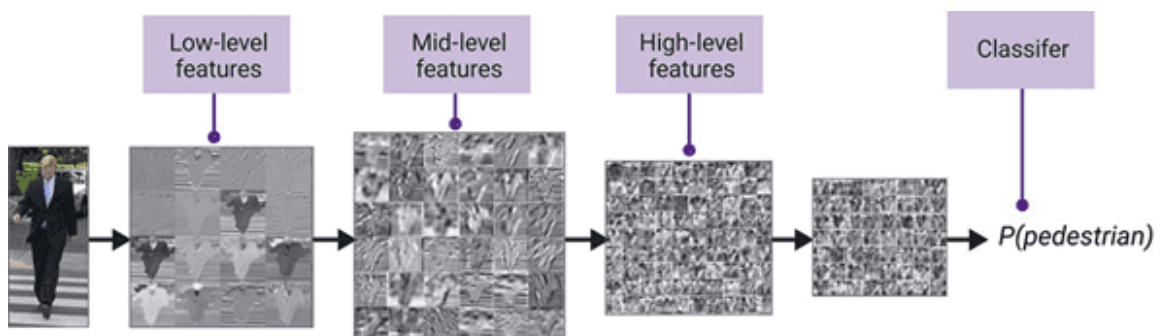
Η μείωση δειγματοληψίας “sample reduction” μέσω συγκέντρωσης είναι ένας βασικός μηχανισμός στα CNN, καθώς μειώνει σταδιακά τις χωρικές διαστάσεις “spatial dimension” των χαρτών χαρακτηριστικών. Αυτή η διαδικασία συμπυκνώνει τις πληροφορίες και ενισχύει την υπολογιστική απόδοση, διευκολύνοντας την ικανότητα του δικτύου να μαθαίνει και να γενικεύει από μεγαλύτερα περιβάλλοντα.



Εικόνα 11: Παράδειγμα υποδειγματοληψίας / [Πηγή](#)

Αντίθετα, τεχνικές υπερδειγματοληψίας “upsampling”, όπως μετατεθειμένες συνελίξεις “transposed convolution” ή παρεμβολή πλησιέστερου γείτονα “nearest neighbor interpolation”, χρησιμοποιούνται σε εργασίες όπως η κατάτμηση εικόνας “image segmentation”, η αποκατάσταση της χωρικής ανάλυσης για ακριβή εντοπισμό.

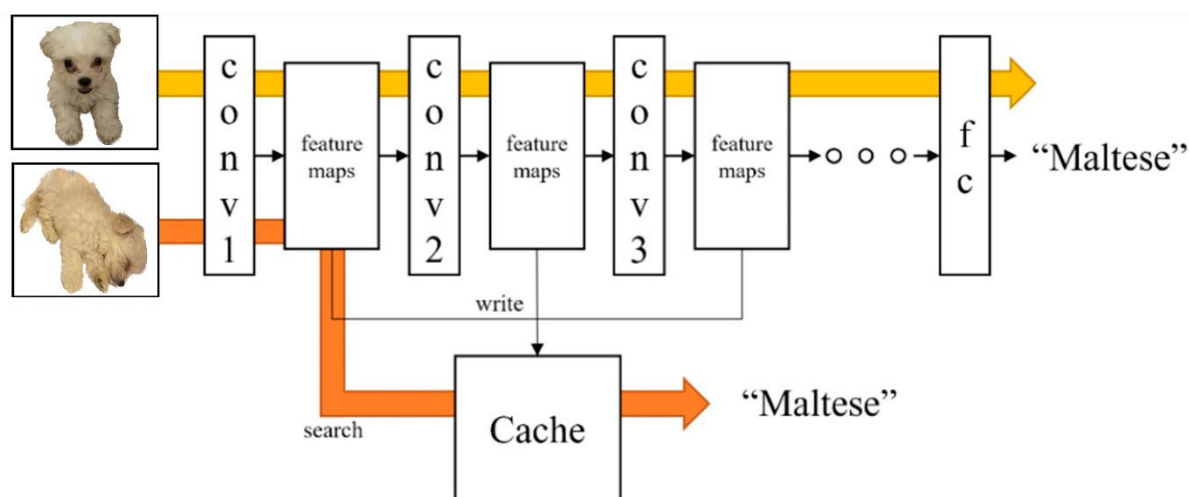
Καθώς το δίκτυο προχωρά μέσω των συνελκτικών επιπέδων του, εμπλέκεται σε χαρτογράφηση πρόβλεψης - μια διαδικασία όπου τα συνοπτικά χαρακτηριστικά “abstract features” συνδυάζονται προοδευτικά για την πρόβλεψη ετικετών - στόχων.



Εικόνα 12: Παράδειγμα των διάφορων επιπέδων χαρακτηριστικών / [Πηγή](#)

Τα πλήρως συνδεδεμένα επίπεδα “fully connected layers”, γνωστά και ως πυκνά στρώματα – επίπεδα “dense layers”, συχνά ακολουθούν τα συνελκτικά επίπεδα, παρέχοντας στο δίκτυο τη δυνατότητα να κάνει προβλέψεις με βάση τις συνοπτικές αναπαραστάσεις που συσσωρεύονται κατά την εξαγωγή και την χαρτογράφηση χαρακτηριστικών «feature extraction and mapping» [31].

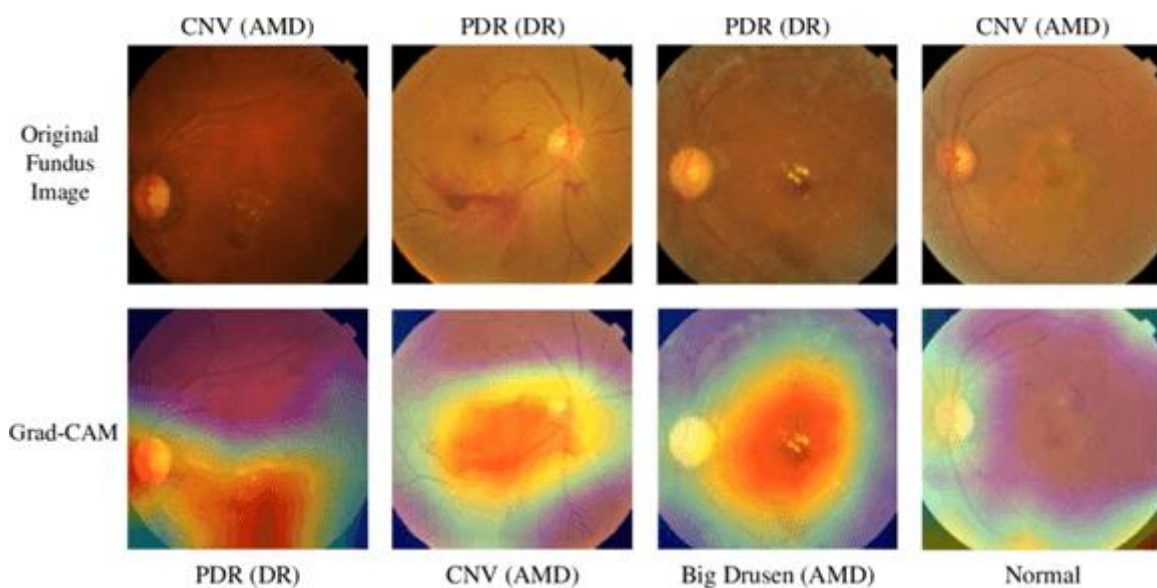
Η ενορχήστρωση αυτών των περίπλοκων μηχανισμών συμβαίνει σε διάφορα επίπεδα στην αρχιτεκτονική του CNN. Στο επίπεδο εισόδου, τα ακατέργαστα δεδομένα pixel υφίστανται διαδοχικούς μετασχηματισμούς μέσω συνελκτικών, ομαδοποιημένων και πλήρως συνδεδεμένων επιπέδων. Κάθε επίπεδο επεξεργάζεται και αφαιρεί χαρακτηριστικά από διαφορετικές κλίμακες, ενισχύοντας τελικά μια λεπτή κατανόηση της εικόνας εισόδου. Το επίπεδο επεξεργασίας εμβαθύνει βαθύτερα, με τα ενδιάμεσα στρώματα να εξάγουν προοδευτικά πολύπλοκες ιεραρχικές αναπαραστάσεις [27].



Εικόνα 13: Παράδειγμα διαδικασίας χαρτογράφησης χαρακτηριστικών / [Πηγή](#)

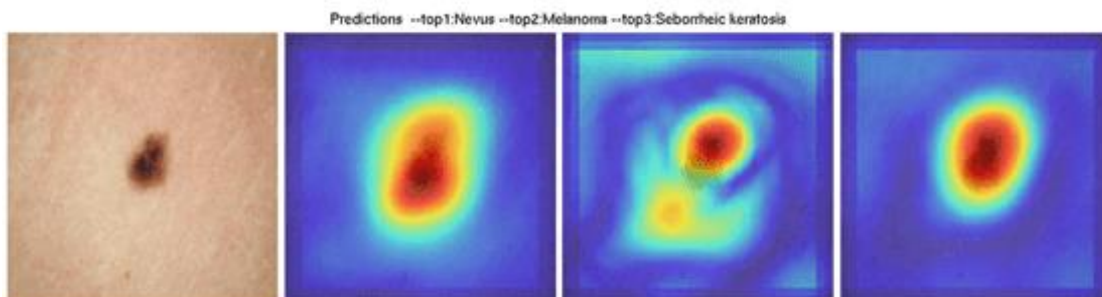
Τα συνελκτικά νευρωνικά δίκτυα αποτελούν την επιτομή στον αρχιτεκτονικό σχεδιασμό, χρησιμοποιώντας τη δυναμική αλληλεπίδραση της συνέλιξης, της χαρτογράφησης χαρακτηριστικών, της μείωσης και της δειγματοληψίας προς τα πάνω. Καθώς το νευρωνικό δίκτυο προχωρά μέσα από τα στρώματά του, ενορχηστρώνει ένα περίπλοκο ‘πακέτο’ εξαγωγής χαρακτηριστικών, με αποκορύφωμα την συναρμολόγηση και αποκρυπτογράφηση του σύνθετου οπτικού περιεχομένου που συλλέχθηκε.

Μετά από αυτό το σημείο, το "activation output map με GRAD CAM" μπορεί να εφαρμοστεί σε αυτό το σύνθετο οπτικό περιεχόμενο που συλλέχθηκε από τα συνελκτικά νευρωνικά δίκτυα. Αυτό σημαίνει ότι, αφού το νευρωνικό δίκτυο παρήγαγε την τελική ταξινόμηση ή την πρόβλεψη, μπορούμε να χρησιμοποιήσουμε το GRAD CAM για να αναλύσουμε ποιες περιοχές της εικόνας είναι υπεύθυνες για αυτή την απόφαση. Με άλλα λόγια μας δίνει μια οπτική αναπαράσταση των σημείων στην εικόνα που είχαν τη μεγαλύτερη επίδραση στην απόφαση του μοντέλου. Αυτό μπορεί να βοηθήσει στην ερμηνεία και την εξήγηση των αποτελεσμάτων του νευρωνικού δικτύου, καθιστώντας πιο σαφές γιατί το μοντέλο έκανε μια συγκεκριμένη ταξινόμηση σε μια εικόνα.



Εικόνα 14: Παράδειγμα activation output map με GRAD CAM / [Πηγή](#)

Η οπτικοποίηση του μετασχηματιστικού αντίκτυπου (prediction and activation) των CNNs, όπως φαίνεται στις εικόνες 14 και 15, αντιστοιχεί σε διάφορους τομείς, κάνοντας πιο εύκολη την κατανόηση των αποτελεσμάτων που προέρχονται από το μοντέλο.



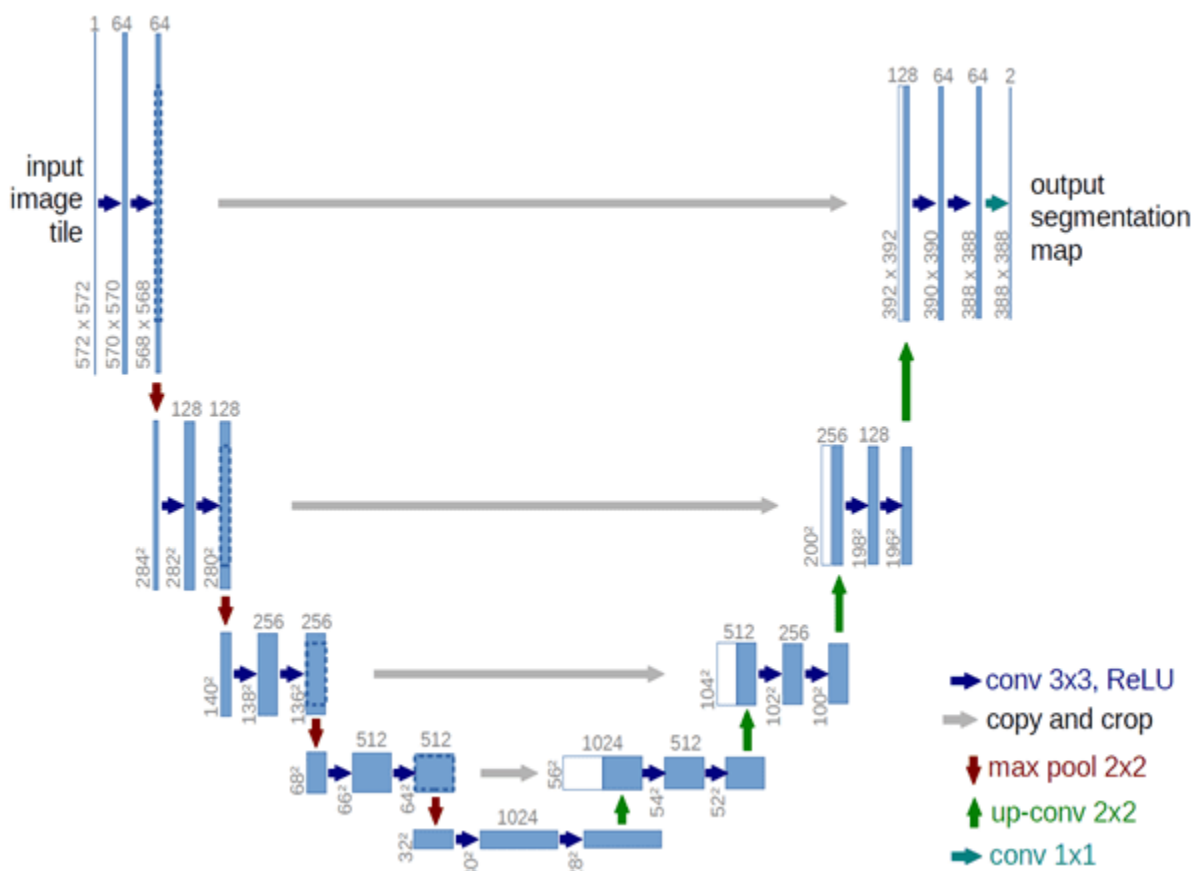
Εικόνα 15: Παράδειγμα activation – feature mapping (έξοδος) / [Πηγή](#)

Κεφάλαιο 2: Δίκτυο U-Net

Το U-Net, που παρουσιάστηκε από τους Olaf Ronneberger, Philipp Fischer και Thomas Brox το 2015, αντιπροσωπεύει μια σημαντική ανακάλυψη στον τομέα της τμηματοποίησης και πλέον της κατάτμησης της ιατρικής εικόνας. Πήρε το όνομά του από τη χαρακτηριστική αρχιτεκτονική του σε σχήμα U.

2.1 Αρχιτεκτονική του δικτύου U-Net

Ο καινοτόμος σχεδιασμός της U-Net αντιμετωπίζει την πρόκληση της κατάτμησης δομών ενδιαφέροντος εντός εικόνων, όπως όργανα ή κακώσεις, αλλοιώσεις ή/και βλάβες, παράγοντας μάσκες ταξινόμησης βάσει εικονοστοιχείων. Η αρχιτεκτονική U-Net έχει βρει εκτεταμένη χρήση στον τομέα της ανάλυσης ιατρικών εικόνων λόγω της αποτελεσματικότητάς της στην τμηματοποίηση δομών ενδιαφέροντος εντός εικόνων [32].



Εικόνα 16: Απεικόνιση της αρχιτεκτονικής του δικτύου U-Net / [Πηγή](#)

2.2 Το U-Net σε εφαρμογές ιατρικού τομέα

Ακολουθούν περισσότερες λεπτομέρειες σχετικά με τη χρήση, τον αντίκτυπο και τα χαρακτηριστικά του στο πλαίσιο των ιατρικών εφαρμογών, χρησιμοποιώντας τα πιο δημοφιλή παραδείγματα [33]:

Ανάλυση Ιατρικής Εικόνας

Σημσιολογική Τμηματοποίηση: Το U-Net χρησιμοποιείται συνήθως για την κατάτμηση ανατομικών δομών ή περιοχών ενδιαφέροντος εντός ιατρικών εικόνων. Αυτό περιλαμβάνει την κατάτμηση οργάνων, όγκων, αιμοφόρων αγγείων και άλλων δομών σε διάφορες μεθόδους ιατρικής απεικόνισης όπως η ακτινογραφία, η αξονική τομογραφία, η μαγνητική τομογραφία και η μικροσκοπηση.

Βιοϊατρική Έρευνα

Το U-Net βοηθά τους ερευνητές και τους κλινικούς γιατρούς στην εξαγωγή ποσοτικών πληροφοριών από ιατρικές εικόνες, επιτρέποντας την πρόοδο στη διάγνωση ασθενειών, τον σχεδιασμό θεραπείας και την κατανόηση των βιολογικών διεργασιών.

Αποκατάσταση και βελτίωση εικόνας

Το U-Net μπορεί να προσαρμοστεί για εργασίες όπως η απαλοιφή θορύβου, η υπέρ-ανάλυση και η ανακατασκευή εικόνας, βελτιώνοντας την ποιότητα των ιατρικών εικόνων για καλύτερη ευκρίνεια, οπτικοποίηση και ανάλυση.

Ανίχνευση και εντοπισμός κακοηθειών

Το U-Net μπορεί να εκπαιδευτεί για την ανίχνευση και τον εντοπισμό ανωμαλιών, όπως η ανίχνευση καρκινικών περιοχών στις μαστογραφίες ή ο εντοπισμός βλαβών, κακώσεων ή αλλοιώσεων σε μαγνητικές τομογραφίες εγκεφάλου.

Αντίκτυπος και Αναφορές στην Ιατρική

Το αποτύπωμα του U-Net, κατά κύριο λόγο στην ανάλυση ιατρικής εικόνας είναι σημαντικός. Έχει υιοθετηθεί ευρέως και αναφέρεται σε πολυάριθμες ερευνητικές εργασίες, καθιστώντας το μια από τις αρχιτεκτονικές με τη μεγαλύτερη επιρροή σε αυτόν τον τομέα.

Το U-Net έχει εφαρμοστεί σε διάφορες εργασίες ιατρικής απεικόνισης, συμπεριλαμβανομένων ενδεικτικά:

- **Τμηματοποίηση MRI εγκεφάλου:** Τμηματοποίηση διαφορετικών εγκεφαλικών ιστών για διάγνωση και έρευνα ασθενειών.
- **Ανάλυση καρδιακής εικόνας:** Ανάλυση δομών και λειτουργιών της καρδιάς από εικόνες μαγνητικής τομογραφίας ή υπερήχων.
- **Ανάλυση εικόνας αμφιβληστροειδούς:** Ανίχνευση και τμηματοποίηση δομών αμφιβληστροειδούς για τη διάγνωση οφθαλμικών ασθενειών.
- **Ανάλυση Εικόνας Παθολογίας:** Προσδιορισμός τύπων και δομών κυττάρων σε ιστοπαθολογικές διαφάνειες.
- **Ενδοσκοπία και Μικροσκοπία:** Ανάλυση εικόνων από ενδοσκόπια και μικροσκόπια για ανίχνευση ασθενειών.

2.3 Αρχιτεκτονική U-Net

Η αρχιτεκτονική U-Net χωρίζεται σε δύο κύρια τμήματα: τη διαδρομή encoder (αριστερή πλευρά του U) και την decoder διαδρομή (δεξιά πλευρά του U), που συνδέεται με ένα κεντρικό σημείο γέφυρας - συμφόρησης. Η διαδρομή down-sampling συλλαμβάνει το πλαίσιο και τα αφηρημένα χαρακτηριστικά, ενώ η upsampling διαδρομή διευκολύνει τον ακριβή εντοπισμό και την τμηματοποίηση.

Το μοντέλο U-Net κληρονομεί χαρακτηριστικά δικτύου από το μοντέλο του SegNet [34], που παρουσιάστηκε έναν χρόνο νωρίτερα. Το πραγματικό πλεονέκτημα και ταυτόχρονα δυνατό σημείο - καινοτομία που εκμεταλλεύεται, αφορά τη χρήση των επιπέδων συνένωσης "concatenation". Σε αυτά τα σημεία - επίπεδα συνένωσης, οι εικόνες που δημιουργήθηκαν στη φάση της διαδρομής του encoder, ενώνονται με τις εικόνες που δημιουργήθηκαν κατά τη φάση της διαδρομής του decoder με αποτέλεσμα να γίνεται καλύτερη ανακατασκευή της τελικής εικόνας από και σε κάθε επίπεδο, μέχρι τη τελική εικόνα. Δημιουργείται επομένως ένα σχήμα της μορφής "U", στο οποίο οφείλεται το όνομα του.

Συγκεκριμένα, η αρχιτεκτονική του δικτύου είναι η εξής [32]:

1. Image Input Layer

2. Διαδρομή encoder:

- **Convolution Layers:** Η εικόνα εισόδου περνά μέσα από μια σειρά συνελκτικών επιπέδων. Αυτά τα επίπεδα εκτελούν συνελίξεις στα δεδομένα εισόδου, εξάγοντας χαρακτηριστικά σε διαφορετικές κλίμακες. Κάθε συνελκτικό επίπεδο εφαρμόζει ένα σύνολο φίλτρων (γνωστά και ως πυρήνες) στην είσοδο, καταγράφοντας διαφορετικές πτυχές της εικόνας.
- **Activation Layers:** Μετά από κάθε συνελκτική λειτουργία, μια συνάρτηση ενεργοποίησης όπως η Διορθωμένη Γραμμική Μονάδα (ReLU) εφαρμόζεται στα στοιχεία για την εισαγωγή μη γραμμικότητας.
- **Pooling Layers:** Περιοδικά, οι χωρικές διαστάσεις των χαρτών χαρακτηριστικών μειώνονται χρησιμοποιώντας επίπεδα συγκέντρωσης (συνήθως max-pooling). Η συγκέντρωση βοηθά στη μείωση δειγματοληψίας των χαρακτηριστικών, μειώνοντας τη χωρική ανάλυση διατηρώντας παράλληλα σημαντικές πληροφορίες.

3. **Bridge - Bottleneck:** Η διαδρομή του encoder τελειώνει με μια διαδρομή – γέφυρα η οποία αντιπροσωπεύει το βαθύτερο επίπεδο της αρχιτεκτονικής. Συνήθως αποτελείται από μια σειρά συνελκτικών επιπέδων που ακολουθούνται από συναρτήσεις ενεργοποίησης. Αυτό βοηθά στη λήψη υψηλού – μεγαλύτερο επιπέδου χαρακτηριστικών πληροφοριών.

4. Διαδρομή decoder:

- **Αντίστροφη συνελίξη (deconvolution):** Η διαδρομή του decoder ξεκινά με αντίστροφα συνελκτικά επίπεδα (επίσης γνωστά ως επίπεδα αποσυνέλιξης ή ανοδικής δειγματοληψίας “upsampling”). Αυτά τα επίπεδα εκτελούν μια αντίστροφη λειτουργία προς τη συνελίξη, αυξάνοντας τη χωρική ανάλυση των χαρτών χαρακτηριστικών.
- **Skip Connections – Επίπεδα Συνένωσης:** Όπως αναφέρθηκε προηγουμένως, μία από τις βασικές καινοτομίες του U-Net είναι η ενσωμάτωση των συνδέσεων παράλειψης μεταξύ των διαδρομών encoder και decoder. Αυτές οι συνδέσεις επιτρέπουν την απευθείας μετάδοση πληροφοριών από προηγούμενα επίπεδα σε μεταγενέστερα

επίπεδα, βοηθώντας στην ανάκτηση των χαμένων χωρικών πληροφοριών, των λεπτών χωρικών λεπτομερειών.

- **Συνένωση:** Σε κάθε στάδιο της διαδρομής του decoder, οι χάρτες χαρακτηριστικών “feature maps” από τις συνδέσεις παράβλεψης “skip connections” συνδέονται με τους χάρτες χαρακτηριστικών από το αντίστοιχο επίπεδο decoder. Αυτός ο συνδυασμός βοηθά στη σύντηξη χαρακτηριστικών χαμηλού και υψηλού επιπέδου.
- **Συνελικτικά επίπεδα:** Οι συνδυασμένοι χάρτες χαρακτηριστικών υποβάλλονται σε συνελικτικές λειτουργίες για να βελτιώσουν τις πληροφορίες και να μειώσουν τις χωρικές διαστάσεις των εικόνων.

5. Επίπεδο εξόδου:

- **Softmax Layer:** Το τελικό επίπεδο εξόδου του U-Net είναι ένα συνελικτικό επίπεδο με λειτουργία ενεργοποίησης softmax. Αυτό δημιουργεί έναν χάρτη τμηματοποίησης που αντιστοιχίζει κάθε pixel στην εικόνα εισόδου σε μια συγκεκριμένη κλάση ή ετικέτα. Ο χάρτης τμηματοποίησης μπορεί να έχει το ίδιο μέγεθος με την εικόνα εισόδου ή μια έκδοση με αλλαγή μεγέθους, ανάλογα με την εργασία και τη συγκεκριμένη υλοποίηση.

Στη διαδρομή γέφυρας “bridge – bottleneck” κατά κύριο λόγο, προαιρετικά τοποθετούνται επίπεδα “layers”, τα οποία ονομάζονται “Dropout Layers” τα οποία χρησιμοποιούνται με σκοπό την αποτροπή – αποφυγή του φαινομένου “overfitting” του μοντέλου, μιας και η λειτουργία του επιπέδου είναι να απενεργοποιεί τυχαία νευρώνες κατά τη διάρκεια της εκπαίδευσης, και αυτό επιτυγχάνεται με το ότι το μοντέλο παύει ή δυσκολεύεται στο να βασιστεί στις εξόδους συγκεκριμένων νευρώνων αλλά να βασιστεί σε μεγαλύτερο φάσμα αυτών. Το dropout layer δέχεται τιμές από 0 έως 1 για ως ποσοστό τυχαίας απενεργοποίησης (π.χ. 0.2 για 20%) και απενεργοποιεί το αντίστοιχο ποσοστό σε αριθμό νευρώνων (20% του συνόλου, σε κάθε επίπεδο).

Κεφάλαιο 3: Προετοιμασία δεδομένων

Η προ επεξεργασία των δεδομένων (στη προκειμένη εργασία, εικόνων) είναι μια ζωτικής σημασίας εργασία [35]. Αποτελεί έναν τρόπο και μια τεχνική εξόρυξης δεδομένων με στόχο την μετατροπή ακατέργαστων δεδομένων σε μία απλούστερη, χρήσιμη, αποτελεσματική και κατανοητή μορφή αυτών, το οποίο επιτρέπει στην πιο ουσιαστική και πετυχημένη ανάλυση από τους υπολογιστές και τη μηχανική μάθηση (δίκτυο).

3.1 Σημασία της επεξεργασίας των δεδομένων

Στη πλειοψηφία των περιπτώσεων, τα δεδομένα που έχουν συλλεχθεί και είναι έτοιμα για χρήση από συστήματα - μοντέλα μηχανικής μάθησης, έρχονται από τη φύση τους σε αριθμητική μορφή, κάτι το οποίο τα καθιστά εύκολα στη χρήση. Ένα παράδειγμα αυτών, είναι οι θερμοκρασίες που έχει συλλέξει ένας αισθητήρας του μετεωρολογικού σταθμού σε μια περίοδο (x), ή οι αιματολογικές εξετάσεις ασθενών σε μια συγκεκριμένη κλινική για κάποιο διάστημα ή από συγκεκριμένο τμήμα. Σε τέτοιες περιπτώσεις όπου η φύση των δεδομένων είναι εξ αρχής αριθμητική, δεν χρειάζεται ιδιαίτερη φροντίδα - επεξεργασία αυτών για τη τροφοδότηση τους στο σύστημα.

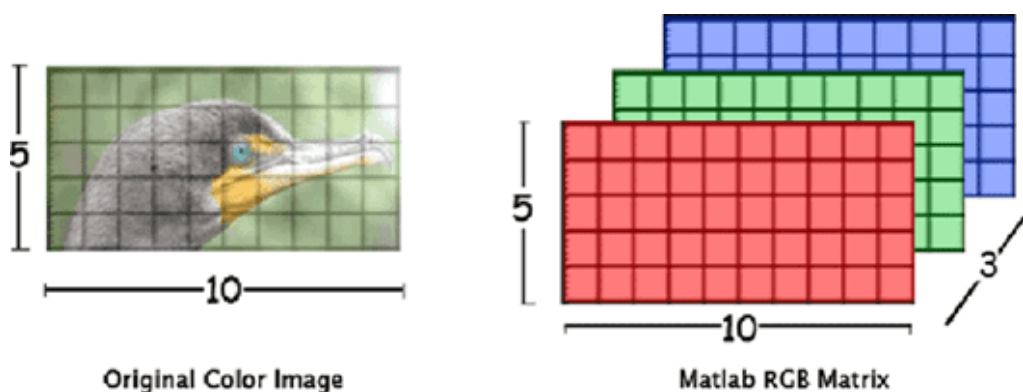
Σε άλλες περιπτώσεις όμως, τα δεδομένα δεν μπορούν να χρησιμοποιηθούν με την ίδια ευκολία λόγω της μορφής στην οποία βρίσκονται και χρήζουν επεξεργασίας έτσι ώστε να είναι αριθμητικά "αποδεκτά", με τέτοιο τρόπο που διευκολύνει το σύστημα να μάθει μέσα από αυτά. Κάποια παραδείγματα αυτών, είναι π.χ. όπως στη παρούσα εργασία, η χρήση εικόνων για την υποβοήθηση στη κατάτμηση αγγείων αλλά και σε άλλες κατηγορίες δεδομένων όπως τα κείμενα και ο ήχος. Ο υπολογιστής σε κάθε μια από αυτές τις περιπτώσεις (εικόνα, κείμενο, ήχος) θα μπορούσε να αντιστοιχήσει το περιεχόμενο αυτών με αριθμητικές τιμές (π.χ. δυαδικό σύστημα, 0 ή 1) αλλά η πληροφορία και το σημείο-αντικείμενο ενδιαφέροντος δεν θα μπορεί να γίνει άμεσα ή και καθόλου αντιληπτή από το σύστημα μηχανικής μάθησης κάτι το οποίο θα κάνει το μοντέλο να ταχθεί προς την πιο εύκολη λύση, που συνήθως είναι να μάθει καλά μια πληροφορία που εμφανίζεται περισσότερο (π.χ. το παρασκήνιο μιας εικόνας), το οποίο συνήθως θεωρείται χαρακτηριστικό χαμηλού επιπέδου - ενδιαφέροντος σε διάφορες εργασίες.

Επομένως, για την επίτευξη υψηλών αποτελεσμάτων σε μετρήσεις ακρίβειας και άλλων μέτρων αξιολόγησης που θα επεξηγηθούν στα

επόμενα κεφάλαια, η προ-επεξεργασία των δεδομένων, ειδικά όταν από τη φύση τους δεν έρχεται σε αριθμητική μορφή, είναι εξαιρετικά σημαντική, και αυτό περιλαμβάνει διάφορες μεθόδους και στάδια και ενδείκνυται για όλα τα συστήματα και μοντέλα, και αυτά της βαθιάς μάθησης, μιας και τα αποτελέσματα δείχνουν ότι μια επαρκή προ-επεξεργασία των δεδομένων πριν την τροφοδότηση, επηρεάζει αρκετά τη τελική εργασία του συστήματος. Οι διάφορες μέθοδοι και στάδια, θα αναφερθούν παρακάτω.

3.2 Επεξεργασία εικόνων – Image Preprocessing

Η εικόνα είναι μια μήτρα, ή πιο συγκεκριμένα ένας ή παραπάνω πίνακας που σε κάθε κελί $M(i, j)$ έχουν τιμές 0-255 σε περίπτωση που είναι RGB (έγχρωμη εικόνα), 0-7 σε περίπτωση που είναι γκρι (ασπρόμαυρη με βάθος του γκρι) και 0-1 σε περίπτωση που είναι δυαδική, δηλαδή τα εικονοστοιχεία έχουν τιμή μαύρο ή άσπρο.



Εικόνα 17: Παράδειγμα πινάκων μιας RGB εικόνας / [Πηγή](#)

Μια εικόνα $Υ \times Π$ σε μοντέλο RGB, αναπαρίσταται βάσει 3 πινάκων (τριδιάστατος πίνακας) του κόκκινου, πράσινου και μπλε, οι οποίοι πίνακες είναι δυο διαστάσεων ($Υ \times Π$), φέρνοντας έτσι λοιπόν την τελική εικόνα που βλέπουμε σε Ύψος x Πλάτος x 3. Κάθε κελί του εκάστοτε πίνακα λαμβάνει μια τιμή από το εύρος $R = \{0..255\}$ με το 0 να είναι η Ελάχιστη ένταση (μαύρο) και το 255 τη μέγιστη ένταση (λευκό), που δηλώνει την ένταση του συγκεκριμένου κελιού - χρώματος (κόκκινο, πράσινο και μπλε), καταλήγοντας σε ένα εικονοστοιχείο, βάσει των τριών τιμών.

3.2.1 Μετατροπή του χρώματος της εικόνας

Στις ιατρικές εφαρμογές, ανάλογα με το σκοπό – στόχο της εργασίας που καλείται να πραγματοποιήσει ένα μοντέλο, μπορούν να χρησιμοποιηθούν διάφορες μέθοδοι μετατροπής χρώματος, όπως η μετατροπή από μοντέλο RGB σε κλίμακα του γκρι - (εικόνα μονού καναλιού) – “Grayscale”:

Οι εικόνες RGB μετατρέπονται σε κλίμακα του γκρι λαμβάνοντας τον μέσο όρο των καναλιών (πινάκων) του κόκκινου, πράσινου και μπλε. Οι εικόνες σε κλίμακα του γκρι περιέχουν μόνο ένα κανάλι και χρησιμοποιούνται συχνά για να απλοποιήσουν τους υπολογισμούς και να τονίσουν τις συνολικές διακυμάνσεις της έντασης. Ορισμένες εργασίες όμως μπορούν να χρησιμοποιήσουν μόνο την πληροφορία έντασης του πράσινου καναλιού “green channel”, όπως το αντικείμενο της παρούσας εργασίας, καθώς είναι το κανάλι που παρουσιάζει Μέγιστη αντίθεση μεταξύ των αιμοφόρων αγγείων του αμφιβληστροειδούς και του φόντου. Στη προκειμένη περίπτωση, τα κανάλια του κόκκινου και του μπλε, εμφανίζουν πιο χαμηλή αντίθεση συγκριτικά με το πράσινο και μεγαλύτερη διακύμανση στην ένταση της φωτεινότητας.

3.2.2 Τεμαχισμός εικόνας – Image Patch Extraction

Ο τεμαχισμός των εικόνων, γνωστός και ως “patches”, αναφέρεται στον διαχωρισμό μιας μεγάλης εικόνας υψηλής ανάλυσης σε μικρότερες, πιο μικρές περιοχές που ονομάζονται τεμάχια, παράθυρα ή τμήματα. Αυτή η τεχνική είναι ευρέως χρησιμοποιούμενη στην επεξεργασία και ανάλυση της εικόνας, συμπεριλαμβανομένης της ανάλυσης ιατρικών εικόνων, και έχει πολλές εφαρμογές και πλεονεκτήματα [36]:

- **Διευκόλυνση Επεξεργασίας:** Ο τεμαχισμός επιτρέπει την επεξεργασία μικρότερων περιοχών μεμονωμένα, αντί για ολόκληρη την εικόνα. Αυτό μπορεί να μειώσει τον υπολογιστικό φόρτο και να βελτιώσει την απόδοση του αλγορίθμου.
- **Εξαγωγή Χαρακτηριστικών:** Με τον τεμαχισμό, μπορούμε να εξάγουμε χαρακτηριστικά από μικρές περιοχές της εικόνας, όπως σημεία ενδιαφέροντος ή δομές. Αυτό μπορεί να βελτιώσει την ακρίβεια της ανάλυσης.
- **Επεξεργασία Δεδομένων σε Τμήματα:** Σε πολλές περιπτώσεις, η επεξεργασία δεδομένων σε μικρότερα τμήματα μπορεί να έχει πιο εύκολη ερμηνεία ή εφαρμογή.

- **Επίτευξη Καλύτερης Εξισορρόπησης Χρωμάτων:** Σε ορισμένες εφαρμογές, ο τεμαχισμός μπορεί να επιτρέψει την εφαρμογή διαφορετικών εξισορροπήσεων χρωμάτων σε διάφορα τμήματα της εικόνας.
- **Εξισορρόπηση και Προσθήκη δεδομένων:** Με τη χρήση των patches, μπορεί να επιτευχθεί η εξισορρόπηση και η προσθήκη δεδομένων υπό εφαρμογή διάφορων σταδίων επεξεργασίας και φιλτραρίσματος σε datasets που δεν έχουν επαρκή αριθμό εικόνων για κάθε κλάση ή μια είναι πολλές φορές μεγαλύτερη σε ποσοστό εμφάνισης από την άλλη.

Στις εικόνες του αμφιβληστροειδούς, η χρήση της τεχνικής patch extraction, δηλαδή της εξαγωγής τεμαχίων από τις εικόνες, έχει τα εξής οφέλη:

- Αντιμετωπίζεται το πρόβλημα του μεγάλου μεγέθους και μεγάλων διαστάσεων των εικόνων καθώς η τεχνολογία εξελίσσεται και οι φωτοευαίσθητοι αισθητήρες τείνουν να μεγαλώνουν με στόχο την αύξηση της ευκρίνειας και της συνολικής ποιότητας της τελικής εικόνας.
- Μειώνεται η ανισορροπία του αριθμού των εικόνων για κάθε κλάση (αγγείο / φόντο) ενώ παράλληλα προσθέτει εικόνες στο τελικό dataset με την εφαρμογή διάφορων φίλτρων και σταδίων επεξεργασίας, καθώς οι διαθέσιμες (δημόσιες) βάσεις περιέχουν μικρό αριθμό εικόνων στο πεδίο αυτό.

Συνολικά, ο τεμαχισμός των εικόνων είναι μια χρήσιμη τεχνική που βοηθά στην επεξεργασία, ανάλυση και εξαγωγή πληροφοριών από εικόνες, ιδίως όταν αυτές είναι μεγάλες ή περιέχουν σύνθετες δομές [37].

3.2.3 Κανονικοποίηση – Standardization

Η κανονικοποίηση και η τυποποίηση είναι δύο τεχνικές προεπεξεργασίας δεδομένων που χρησιμοποιούνται σε διάφορα πεδία, συμπεριλαμβανομένης της αναγνώρισης προτύπων, της επεξεργασίας εικόνων και της ανάλυσης αμφιβληστροειδούς. Αν και συνήθως χρησιμοποιούνται για διαφορετικούς σκοπούς, έχουν ομοιότητες και διαφορές [38].

Κανονικοποίηση:

Η κανονικοποίηση αναφέρεται στη διαδικασία μετατροπής των τιμών ενός χαρακτηριστικού ή μιας μεταβλητής έτσι ώστε να βρίσκονται εντός ενός συγκεκριμένου εύρους. Αυτό μπορεί να επιτευχθεί με την κλιμάκωση των τιμών ή τη μετατροπή τους σε ποσοστιαία κλίμακα. Ο σκοπός είναι να εξασφαλιστεί η σύγκριση ή η επεξεργασία δεδομένων σε μια πιο συνεπή και συγκρίσιμη βάση. Στον τομέα του αμφιβληστροειδούς, η κανονικοποίηση μπορεί να χρησιμοποιηθεί για την προ-επεξεργασία των μετρήσεων της διαμέτρου του αμφιβληστροειδούς προκειμένου να εξασφαλιστεί ότι οι τιμές ταιριάζουν σε ένα κοινό πλαίσιο ανεξάρτητα από τις μεθόδους μέτρησης.

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad \text{Τύπος 3.1}$$

Τυποποίηση:

Η τυποποίηση, γνωστή και ως Z-score standardization, αναφέρεται στη διαδικασία μετατροπής των τιμών ενός χαρακτηριστικού ή μιας μεταβλητής έτσι ώστε να έχουν μέσο όρο 0 και τυπική απόκλιση 1. Αυτό επιτυγχάνεται αφαιρώντας τον μέσο όρο και διαιρώντας με την τυπική απόκλιση. Η τυποποίηση μπορεί να βοηθήσει στην αποφυγή προβλημάτων όπως η υπερβολική επίδραση των μεγάλων τιμών. Στο πλαίσιο της αμφιβληστροειδούς, η τυποποίηση μπορεί να χρησιμοποιηθεί για την επεξεργασία των μετρήσεων του εσωτερικού πίεσης του ματιού, καθιστώντας ευκολότερη τη σύγκριση μεταξύ διαφορετικών ατόμων.

$$x' = \frac{x - \bar{x}}{\sigma} \quad \text{Τύπος 3.2}$$

3.2.4 Εξισορρόπηση Ιστογράμματος – CLAHE

Η εξισορρόπηση ιστογράμματος, γνωστή και ως CLAHE (Contrast Limited Adaptive Histogram Equalization), είναι μια παραλλαγή της προσαρμοστικής ισοστάθμισης ιστογράμματος που φροντίζει για την υπερβολική ενίσχυση της αντίθεσης με σκοπό τη μείωση του ενισχυμένου θορύβου. Αυτό το κάνει κατανέμοντας εκείνο το τμήμα του ιστογράμματος που υπερβαίνει το όριο 'clip' εξίσου σε όλα τα ιστογράμματα. Η βασική ιδέα της είναι να εξισορροπήσει το ιστόγραμμα της εικόνας, δηλαδή την κατανομή των φωτεινότητων, προκειμένου να βελτιστοποιήσει την ορατότητα λεπτομερειών που μπορεί να χαθούν λόγω ανεπαρκούς αντίθεσης.

Η τεχνική CLAHE λειτουργεί ως εξής:

- Διαιρεί την εικόνα σε μικρά τμήματα, γνωστά ως παράθυρα.
- Για κάθε παράθυρο, εκτελεί την κανονικοποίηση του ιστογράμματος (CDF), η CDF αναπαριστά τη συσσώρευση της συχνότητας των τιμών των pixel. Στη συνέχεια, οι τιμές φωτεινότητας στο ιστόγραμμα προσαρμόζονται με το να τις αντιστοιχίσει σε νέες τιμές βάσει της CDF. Αυτή η διαδικασία αναδιανέμει τις τιμές των pixel, περιορίζοντας την αύξηση της αντίθεσης ώστε να αποφευχθεί η υπερβολική ενίσχυση θορύβου.
- Στη συνέχεια, συνθέτει τα επεξεργασμένα παράθυρα πίσω χρησιμοποιώντας διγραμμική παρεμβολή (binary interpolation) για την αφαίρεση των τεχνητών ορίων και δημιουργεί τη βελτιωμένη εικόνα.

Η τεχνική CLAHE είναι ιδιαίτερα χρήσιμη σε περιπτώσεις όπου υπάρχουν μεγάλες διακυμάνσεις στη φωτεινότητα και η εικόνα παρουσιάζει περιοχές με υψηλή και χαμηλή αντίθεση ταυτόχρονα. Είναι ευρέως χρησιμοποιούμενη σε εφαρμογές επεξεργασίας εικόνων, ιατρικής εικονικής απεικόνισης (όπως ανάλυση αμφιβληστροειδούς), επεξεργασίας φωτογραφιών, και άλλες εφαρμογές που απαιτούν βελτιωμένη οπτική αντίθεση και λεπτομέρεια [39].

3.2.5 Επαύξηση των δεδομένων – Data Augmentation

Η διαδικασία του data augmentation στην επεξεργασία εικόνων αναφέρεται στον τρόπο με τον οποίο αυξάνουμε το σύνολο δεδομένων μας προσθέτοντας ποικιλία στις υπάρχουσες εικόνες μέσω μιας σειράς μετασχηματισμών. Αυτό το είδος επεξεργασίας είναι κοινό στη μηχανική μάθηση, ιδίως όταν ασχολούμαστε με εκπαίδευση μοντέλων νευρωνικών δικτύων.

Ο στόχος του data augmentation είναι να βελτιώσει τη γενίκευση και την απόδοση του μοντέλου μας. Προσθέτοντας ποικιλία στις εικόνες μας, το μοντέλο μας μαθαίνει να αντιμετωπίζει διάφορες συνθήκες, όπως αλλαγές στον φωτισμό, την περιστροφή, την αλλαγή κλίμακας, την αλλαγή προοπτικής κ.λπ. Η προσθήκη νέων δεδομένων, ύστερα από εφαρμογή μετασχηματισμών στο σύνολο των δεδομένων εκπαίδευσης (training dataset), ειδικά όταν αυτό είναι περιορισμένο είτε στη μία ή σε όλες τις κλάσεις, βοηθάει το μοντέλο να αποφεύγει το φαινόμενο της υπερπροσαρμογής “overfitting” του στα δεδομένα εκπαίδευσης, το οποίο μπορεί να παρουσιαστεί σε πολλαπλά σενάρια, αλλά κυρίως όταν ο αριθμός των εισαχθέντων (input) δεδομένων είναι πολλές φορές μικρότερος από τον αριθμό των παραμέτρων που μπορεί να συγκεντρώσει το μοντέλο (learning parameters). Οι μετασχηματισμοί που μπορούν να εφαρμοστούν κατά το data augmentation περιλαμβάνουν [40]:

- **Περιστροφή:** Περιστροφή της εικόνας κατά τυχαία γωνία.
- **Ανακατανομή Χρωματικών Καναλιών:** Αλλαγή των χρωματικών καναλιών της εικόνας.
- **Κατακόρυφος και Οριζόντιος Ταξινομικός Χειρισμός:** Αναστροφή της εικόνας οριζόντια ή κατακόρυφα.
- **Αλλαγή Κλίμακας:** Μεταβολή του μεγέθους της εικόνας.
- **Περικοπή:** Τυχαία περικοπή μιας περιοχής της εικόνας.
- **Ανακατανομή Φωτεινότητας:** Τυχαία αλλαγή της φωτεινότητας

Είναι σημαντικό να επιλέξετε τους μετασχηματισμούς που είναι συνεπείς με τη φύση των δεδομένων σας και την εφαρμογή σας. Η σωστή εφαρμογή του data augmentation μπορεί να βελτιώσει την ικανότητα γενίκευσης του μοντέλου σας και να βοηθήσει στην απόδοσή του σε νέα, ανεξάρτητα δεδομένα.

Κεφάλαιο 4: Εφαρμογή Aria

Σε αυτό το κεφάλαιο, θα παρουσιάσουμε την εφαρμογή Aria, στην έκδοση LittleOne. Η έκδοση LittleOne είναι η βασική και μοναδική έκδοση της εφαρμογής, η οποία σηματοδοτεί και την αρχή και το τέλος κατά τη συγγραφή αυτής της διπλωματικής εργασίας.

4.1 Παρουσίαση εφαρμογής Aria App | LittleOne Edition

Η Aria, δημιουργήθηκε τόσο στη θεωρία αλλά και μετέπειτα στη πράξη, ως ένα εργαλείο υποβοήθησης στην προ-επεξεργασία/επεξεργασία των δεδομένων, στο τομέα της οφθαλμολογίας και πιο συγκεκριμένα στην ευκρινής εμφάνιση των αγγείων του αμφιβληστροειδούς, μια διαδικασία η οποία μπορεί να είναι χρονοβόρα και πολύπλοκη, όταν απαιτείται γνώση προγραμματισμού. Σε συνέχεια της υλοποίησης της Aria, μέσα από συζητήσεις και διάφορες δοκιμασίες της εφαρμογής από την τελειόφοιτη γιατρό Περσίλα Κουμάτη, ήταν άμεσο το ενδιαφέρον για τη χρήση της εφαρμογής αυτής σε μια μορφή όπου θα μπορούσαν επαγγελματίες υγείας να πειραματίζονται με το ίδιο σύστημα που επεξεργάζονται εικόνες, να μπορούν έστω πειραματικά, ερευνητικά, δοκιμαστικά να φτιάχνουν δικά τους μοντέλα, και να εξελίσσονται περαιτέρω στο κομμάτι του “AI – Deep Learning”, έστω και σε θεωρητικό υπόβαθρο, ακολουθώντας τα βήματα από διάφορα ερευνητικά άρθρα που δημοσιεύονται ή/και γράφονται δικά τους. Βάσει αυτής της πληροφορίας, η Aria πλέον μοιράζεται τις δυνατότητες της μαζί με αυτές που προσφέρει η έκδοση LittleOne, όπου πλέον προ-επεξεργασία, εκπαίδευση, δοκιμή και αξιολόγηση εικόνων και δικτύων γίνονται ένα, φέρνοντας την Aria App LittleOne Edition, όπου τη καθιστά μια έξυπνη, minimal, εφαρμογή, με όμορφο και λιτό UI/UX, και χρήση σκοτεινού θέματος.

Στη συνέχεια του κεφαλαίου 5 και 6, θα γίνει λεπτομερής επεξήγηση όλων των αντικειμένων (components) της εφαρμογής, τόσο σε κομμάτι front-end αλλά και back-end τεχνολογιών που χρησιμοποιήθηκαν.

4.2 Τεχνολογική επισκόπηση της Aria App

Η Aria App, «χτίστηκε» στο προγραμματιστικό περιβάλλον MATLAB R2023a, τόσο σε επίπεδο front-end αλλά και σε back-end. Η εφαρμογή έρχεται σε μορφή εκτελέσιμου αρχείου (.exe, executable file) το οποίο «τρέχει» σε λειτουργικό σύστημα Windows, οποιασδήποτε έκδοσης, εφόσον υποστηρίζονται τα .NET frameworks 3.5 και 4.0.

Στο κομμάτι του front-end, έχουν χρησιμοποιηθεί όλα τα εργαλεία που είναι διαθέσιμα στο περιβάλλον MATLAB σε συνδυασμό με την χρήση νέων αντικειμένων (components, objects) που αποτελούνται από προγραμματιστικές γλώσσες – περιβάλλον web development, όπως:

- HTML/HTML5
- CSS
- JavaScript-Vue.JS
- PHP (για συγκεκριμένες λειτουργίες, που αφορούν κυρίως το back-end) μέσω της επιλογής της εισαγωγής εξατομικευμένων αντικειμένων (components, objects).

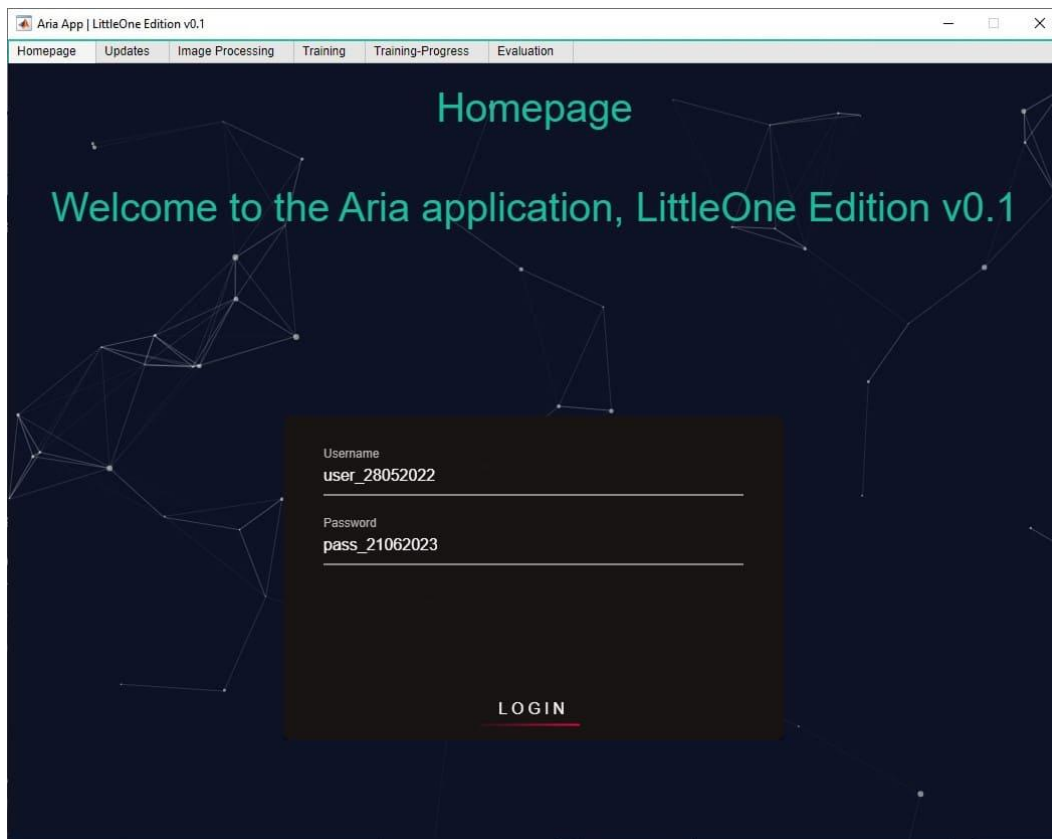
Στη πλευρά του back-end, έχει χρησιμοποιηθεί πλήρως η γλώσσα προγραμματισμού MATLAB, για όλες τις διαδικασίες, λειτουργίες, γραφικές – εικονικές παρουσιάσεις και μαθηματικές πράξεις. Στη πλειοψηφία των σημείων του κώδικα, η εκτέλεση, εφόσον αυτή είναι εφικτή, γίνεται με τη πρώτη διαθέσιμη κάρτα γραφικών για την γρηγορότερη επεξεργασία – ανάλυση των δεδομένων / πράξεων.

Στις επόμενες σελίδες του υπό-κεφαλαίου 5.2., θα γίνει μια σύντομη παρουσίαση των καρτελών της εφαρμογής:

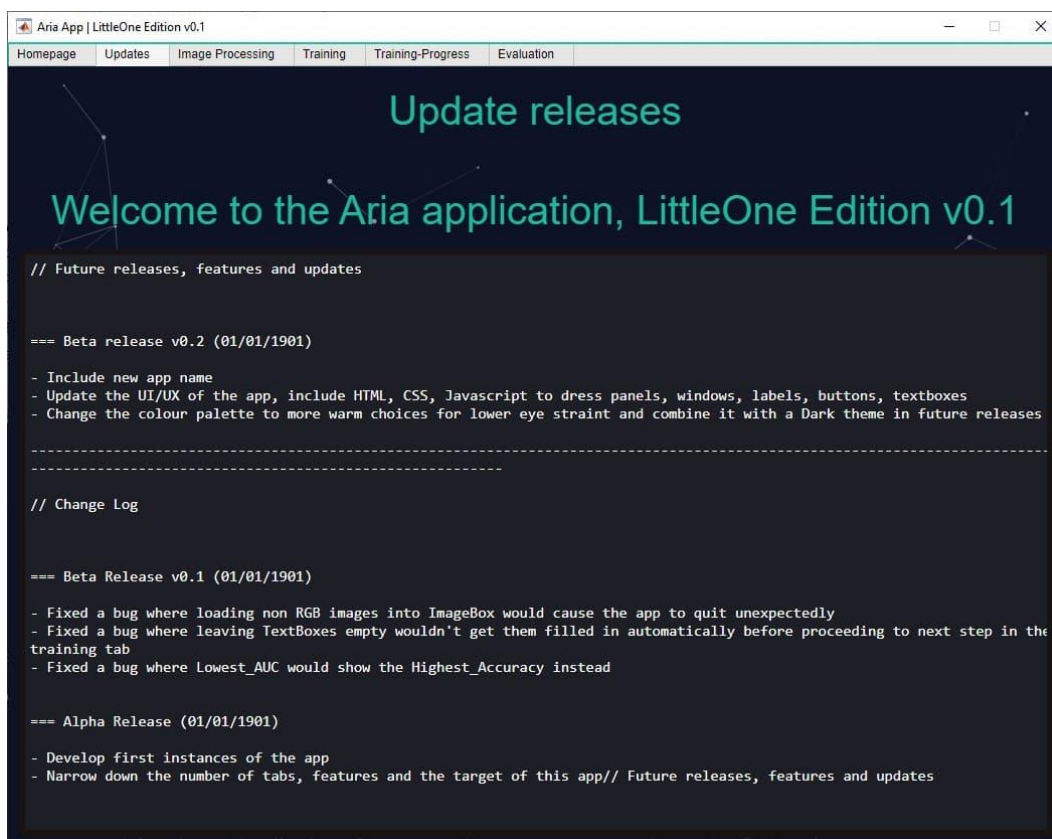
- Homepage
- Updates
- Image Processing
- Training
- Training-Progress
- Evaluation.

Ενώ στη συνέχεια, θα γίνει περιληπτική ανάλυση όλων των καρτελών σε επίπεδο καρτελών και το περιεχόμενο αυτών, τόσο στο κομμάτι του front-end αλλά και back-end, και ειδικότερα θα δοθεί έμφαση στη λειτουργία αυτών.

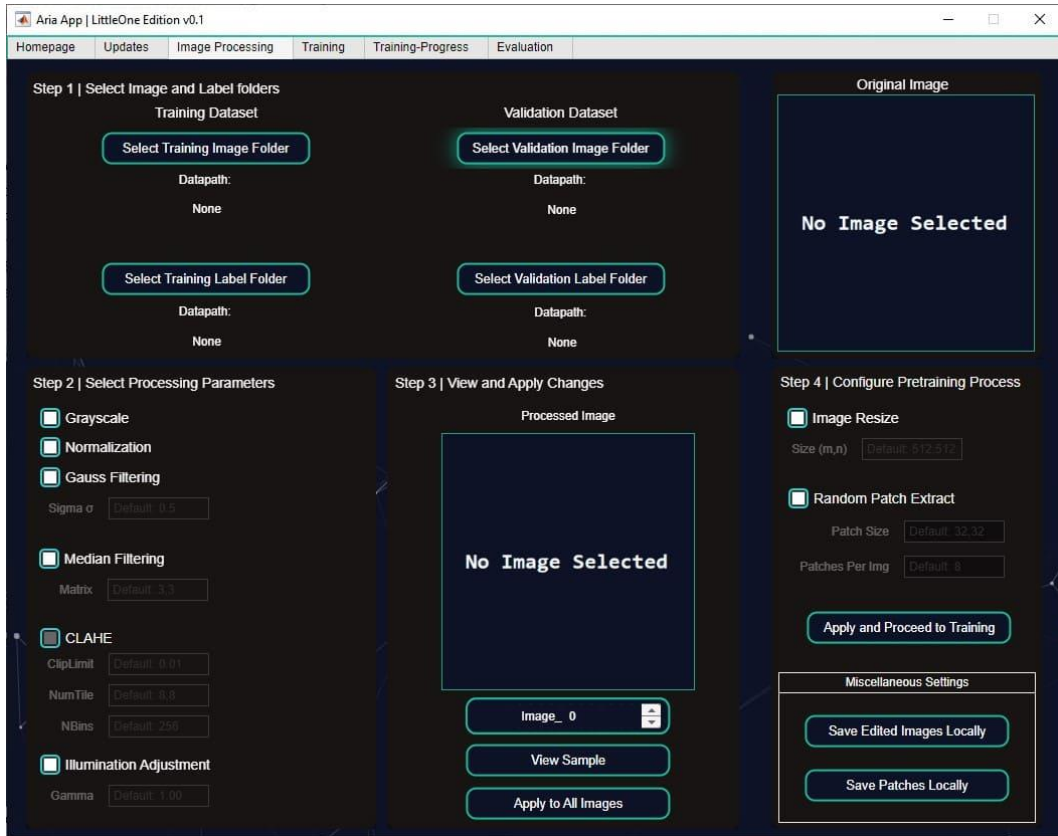
4.3 Overview: Aria App – Tabs



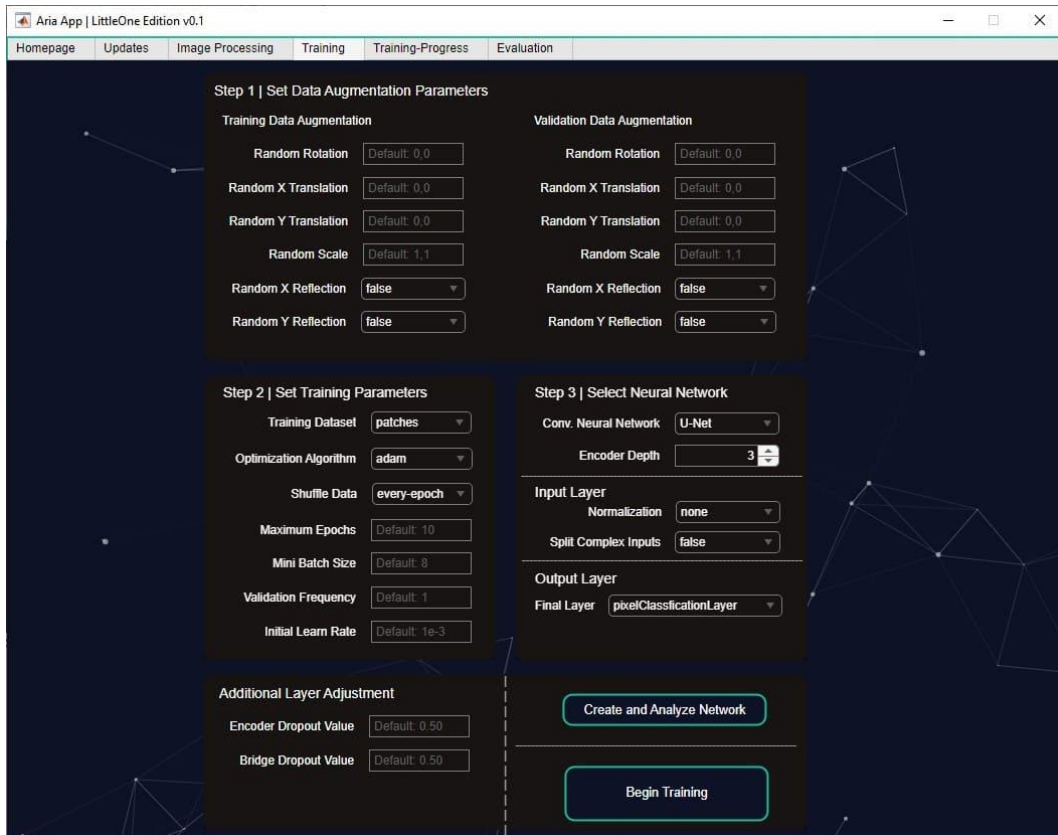
Εικόνα 18: Homepage – Αρχική καρτέλα



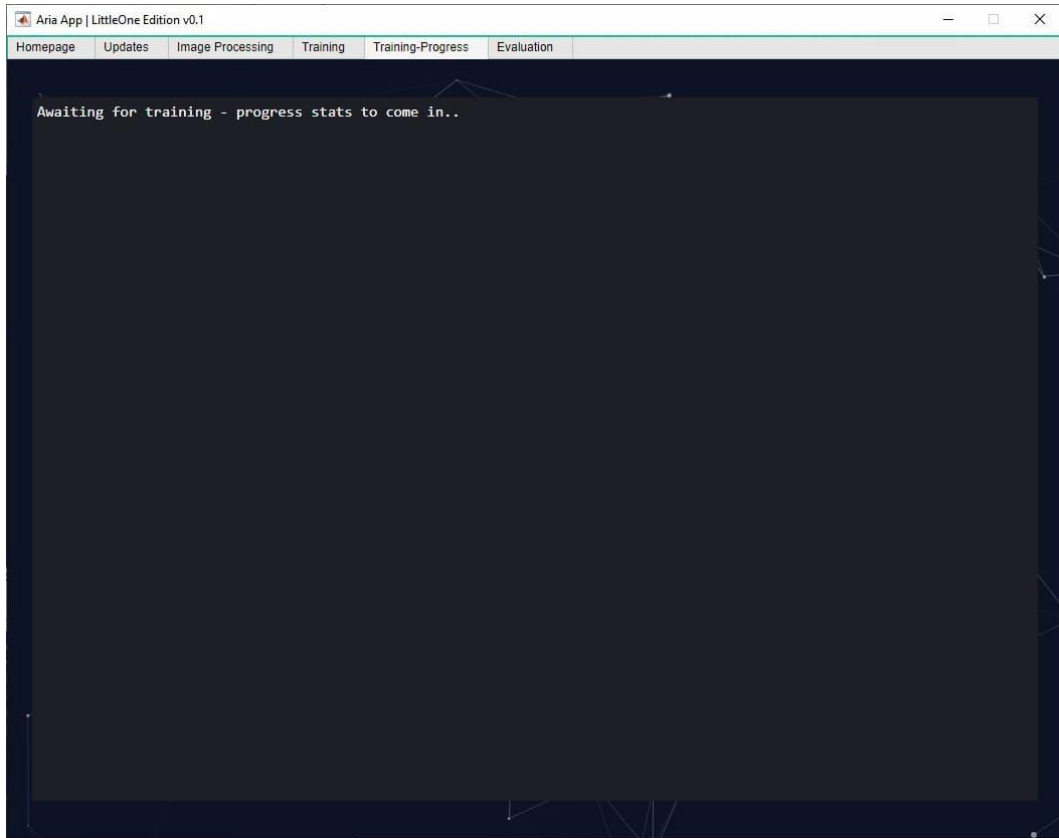
Εικόνα 19: Updates – Καρτέλα ενημερώσεων



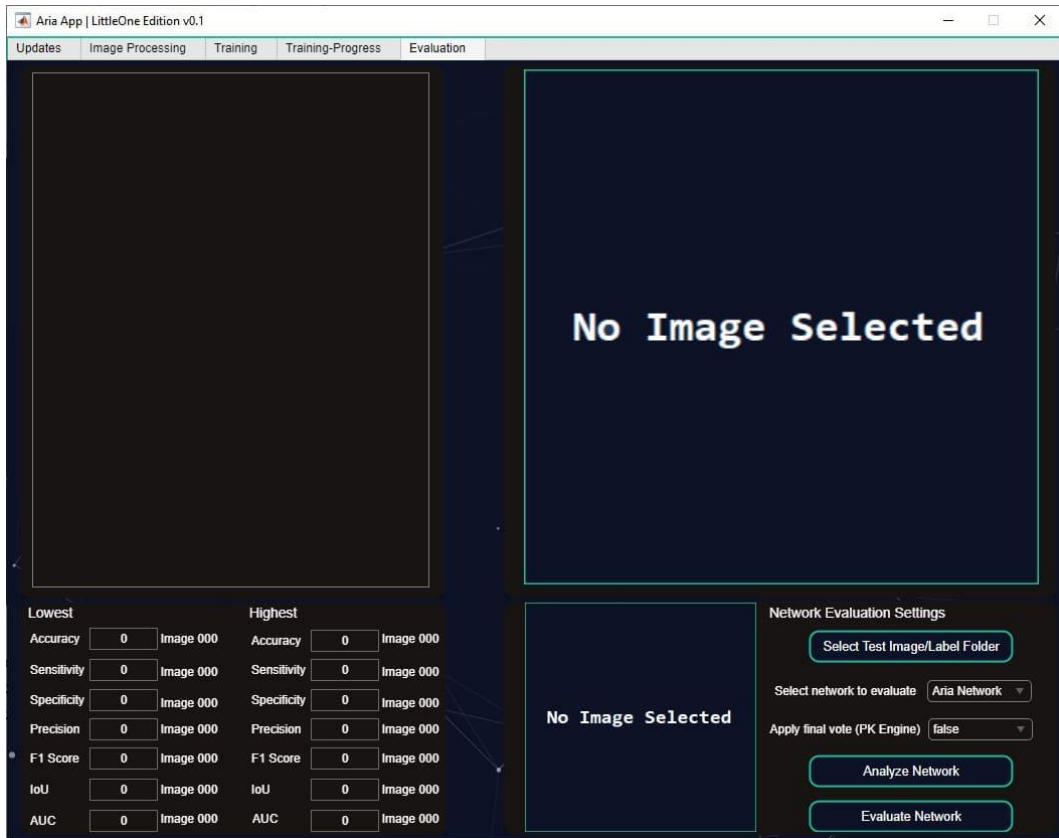
Εικόνα 20: Image Processing – Οθόνη προεπεξεργασίας



Εικόνα 21: Training – Καρτέλα εκπαίδευσης μοντέλου



Εικόνα 22: Training Progress – Καρτέλα προόδου του μοντέλου

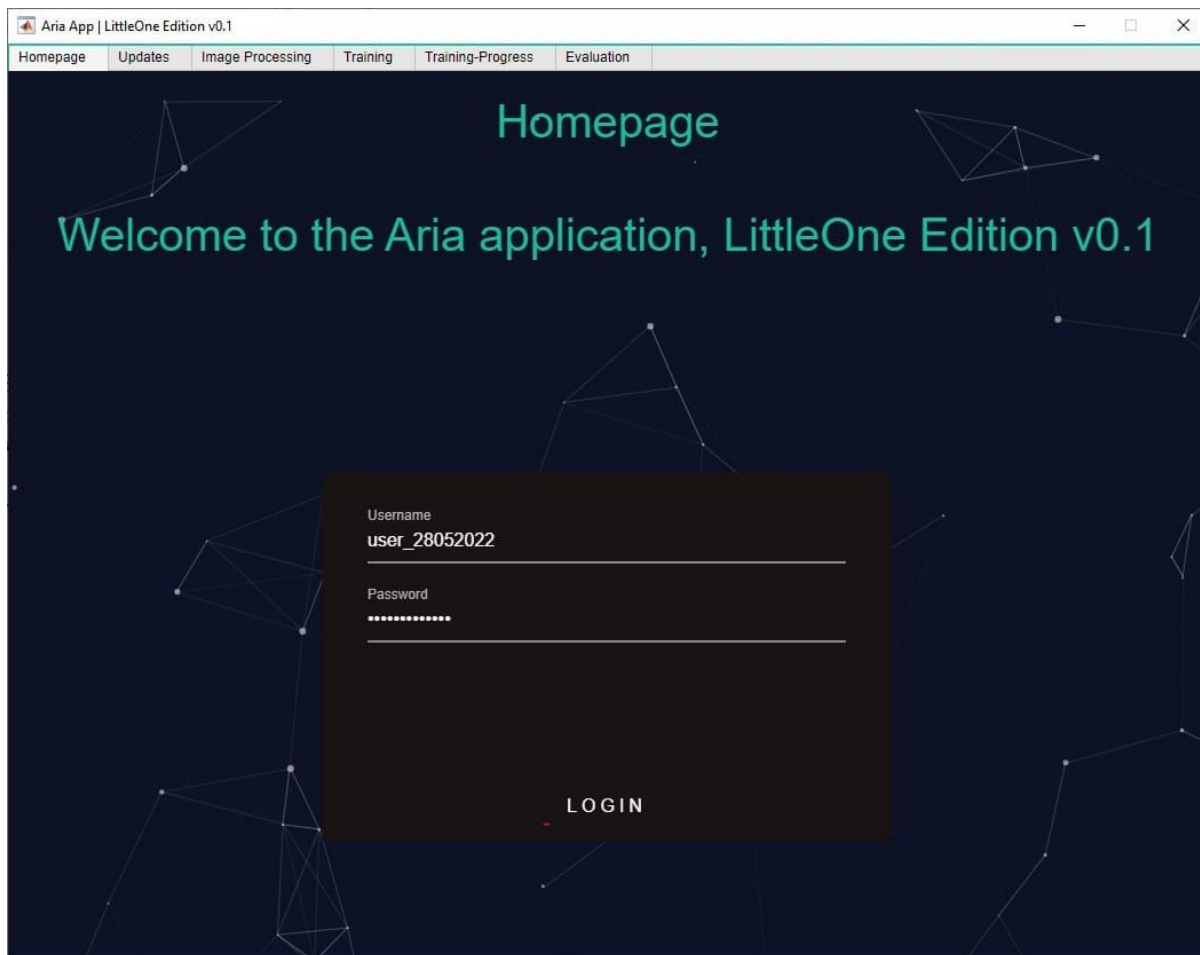


Εικόνα 23: Evaluation – Καρτέλα αξιολόγησης μοντέλου

Κεφάλαιο 5: Aria App - Frontend

Στην παρόν κεφάλαιο 5, θα επεξηγηθεί το κομμάτι του front-end της εφαρμογής, ξεκινώντας από το την αρχική καρτέλα “Homepage”.

5.1 Aria App: Frontend – Homepage Tab



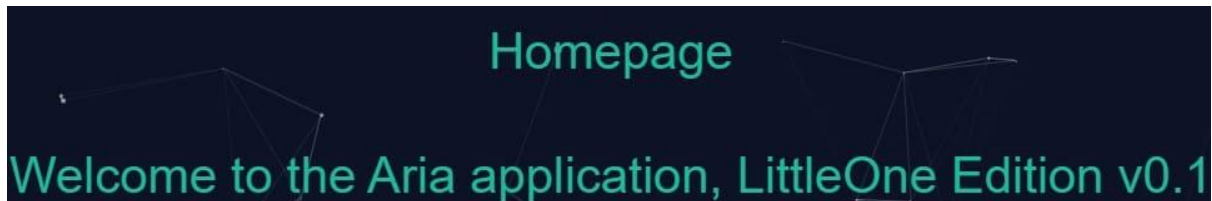
Εικόνα 24: Homepage – Αρχική καρτέλα

Η παρούσα καρτέλα εμφανίζεται κατά την εκτέλεση της εφαρμογής και έχει το ρόλο της συγκράτησης ρυθμίσεων της εφαρμογής (settings) και ταυτόχρονα της σελίδας ταυτοποίησης χρήστη, αφού διαθέτει φόρμα σύνδεσης σε λογαριασμό έτσι ώστε να μπορέσει ο χρήστης να αποκτήσει πρόσβαση στις υπόλοιπες καρτέλες της εφαρμογής, πλην της καρτέλας Updates, στην οποία δεν απαιτείται ταυτοποίηση, και μπορεί ελεύθερα ο χρήστης να έχει πρόσβαση πριν κάνει τη ταυτοποίηση του λογαριασμού του.

Η καρτέλα αποτελείται από components του AppDesigner:

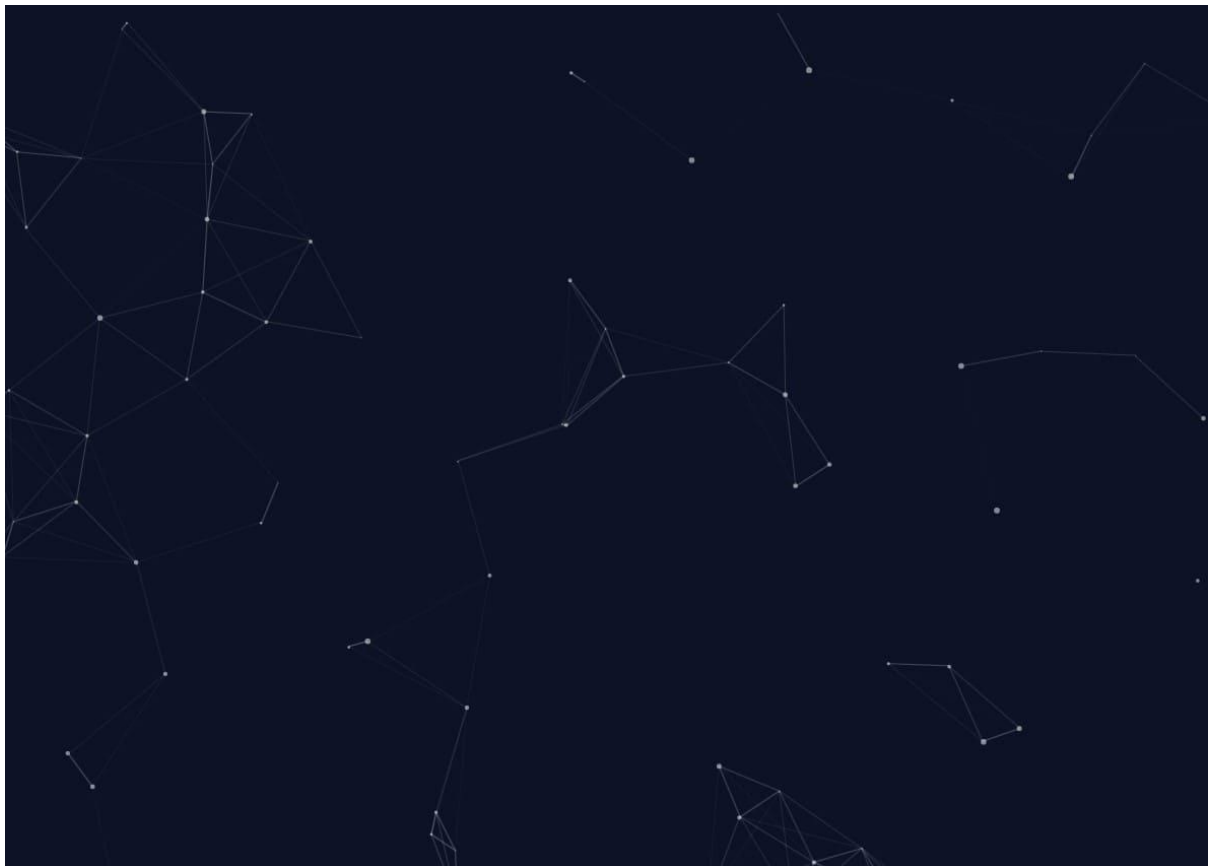
- Label_Component
- HTML_Component

Όπου το Label_Component, εμφανίζεται ως στατικός τίτλος καλωσορίσματος στην καρτέλα Homepage:



Εικόνα 25: Homepage – Τίτλος αρχικής καρτέλας

Και τα HTML_Component(s) εμφανίζονται ως το δυναμικό παρασκήνιο (background) της καρτέλας:



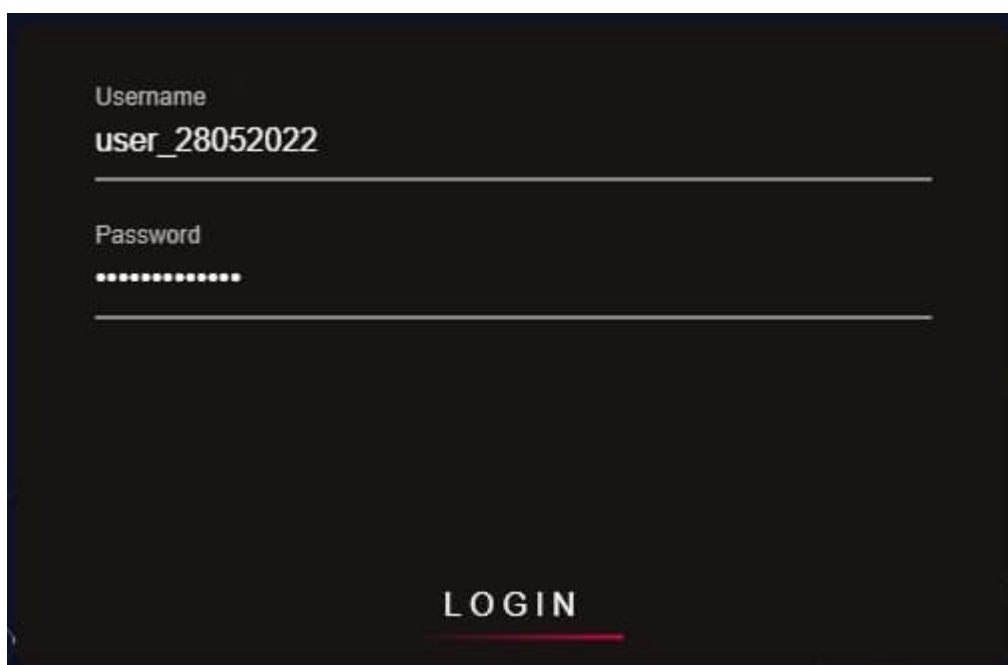
Εικόνα 26: Background (Παρασκήνιο)

Το οποίο παρασκήνιο είναι αποτέλεσμα χρήσης [...]

[...] χρήσης HTML, CSS και JavaScript και είναι βασισμένο στη βιβλιοθήκη “particles.js by VincentGarreau» η οποία είναι ανοιχτού κώδικα “open source” και μπορεί να χρησιμοποιηθεί ελεύθερα με την προϋπόθεση ότι η (ελεύθερη) άδεια χρήσης, βρίσκεται εντός της εργασίας.

Για την παρουσίαση του συγκεκριμένου θέματος / τρόπου εναλλαγής των εφέ αλλά και τη συμπεριφορά κατά τη περιφορά του κέρσορα “cursor” πάνω στην οθόνη, έχουν γίνει αλλαγές στην βάση του κώδικα της βιβλιοθήκης έτσι ώστε να επιτευχθεί μεγαλύτερη ταχύτητα ανταπόκρισης και χρήσης λιγότερης μνήμης RAM.

Σε συνέχεια του παρασκήνιου, εμφανίζεται το δεύτερο και τελευταίο HTML_Component, όπου είναι η φόρμα σύνδεσης – ταυτοποίησης λογαριασμού – χρήστη η οποία αποτελείται από HTML, CSS και JavaScript.



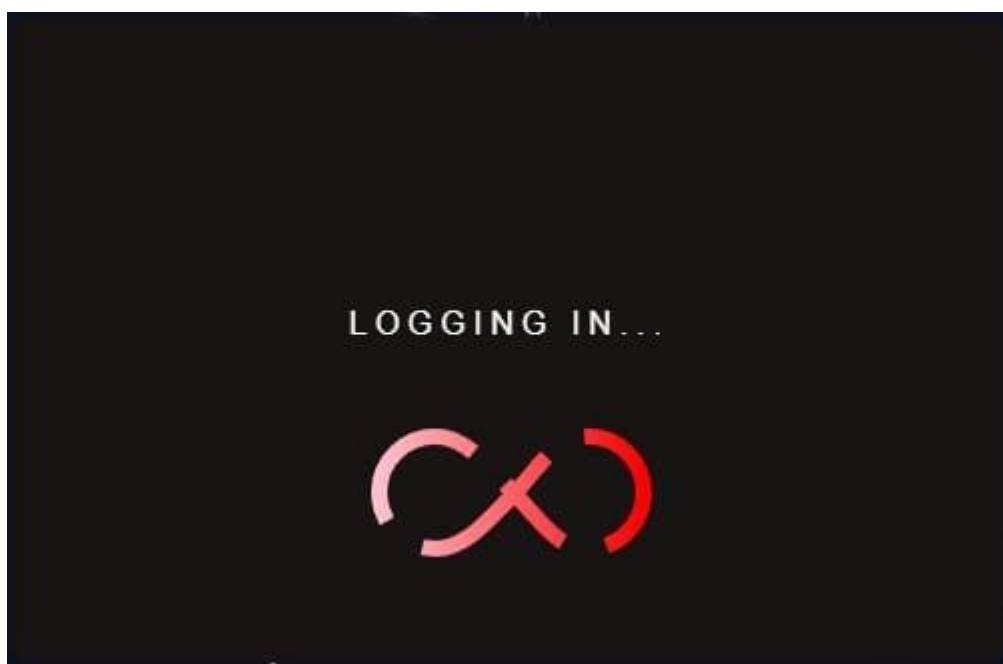
The image shows a login form on a dark background. It consists of two input fields. The first field is labeled 'Username' and contains the text 'user_28052022'. The second field is labeled 'Password' and contains a series of dots representing masked characters. Below these fields is a button labeled 'LOGIN' in white capital letters, which has a red underline.

Εικόνα 27: Homepage – Φόρμα σύνδεσης

Η οποία δέχεται ως διαπιστευτήρια ένα όνομα χρήστη και κωδικό πρόσβασης, όπως εμφανίζονται στην εικόνα XX. Το κομμάτι του CSS αναλαμβάνει την τροποποίηση του χρώματος, μεγέθους, σχήματος και ειδικών εφέ (login), ενώ η JavaScript αναλαμβάνει την παρουσίαση της εναλλαγής της κατάστασης της φόρμας μόλις ζητηθεί η είσοδος στην εφαρμογή μέσω του κουμπιού LOGIN και ταυτόχρονα παρέχει την δυνατότητα στον χρήστη να προβάλλει τον κωδικό του, κάνοντας παρατεταμένα κλικ στο πεδίο Password.

Παρακάτω εμφανίζονται:

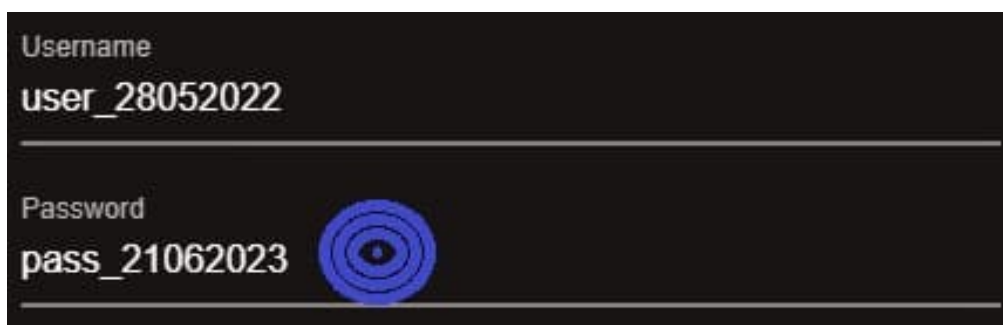
Εναλλαγή κατάστασης της φόρμας (Loading):



Εικόνα 28: Homepage – Loading screen

Η οποία περιλαμβάνει το μήνυμα “LOGGING IN...” και μια δυναμική εικόνα τύπου .gif, η οποία περιέχει το σύμβολο του απείρου να περιστρέφεται κατά τη διάρκεια της διαπίστευσης των στοιχείων που εισήχθησαν.

Προβολή κωδικού χρήστη (Password):



Εικόνα 29: Homepage – Εμφανές πεδίο του κωδικού

Όπου ο ενσωματωμένος κώδικας JavaScript ελέγχει για τις εξής ενέργειες του κέρσορα: onmouseover, onmousedown και onmousemove, όταν ο κέρσορας βρίσκεται στο πεδίο του κωδικού, και έτσι δίνει τη δυνατότητα στον χρήστη να προβάλλει τον κωδικό του κάνοντας παρατεταμένο κλικ οπουδήποτε, εφόσον βρίσκεται εντός της περιοχής που εισάγεται ο κωδικός πρόσβασης.

Μόλις το παρατεταμένο κλικ διακοπεί ή/και ο κέρσορας μεταφερθεί σε άλλο σημείο, εκτός της περιοχής του πεδίου του κωδικού, τότε αυτομάτως ο κωδικός ξαναπαίρνει τη κρυφή του μορφή:

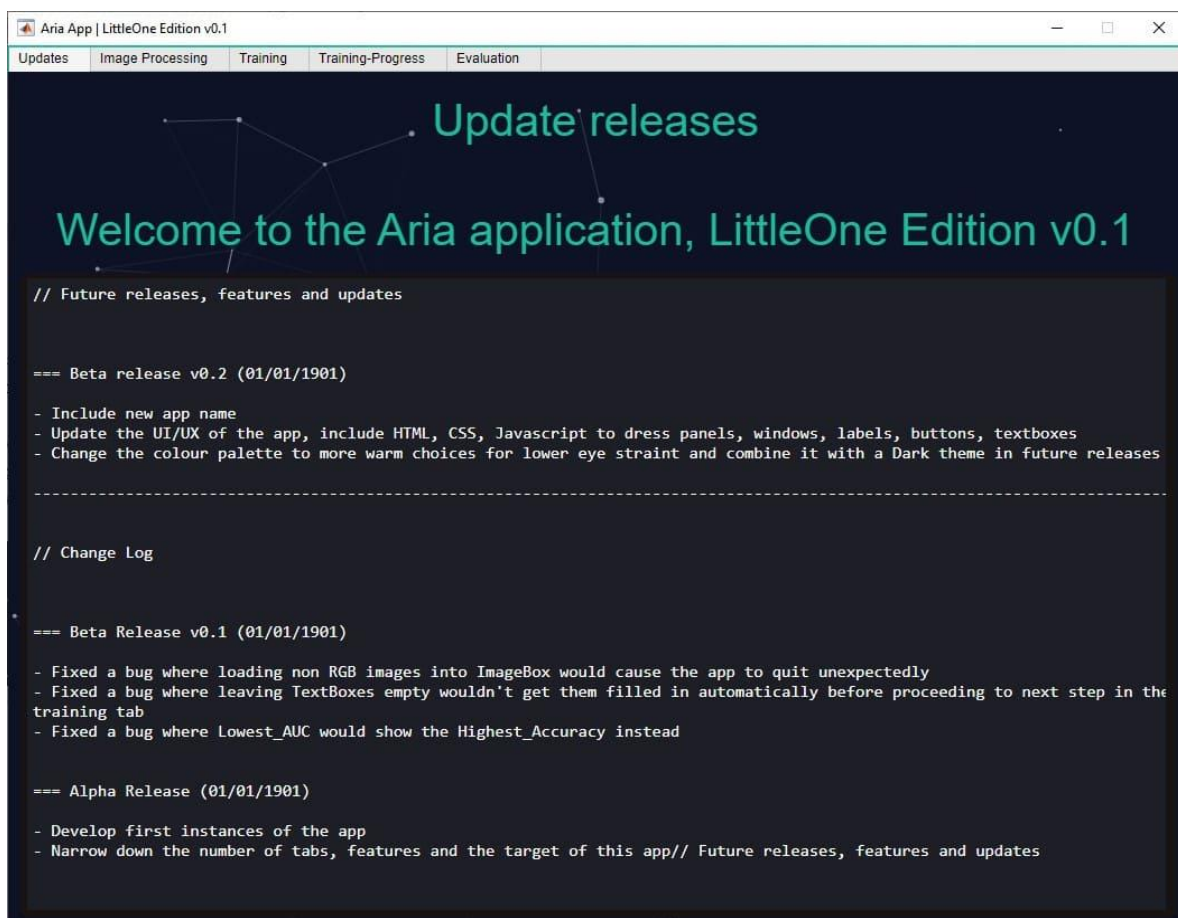


Εικόνα 30: Homepage – Μη εμφανές πεδίου του κωδικού

και η λειτουργία αυτή, αποτελεί μέρος της ευκολότερης χρήσης της εφαρμογής και ταυτόχρονα, ως ένας τρόπος ασφαλείας.

5.2 Aria App: Frontend – Updates Tab

Η καρτέλα Updates, είναι ακριβώς η επόμενη σε σειρά καρτέλα η οποία εμφανίζεται μόλις επιτευχθεί η σύνδεση στο λογαριασμό του χρήστη ή/και προτού γίνει η επαλήθευση του λογαριασμού, αφού είναι ανοιχτής πρόσβασης.

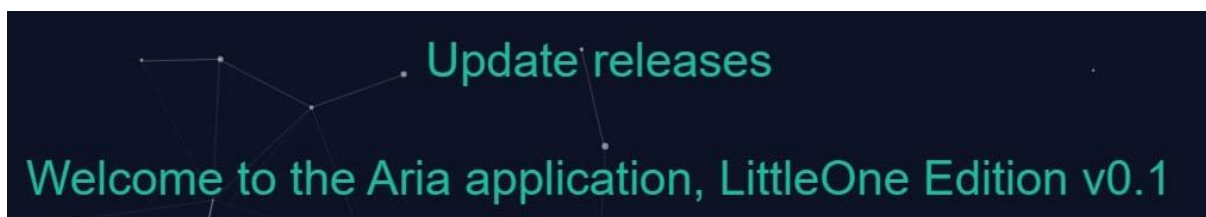


Εικόνα 31: Updates – Καρτέλα ενημερώσεων

Η καρτέλα αποτελείται από components του AppDesigner:

- Label_Component
- HTML_Component

Όπου το ένα από τα HTML_Component, κατέχει τη θέση του παρασκήνιου (background) όπως και στην προηγούμενη καρτέλα, με τις ίδιες ρυθμίσεις, αλλαγές στην βάση του κώδικα της βιβλιοθήκης. Ενώ το Label_Component στη προκειμένη περίπτωση είναι ο τίτλος καλωσορίσματος στην καρτέλα Updates, και εμφανίζεται στο ίδιο σημείο όπως στην καρτέλα Homepage, που εμφανίστηκε νωρίτερα:



Εικόνα 32: Updates – Τίτλος καρτέλας ενημερώσεων

Και το τελευταίο HTML_Component εμφανίζεται ως το στατικό πεδίο κειμένου το οποίο περιλαμβάνει όλες τις τελευταίες ενημερώσεις της εφαρμογής:

```
// Future releases, features and updates

=== Beta release v0.2 (01/01/1901)
- Include new app name
- Update the UI/UX of the app, include HTML, CSS, Javascript to dress panels, windows, labels, buttons, textboxes
- Change the colour palette to more warm choices for lower eye strain and combine it with a Dark theme in future releases

-----

// Change Log

=== Beta Release v0.1 (01/01/1901)
- Fixed a bug where loading non RGB images into ImageBox would cause the app to quit unexpectedly
- Fixed a bug where leaving TextBoxes empty wouldn't get them filled in automatically before proceeding to next step in the training tab
- Fixed a bug where Lowest_AUC would show the Highest_Accuracy instead

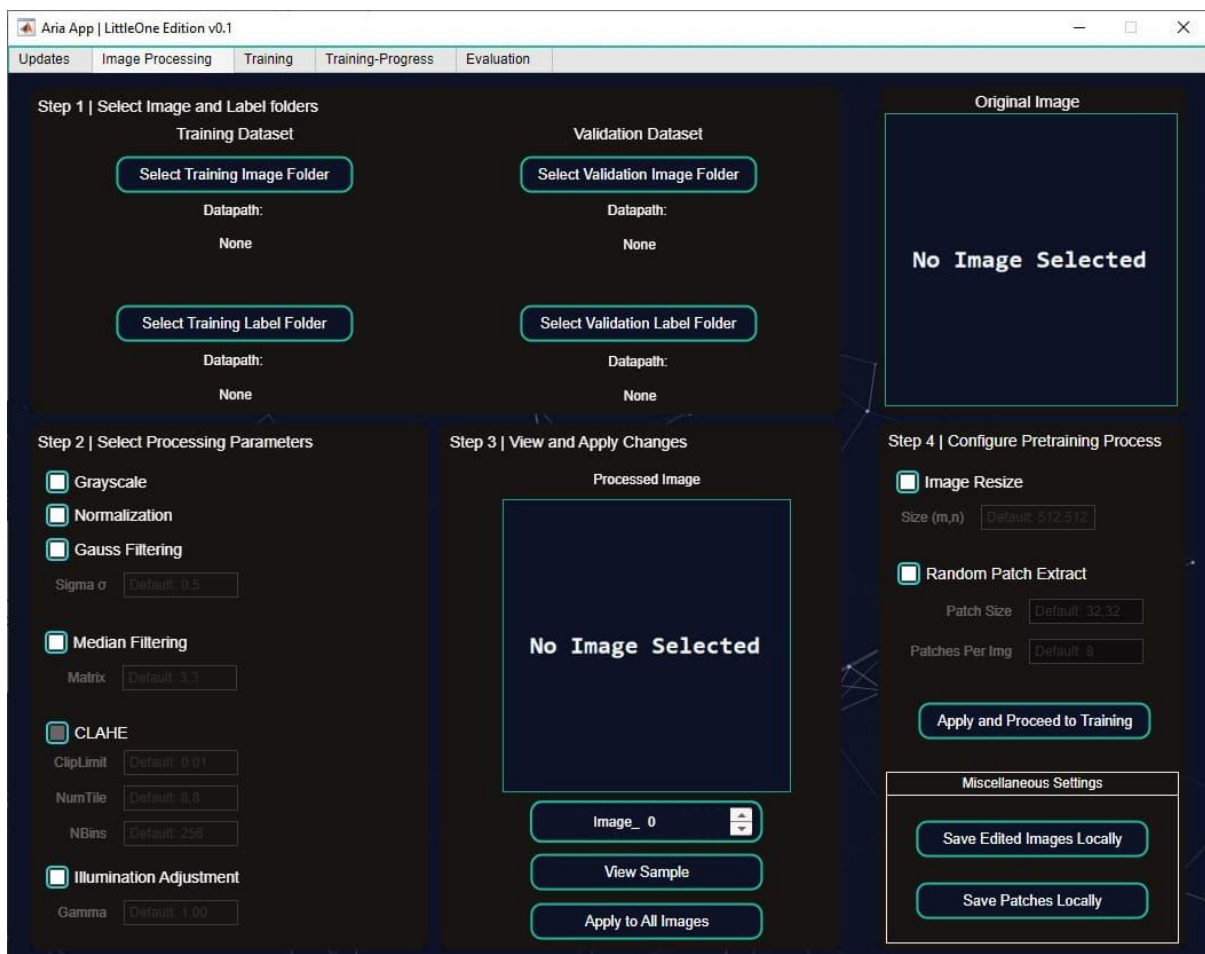
=== Alpha Release (01/01/1901)
- Develop first instances of the app
- Narrow down the number of tabs, features and the target of this app// Future releases, features and updates
```

Εικόνα 33: Updates – Πεδίο text των ενημερώσεων

Το οποίο βασίζεται σε HTML, CSS και JavaScript. Το κομμάτι του CSS, για την μορφοποίηση της φόρμας, του πεδίου και το κομμάτι της JavaScript για την τοποθέτηση του κειμένου, όποτε υπάρχει αλλαγή. Η επιλογή τοποθέτησης μιας εξατομικευμένης φόρμας αντί για αυτή που προσφέρει η MATLAB μέσω του App Designer, έγινε λόγω του απλούστερης διαδικασίας μορφοποίησης του κειμένου και των χαρακτηριστικών αυτού και παράλληλα για την καλύτερη διαχείριση του κειμένου, αφού ο διερμηνέας της JavaScript υποστηρίζει περισσότερα wildcards από ότι ένα απλό TextBox της MATLAB. Επιπροσθέτως, ένα εξατομικευμένο HTML/CSS πεδίο κειμένου, επιτρέπει την επιλογή μη εμφάνισης της μπάρας κύλισης “scrolling bar” κάτι το οποίο κάνει τη καρτέλα να είναι πιο «καθαρή».

5.3 Aria App: Frontend – Image Processing Tab

Η καρτέλα Image Processing, είναι η τρίτη σε σειρά καρτέλα η οποία εμφανίζεται και είναι προσβάσιμη, εφόσον τα διαπιστευτήρια του χρήστη έχουν το απαραίτητο επίπεδο πρόσβασης (access level) για να την προβάλλει.



Εικόνα 34: Image Processing – Καρτέλα προ-επεξεργασίας

Η καρτέλα αποτελείται από components του AppDesigner:

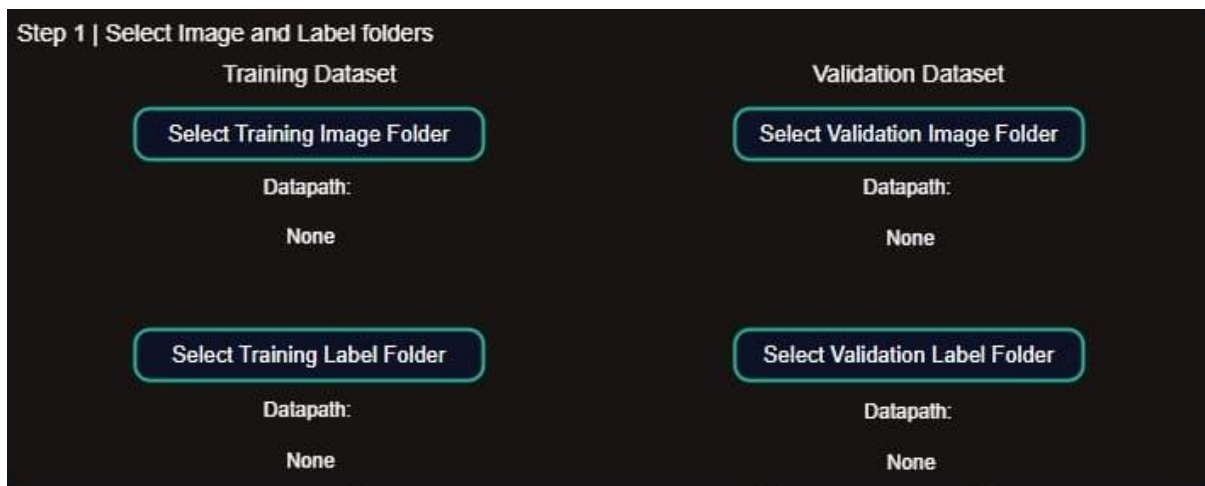
- Label_Component
- Edit_Field_Component (Numeric)
- Edit_Field_Component (Text)
- Check_Box_Component
- Image_Component
- Panel_Component
- HTML_Component

Όπου το ένα από τα HTML_Component, κατέχει τη θέση του παρασκήνιου (background) όπως και στις προηγούμενες καρτέλες, με τις ίδιες ρυθμίσεις, αλλαγές στην βάση του κώδικα της βιβλιοθήκης. Τα υπόλοιπα αντικείμενα (components) έχουν κατανεμηθεί ως εξής:

Panel Component (Πλαίσιο – Πίνακας):

Το πλαίσιο – πίνακας, είναι αντικείμενο το οποίο βρίσκεται στα διαθέσιμα αντικείμενα του AppDesigner, και χρησιμοποιείται ως ένας τρόπος διαχωρισμού οργάνωσης των βημάτων που πρέπει να ακολουθηθούν κατά τη διαδικασία της προ-επεξεργασίας των εικόνων του αμφιβληστροειδούς.

Panel 1:



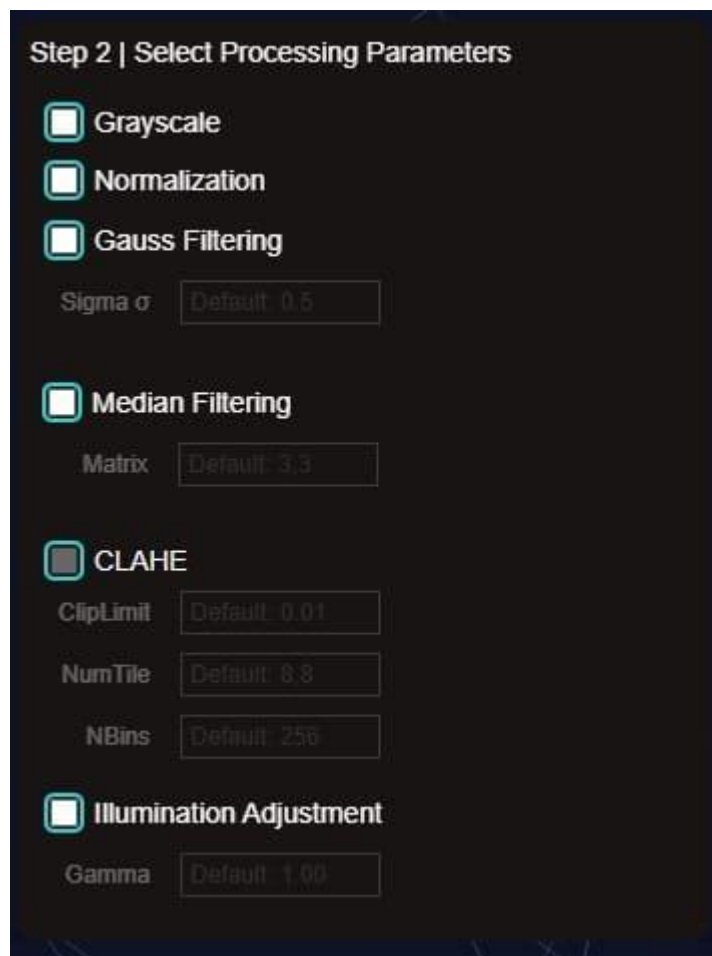
Εικόνα 35: Image Processing – Βήμα ένα της προ-επεξεργασίας

Το πρώτο πλαίσιο, το οποίο περιέχει τις επιλογές του πρώτου βήματος της διαδικασίας, τα οποία είναι:

- Label_Component
- HTML_Component

Στη προκειμένη περίπτωση, τα Label_Component, είναι τα Training, Validation Datasets και τα Datapath: None, τα οποία με τη σειρά τους αλλάζουν δυναμικά, με τη διεύθυνση των φακέλων που επιλέχθηκε, χρησιμοποιώντας τα HTML_Component, που βασίζονται σε HTML/CSS/JavaScript για την δημιουργία των κουμπιών Select Training/Validation/Training Label/Validation Label Folder.

Panel 2:



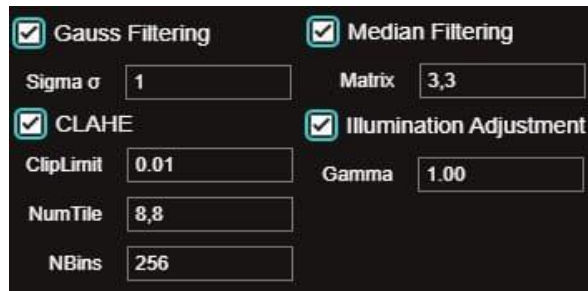
Εικόνα 36: Image Processing – Βήμα 2 της προ-επεξεργασίας

Το δεύτερο πλαίσιο, το οποίο περιέχει τις επιλογές του δεύτερου βήματος της διαδικασίας, τα οποία είναι:

- Check_Box_Component
- Edit_Field_Component (Numeric)
- Edit_Field_Component (Text)
- HTML_Component

Όπου τα Check_Box_Component: Grayscale, Normalization, Gauss Filtering, Median Filtering, CLAHE, Illumination Adjustment είναι κομμάτι των

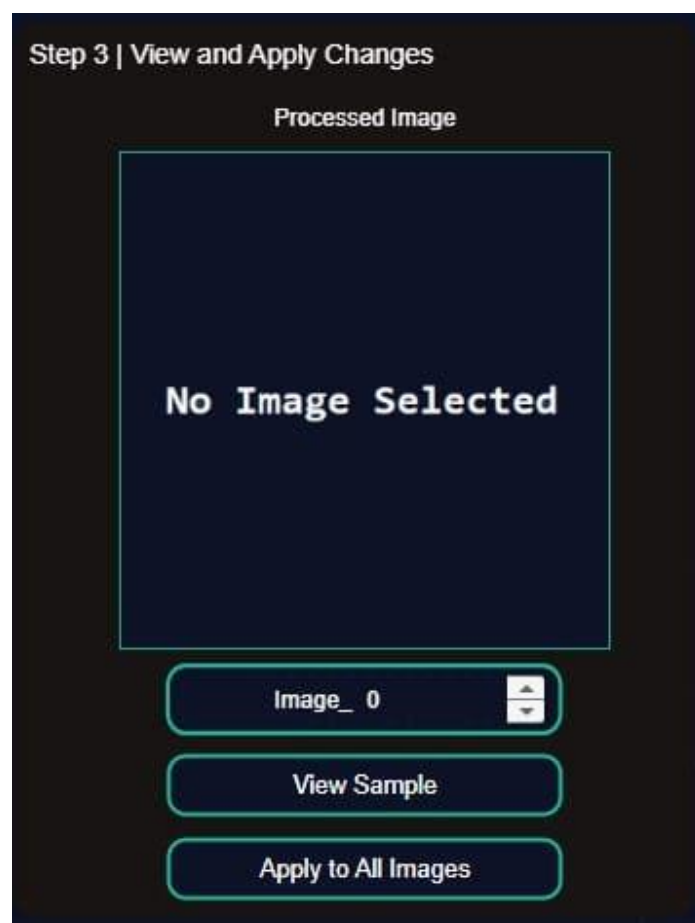
αντικειμένων που είναι διαθέσιμα προς χρήση μέσω του AppDesigner, και έχουν την δυνατότητα να επιλεχθούν (tick) και να επιτρέψουν την δυνατότητα στον χρήστη να πληκτρολογήσει, εφόσον το επιθυμεί, τα νούμερα που θέλει για κάθε επιλογή, στα αντίστοιχα Edit_Field_Component (Numeric) και Edit_Field_Component (text), με τον εξής τρόπο:



Εικόνα 37: Image Processing – Επιλογές του βήματος 2

Όπως εμφανίζονται παραπάνω. Στο συγκεκριμένο πλαίσιο – πίνακα, το HTML_Component, έχει το ρόλο του ντυσίματος των Check_Box_Component, δίνοντας το ελαφρύ μπλε εφέ, γύρω από αυτό.

Panel 3:



Εικόνα 38: Image Processing – Βήμα 3 της προ-επεξεργασίας

Στο πλαίσιο αυτό της εικόνας 38, το Label_Component 'Processed Image' δηλώνει τη λειτουργία – θέση του Image_Box_Component, όπου περιέχει κατά βάση μια εικόνα μορφής .jpg, έχει το ρόλο του place holder, και υποδηλώνει ότι δεν έχει επιλεγθεί κάποιος φάκελος με εικόνες ή/και δεν έχει γίνει ακόμη καμία επεξεργασία εικόνας, παρουσιάζοντας με αυτό το τρόπο ότι είναι κενός ο χώρος που εμφανίζονται οι επεξεργασμένες εικόνες.



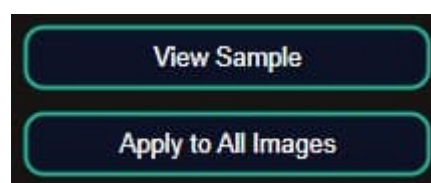
Εικόνα 39: Image Processing – Placeholder – Πεδίο εικόνας

Στη συνέχεια, βρίσκεται κάτω από το Image_Box_Component, ο μετρητής – δείκτης εικόνας, το οποίο είναι ένα Spinner_Box_Component και είναι μέρος των αντικειμένων προς χρήση από το AppDesigner, το οποίο δείχνει ποια εικόνα προβάλλεται στο Image_Box_Component.



Εικόνα 40: Image Processing – Δείκτης τωρινής εικόνας προς παρουσίαση

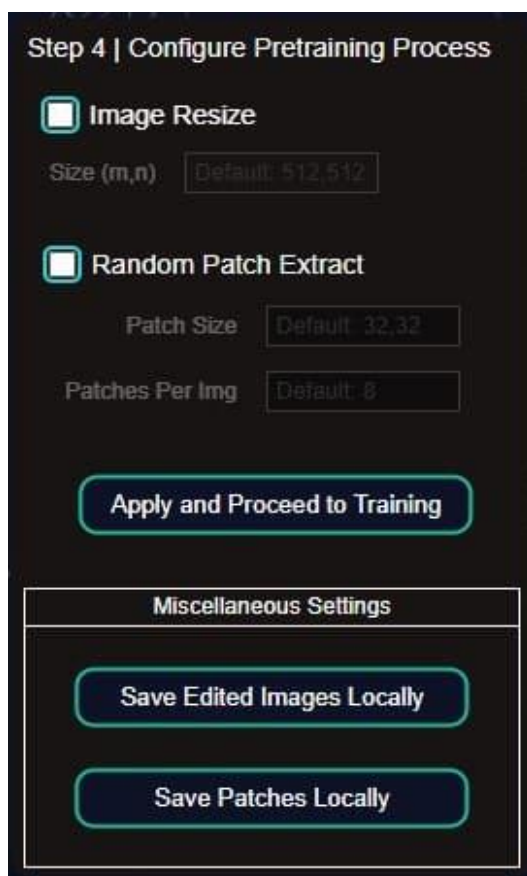
Το Spinner_Box_Component είναι ντυμένο με HTML_Component, το οποίο στην ουσία είναι ένα αρχείο HTML/CSS που δίνει την μορφή του περιγράμματος όπως φαίνεται στην εικόνα παραπάνω. Στη συνέχεια, και τέλος του πλαισίου 3 (εικόνα 41):



Εικόνα 41: Image Processing – Κουμπιά προβολής δείγματος και εφαρμογής επιλογών στο dataset

Βρίσκονται τα HTML_Component, τα οποία είναι βασισμένα σε συνδυασμό HTML/CSS/JavaScript και έχουν τη λειτουργία των κουμπιών, για τις λειτουργίες View Sample που δείχνει δείγμα επεξεργασμένης εικόνας μέσα στο πλαίσιο του Image_Box_Component και της λειτουργίας Apply to All Images, το οποίο στη πράξη, εφαρμόζει τις παραπάνω διαδικασίες επεξεργασίας που εμφανίζονται στο δείγμα (όπου δείγμα, φαίνονται ανάλογα την επιλογή – δείκτη που επιλέγει ο χρήστης).

Panel 4:



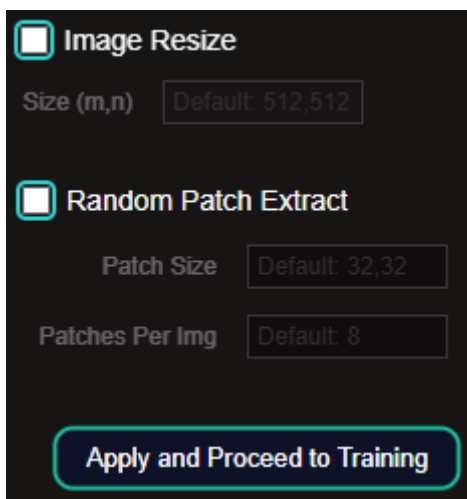
Εικόνα 42: Image Processing – Βήμα 4 της προ-επεξεργασίας

Το τρίτο πλαίσιο, το οποίο περιέχει τις επιλογές του τρίτου βήματος της διαδικασίας, τα οποία είναι:

- Check_Box_Component
- Edit_Field_Component (Numeric)
- Edit_Field_Component (Text)
- HTML_Component

Όπου τα Check_Box_Component είναι για τις λειτουργίες Image Resize και Random Patch Extract, τα οποία εφόσον επιλεγθούν, δίνουν τη δυνατότητα

στον χρήστη να εισάγει τις τιμές που θέλει, στα αντίστοιχα Edit_Field_Component (Numeric και Text). Στη συνέχεια, ακολουθούν τα κουμπιά που βασίζονται σε HTML/CSS/JavaScript, και έχουν τη λειτουργία των Apply and Proceed to Training, όπου στη πράξη, εφαρμόζει τις δύο προηγούμενες επιλογές των Check_Box_Component, και μεταφέρει τη λειτουργία, μαζί με όλες τις προηγούμενες επιλογές Check_Box_Component, από όλα τα πλαίσια 2 και 3, σε συνδυασμό με το 4, και ύστερα μεταφέρεται στην καρτέλα Training.



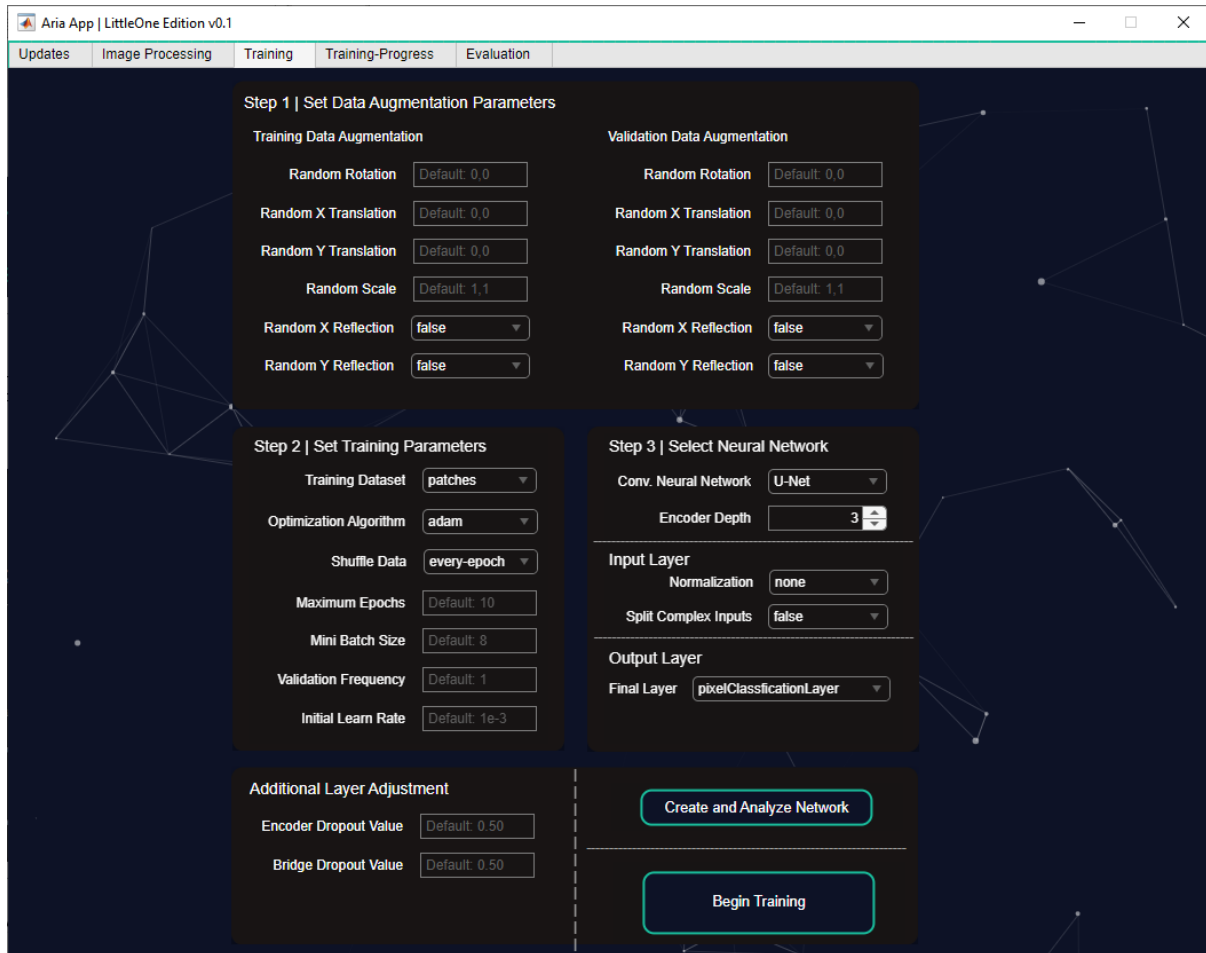
Εικόνα 43: Image Processing – Τελευταία επιλογή προ-επεξεργασίας πριν την εκπαίδευση

Επιπροσθέτως, παρέχεται και το υπό-πλαίσιο “Miscellaneous Settings” όπου αποτελείται από 2 HTML_Component, που βασίζονται σε HTML/CSS/JavaScript και έχουν τη λειτουργία Save Edited Images και Patches, Locally. Που δίνει τη δυνατότητα στον χρήστη να αποθηκεύσει τις εικόνες τοπικά, στον υπολογιστή του, εφόσον το επιθυμεί.

Το τελευταίο αντικείμενο στη καρτέλα Image Processing, είναι το αντικείμενο Image_Box_Component, σε συνδυασμό με το Label_Component, που υποδηλώνει τη λειτουργία Original Image, και αντικατοπτρίζει το χώρο του Image_Box_Component, στο οποίο προβάλλεται η εικόνα χωρίς καμία επεξεργασία, αφού επιλεγθεί φάκελος που περιέχει εικόνες του αμφιβληστροειδή. Αλλιώς, αν δεν επιλεγθεί κάποιος φάκελος, ο χώρος αυτός περιέχει την εικόνα place holder – “No Image Selected”, τύπου .jpg, που δηλώνει ότι ο χώρος είναι κενός.

5.4 Aria App: Frontend – Training Tab

Η καρτέλα Training, είναι η τέταρτη σε σειρά καρτέλα η οποία εμφανίζεται και είναι προσβάσιμη, εφόσον τα διαπιστευτήρια του χρήστη έχουν το απαραίτητο επίπεδο πρόσβασης (access level) για να την προβάλλει. Στην εικόνα 44, παρουσιάζεται η καρτέλα αυτή.



Εικόνα 44: Training – Καρτέλα παραμετροποίησης δικτύου προς εκπαίδευση

Η καρτέλα αποτελείται από components του AppDesigner:

- Label_Component
- Drop_Down_Box_Component
- Edit_Field_Component (Numeric)
- Edit_Field_Component (Text)
- Check_Box_Component
- Image_Component
- Panel_Component
- HTML_Component

Όπου το ένα από τα HTML_Component, κατέχει τη θέση του παρασκήνιου (background) όπως και στις προηγούμενες καρτέλες, με τις ίδιες ρυθμίσεις, αλλαγές στην βάση του κώδικα της βιβλιοθήκης. Τα υπόλοιπα αντικείμενα (components) έχουν καταναμεμηθεί ως εξής:

Panel Component (Πλαίσιο – Πίνακας):

Το πλαίσιο – πίνακας, είναι αντικείμενο το οποίο βρίσκεται στα διαθέσιμα αντικείμενα του AppDesigner, και χρησιμοποιείται ως ένας τρόπος διαχωρισμού οργάνωσης των βημάτων που πρέπει να ακολουθηθούν κατά τη διαδικασία της προ-εκπαίδευσης του μοντέλου, θέτοντας τις απαραίτητες παραμέτρους.

Panel 1:

Step 1 Set Data Augmentation Parameters	
Training Data Augmentation	Validation Data Augmentation
Random Rotation	Random Rotation
Random X Translation	Random X Translation
Random Y Translation	Random Y Translation
Random Scale	Random Scale
Random X Reflection	Random X Reflection
Random Y Reflection	Random Y Reflection

Εικόνα 45: Training – Βήμα 1 της καρτέλας εκπαίδευσης

Το πρώτο πλαίσιο, το οποίο περιέχει τις επιλογές του πρώτου βήματος της διαδικασίας και τα αντίστοιχα components (αντικείμενα) είναι:

- Label_Component
- Edit_Field_Component (Numeric)
- Edit_Field_Component (Text)
- Drop_Down_Box_Component
- Panel_Component

Στο οποίο τα Label_Component, Training / Validation Data Augmentation δηλώνουν τη κατηγορία στην οποία εφαρμόζονται οι επιλογές – λειτουργίες που βρίσκονται από κάτω τους.

Στο συγκεκριμένο πλαίσιο, περιέχονται τα `Edit_Field_Component` (Numeric) και `Edit_Field_Component` (Text) τα οποία έχουν αναλυθεί εκτενώς στις προηγούμενες υπό-ενότητες και δίνουν την επιλογή στον χρήστη να θέσει τις δικές του τιμές, για τις λειτουργίες:

- Random Rotation
- Random X Translation
- Random Y Translation
- Random Scale

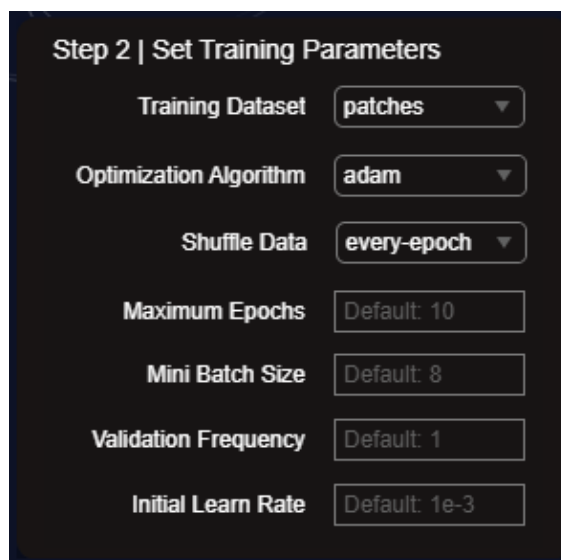
Που εφαρμόζονται ξεχωριστά για της κατηγορίες δεδομένων (εικόνων) Training / Validation Data.

Με τη σειρά τους, παρουσιάζονται προς το τέλος δύο νέα components, τα οποία είναι τα `Drop_Down_Box_Component`, τα οποία δίνουν τη δυνατότητα να επιλέξει ο χρήστης αν θέλει να ενεργοποιήσει (true) ή να απενεργοποιήσει (false) τις λειτουργίες:

- Random X Reflection
- Random Y Reflection

Που εφαρμόζονται και αυτά, ξεχωριστά για της κατηγορίες δεδομένων (εικόνων) Training / Validation Data.

Panel 2:



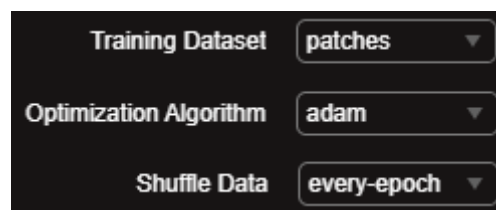
Step 2 Set Training Parameters	
Training Dataset	patches
Optimization Algorithm	adam
Shuffle Data	every-epoch
Maximum Epochs	Default: 10
Mini Batch Size	Default: 8
Validation Frequency	Default: 1
Initial Learn Rate	Default: 1e-3

Εικόνα 46: Training – Βήμα 2 της καρτέλας εκπαίδευσης

Το πρώτο πλαίσιο, το οποίο περιέχει τις επιλογές του πρώτου βήματος της διαδικασίας και τα αντίστοιχα components (αντικείμενα) είναι:

- Drop_Down_Box_Component
- Edit_Field_Component (Numeric)
- Edit_Field_Component (Text)

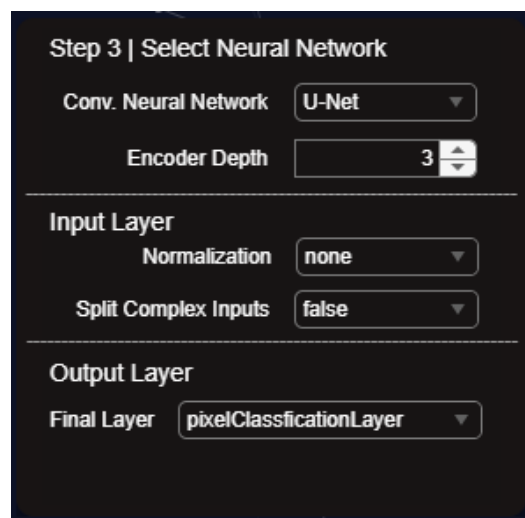
Και το Drop_Down_Box_Component εμφανίζεται στις πρώτες επιλογές, δίνοντας την επιλογή στον χρήστη να θέσει το είδος της λειτουργίας για το Training Dataset, Optimization Algorithm και Shuffle Data, ως μέρος της παραμετροποίησης της εκπαίδευσης του μοντέλου.



Εικόνα 47: Training – Επιλογές του βήματος 2 για την παραμετροποίηση της εκπαίδευσης

Στη συνέχεια, παρουσιάζονται τα Edit_Field_Component (Numeric) και Edit_Field_Component (Text) για τις επιλογές – λειτουργίες Maximum Epochs, Mini Batch Size, Validation Frequency, Initial Learn Rate, τα οποία επίσης μέρος της παραμετροποίησης της εκπαίδευσης του μοντέλου και τα components έχουν την ίδια ακριβώς συμπεριφορά όπως έχει αναλυθεί εκτενώς στις προηγούμενες υπό-ενότητες.

Panel 3:

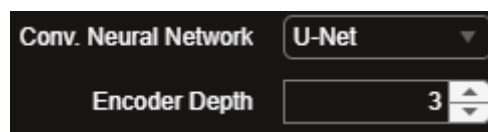


Εικόνα 48: Training – Βήμα 3 της καρτέλας εκπαίδευσης

Το τρίτο και τελευταίο πλαίσιο των βασικών βημάτων – παραμέτρων, το οποίο περιέχει τις επιλογές του τρίτου βήματος της διαδικασίας και τα αντίστοιχα components (αντικείμενα) είναι:

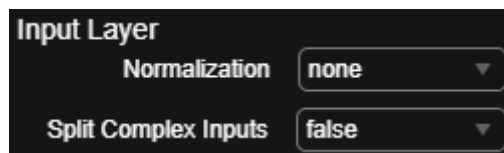
- Label_Component
- Drop_Down_Box_Component
- Spinner_Box_Component

Όπου τα Label_Component Input και Output Layer δηλώνουν το σημείο του δικτύου στο οποίο εφαρμόζονται οι λειτουργίες – επιλογές των Drop_Down_Box_Component και Spinner_Box_Component:



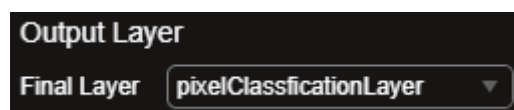
Εικόνα 49: Training – Επιλογή δικτύου και βάθος του encoder side

Ξεκινώντας από το δίκτυο (στη περίπτωση της παρούσας εργασίας, είναι το U-Net) με τη χρήση Drop_Down_Box_Component και το βάθος (αριθμός σταδίων encoder – decoder layers) μέσω ενός Spinner_Box_Component που δίνει τη δυνατότητα να αυξομειώνει τον αριθμό των σταδίων.



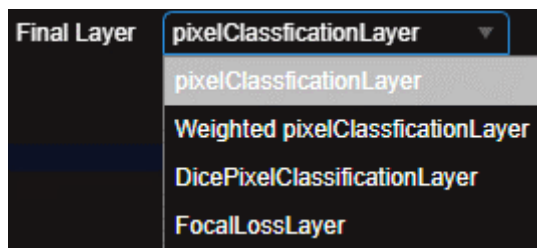
Εικόνα 50: Training – Επιλογές – παράμετροι του ImageInputLayer

Αμέσως μετά βρίσκεται το Label_Component για την υποκατηγορία Input Layer, που αντικατοπτρίζει τις παραμέτρους που μπορούν να τροποποιηθούν κατά τη θέληση του χρήστη, για τις επιλογές Normalization (none, zero center, zscore, rescale-symmetric και rescale-zero-one) και Split Complex Inputs (true / false), μέσω των Drop_Down_Box_Component.



Εικόνα 51: Training – Στάδιο επιλογής final output layer

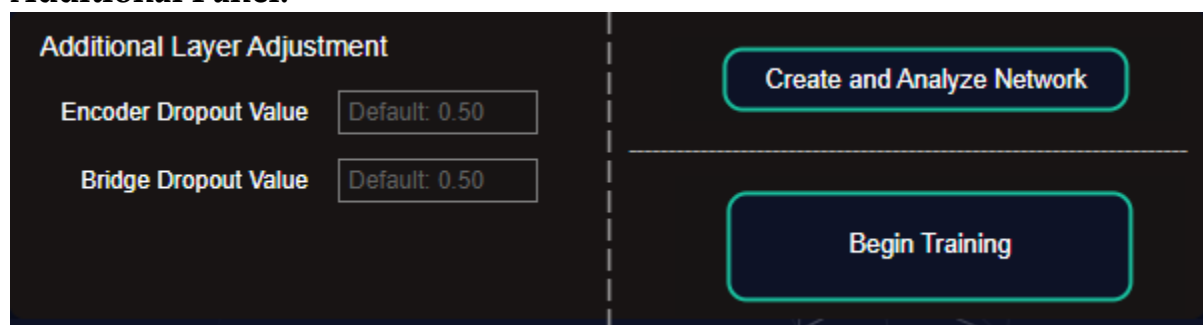
Και τέλος του πλαισίου 3, βρίσκεται το Label_Component για την υποκατηγορία Output Layer, που δίνει τη δυνατότητα στον χρήστη μέσω ενός Drop_Down_Box_Component, τη δυνατότητα να επιλέξει ποιο θα είναι το τελευταίο επίπεδο εξόδου στο δίκτυο του:



Εικόνα 52: Training – Λίστα επιλογών final output layers

Βάσει του αποτελέσματος / εργασίας και αποτελέσματος – στόχου που θέλει να επιτευχθεί μέσω αυτού του μοντέλου.

Additional Panel:

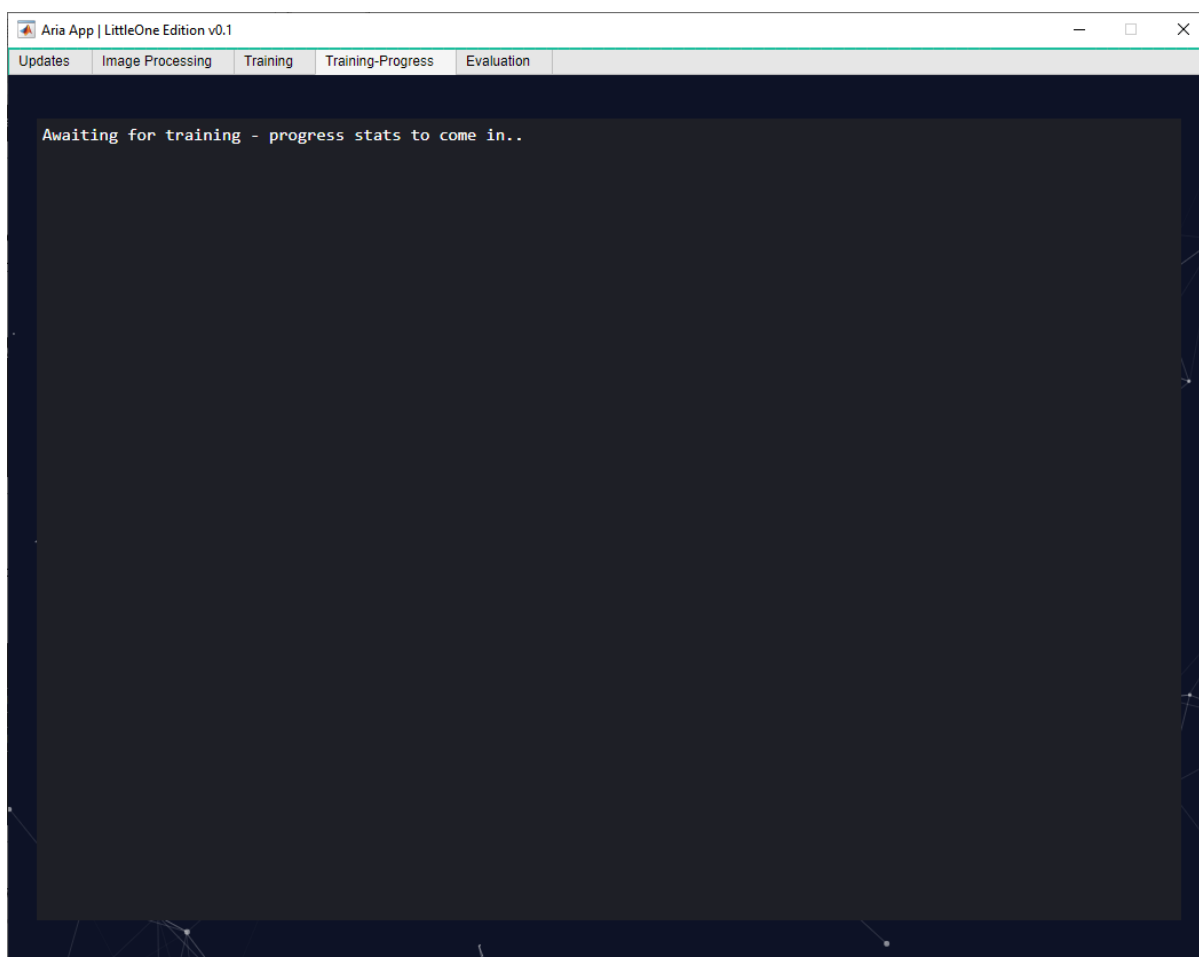


Εικόνα 53: Training – Επιπλέον πλαίσιο παραμετροποίησης Dropout layer και δημιουργίας δικτύου

Το τελευταίο πλαίσιο, το οποίο δεν ανήκει στην σειρά των προηγούμενων πλαισίων, είναι χωρισμένο σε μια προαιρετική επιλογή επιπρόσθετης παραμετροποίησης των επιπέδων – στρωμάτων (layers) Dropout, μέσω Edit_Field_Component (Numeric) στα αριστερά, και δίνει τη δυνατότητα ο χρήστης να θέσει δικές του τιμές και στα δεξιά, βρίσκονται δύο HTML_Component, βασισμένα σε HTML/CSS/JavaScript τα οποία είναι το Create and Analyze Network, που δημιουργεί το δίκτυο βάσει των προηγούμενων παραμέτρων και το Begin Training το οποίο ξεκινάει την εκπαίδευση του μοντέλου, αφού δημιουργηθεί και αναλυθεί.

5.5 Aria App: Frontend – Training Progress Tab

Η καρτέλα Training Progress, είναι η Πέμπτη και προτελευταία σε σειρά καρτέλα η οποία εμφανίζεται και είναι προσβάσιμη, εφόσον τα διαπιστευτήρια του χρήστη έχουν το απαραίτητο επίπεδο πρόσβασης (access level) για να την προβάλλει. Στην εικόνα 54 προβάλλεται η καρτέλα αυτή.



Εικόνα 54: Training Progress – Πεδίο προβολής της προόδου εκπαίδευσης του δικτύου

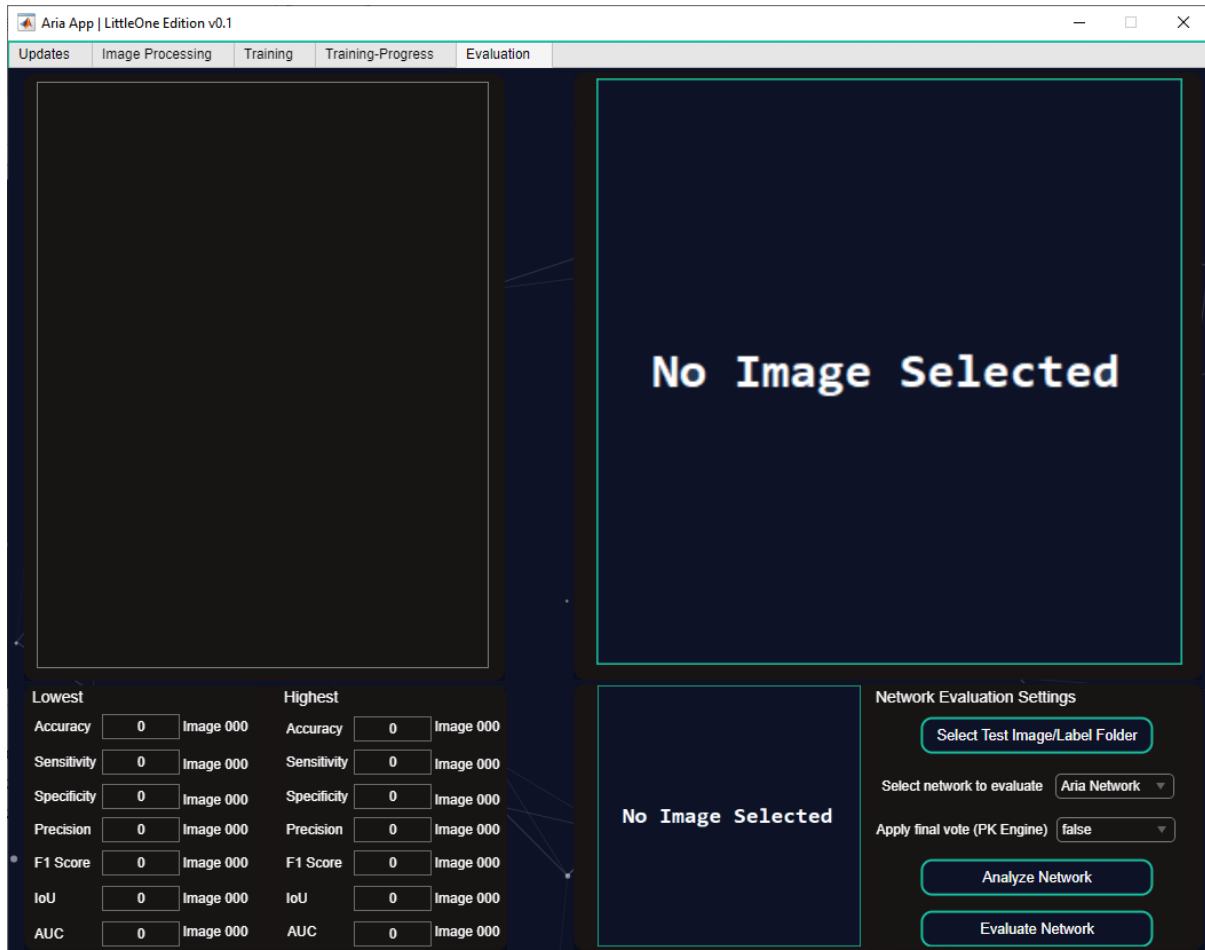
Η καρτέλα αποτελείται από components του AppDesigner:

- HTML_Component

Όπου το ένα από τα HTML_Component, κατέχει τη θέση του παρασκήνιου (background) όπως και στις προηγούμενες καρτέλες, με τις ίδιες ρυθμίσεις, αλλαγές στην βάση του κώδικα της βιβλιοθήκης. Το δεύτερο και τελευταίο σε σειρά HTML_Component, είναι αυτό που λειτουργεί ως TextBox και είναι βασισμένο σε HTML/CSS/JavaScript το οποίο προσφέρει τις ίδιες δυνατότητες και ικανότητες, όπως αναλύθηκε στην υπό-ενότητα 4.

5.6 Aria App: Frontend – Evaluation Tab

Η καρτέλα Evaluation, είναι η έκτη και τελευταία σε σειρά καρτέλα η οποία εμφανίζεται και είναι προσβάσιμη, εφόσον τα διαπιστευτήρια του χρήστη έχουν το απαραίτητο επίπεδο πρόσβασης (access level) για να την προβάλλει.



Εικόνα 55: Evaluation – Καρτέλα αξιολόγησης δικτύων Aria Network, et al

Η καρτέλα αποτελείται από components του AppDesigner:

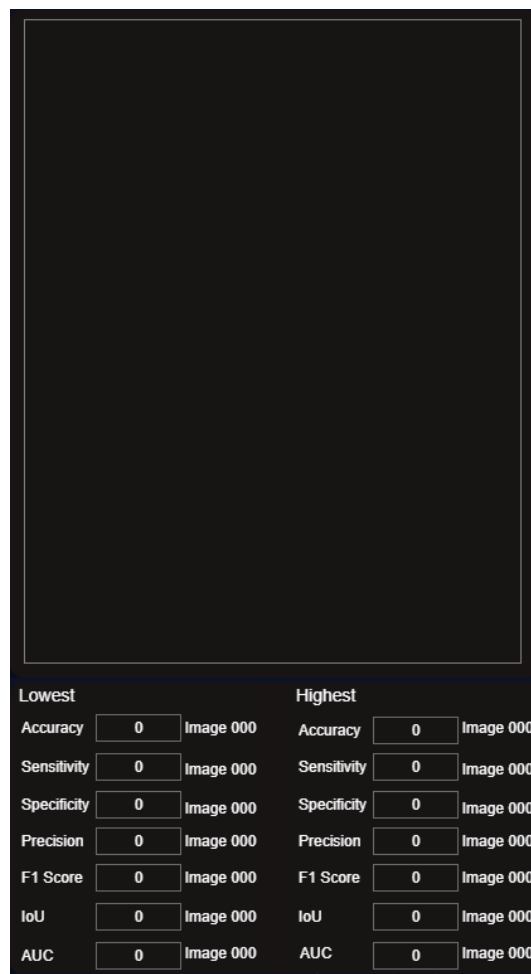
- Text_Box_Component
- Image_Box_Component
- Drop_Down_Box_Component
- HTML_Component

Όπου το ένα από τα HTML_Component, κατέχει τη θέση του παρασκήνιου (background) όπως και στις προηγούμενες καρτέλες, με τις ίδιες ρυθμίσεις, αλλαγές στην βάση του κώδικα της βιβλιοθήκης.

Ξεκινώντας από το αριστερό κομμάτι της καρτέλας, υπάρχει το Text_Box_Component, το οποίο είναι ένα από τα διαθέσιμα αντικείμενα προς χρήση από το AppDesigner, και κατέχει το ρόλο της προβολής των αποτελεσμάτων του δικτύου που επιλέγεται για αξιολόγηση.

Το Text_Box_Component, συνδυάζεται με τις επιλογές που βρίσκονται από κάτω, τα οποία χωρίζονται σε κατηγορίες Lowest και Highest, για τις μετρικές αξιολογήσεις – επιλογές:

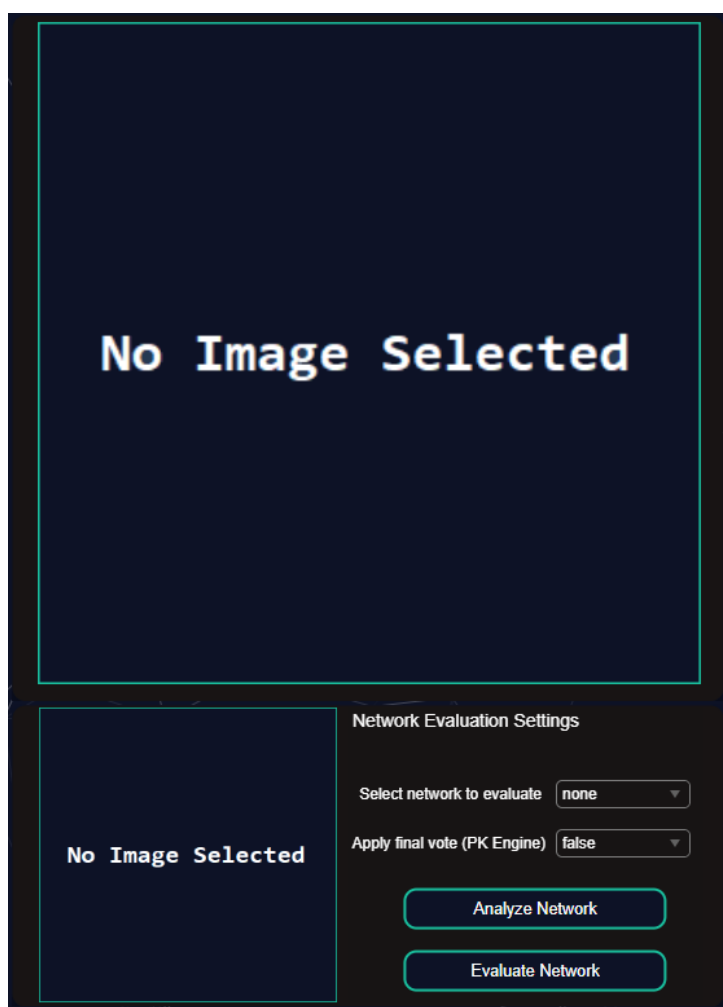
- Accuracy
- Sensitivity
- Specificity
- Precision
- F1 Score
- IoU (Intersection over Union)
- AUC (Area Under the Curve)



Εικόνα 56: Evaluation – Πεδίο προβολής της αξιολόγησης του δικτύου για εικόνες ατομικά και συλλογικά

Τα οποία παρουσιάζονται ξεχωριστά τόσο στις κατηγορίες Lowest / Highest αλλά και στο Text_Box_Component, στο οποίο παρουσιάζονται τα αποτελέσματα ξεχωριστά σε βαθμό εικόνας, βάσει του αριθμού των εικόνων που περιέχονται στον φάκελο (στην παρούσα εργασία, DRIVE dataset, που περιέχει στο σύνολο 20 εικόνες για αξιολόγηση).

Και τελικώς, φθάνοντας στο τελευταίο πλαίσιο, στη δεξιά πλευρά της καρτέλας όπως φαίνεται στην εικόνα 57 παρακάτω, βρίσκονται τα Image_Box_Component όπου περιέχουν κατά βάση μια εικόνα μορφής .jpg, που έχει το ρόλο του place holder, και υποδηλώνει ότι δεν έχει αξιολογηθεί ακόμη εικόνα για να προβάλει τα αποτελέσματα του δικτύου στο κομμάτι της κατάτμησης των αγγείων.



Εικόνα 57: Evaluation – Πεδία προβολής τελικού αποτελέσματος κατάτμησης και επιπλέον ρυθμίσεις

Στο μεγαλύτερο σε μέγεθος Image_Box_Component, προβάλλεται η εικόνα που αξιολογήθηκε σε συνδυασμό με τη μάσκα που δημιουργήθηκε από το δίκτυο, με μπλε χρώμα. Στο μικρότερο σε μέγεθος Image_Box_Component προβάλλεται η μάσκα απευθείας, σε μαύρο και άσπρο (pixels σε 0 ή 1).

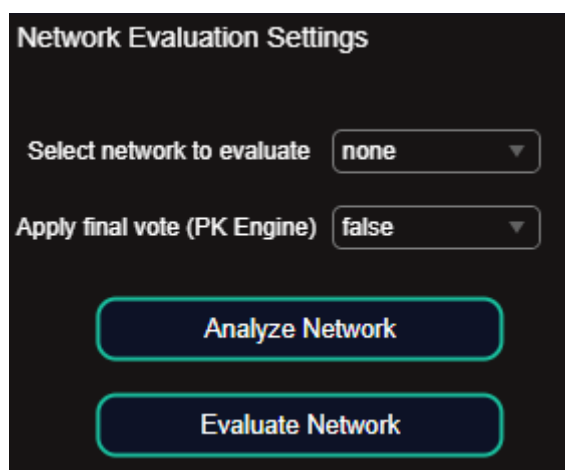
Στη συνέχεια των παραπάνω, βρίσκονται το HTML_Component που βασίζονται σε HTML/CSS/JavaScript για τη δημιουργία κουμπιών:

- Analyze Network
- Evaluate Network

Με την ίδια συμπεριφορά όπως αναλύθηκε στις προηγούμενες ενότητες που περιείχαν εξατομικευμένα κουμπιά.

Το Label_Component, το οποίο υποδηλώνει τη διαθεσιμότητα των επιλογών – ρυθμίσεων προς:

- Την επιλογή δικτύου, η οποία θα είναι “Aria Network” αν υπάρχει δίκτυο που εκπαιδεύτηκε μέσω της εφαρμογής ή “browse..” και γίνεται μέσω Drop_Down_Box_Component, αν ο χρήστης θέλει να επιλέξει άλλο δίκτυο σε μορφή .mat
- Εφαρμογής ψήφου πλειοψηφίας (Μηχανισμός PK) με την επιλογή να είναι μεταξύ false ή true.



Εικόνα 58: Evaluation – Επιλογή δικτύου προς αξιολόγηση και επιλογή ενεργοποίησης μηχανισμού PK

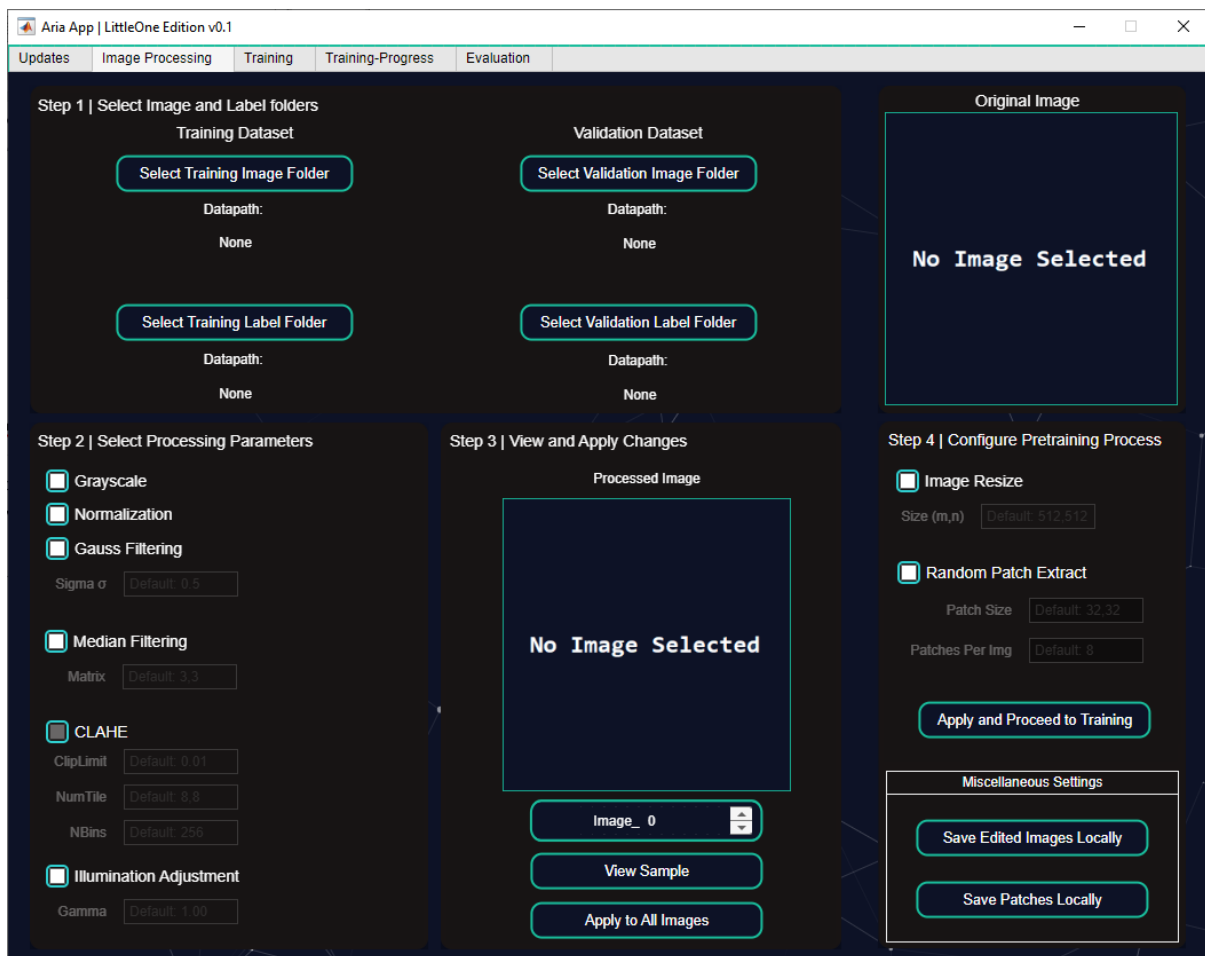
και έτσι φτάνει στο τέλος η ενότητα αυτή, στην οποία έγινε πλήρης περιγραφή όλων των αντικειμένων που εμφανίζονται στην εφαρμογή και απαρτίζουν το αισθητικό και σχεδιαστικό κομμάτι της.

Στην επομένη σελίδα, η ενότητα 5.7 (σελίδα 83) θα αναφερθούν επιπλέον λειτουργίες οι οποίες εμφανίζονται στις καρτέλες Image Processing και Evaluation.

5.7 Aria App: Frontend – Επιπλέον δυνατότητες

Σε όλες τις παραπάνω καρτέλες που αναλύθηκαν και συγκεκριμένα στις καρτέλες που περιέχουν Image_Box_Component, έχει υλοποιηθεί η λειτουργία εναλλαγής των εικόνων από αριστερά προς δεξιά και δεξιά προς αριστερά για την προβολή προηγούμενης ή επόμενης εικόνας αλλά και την μεγέθυνση των εικόνων αυτών που προβάλλονται στο Image_Box_Component.

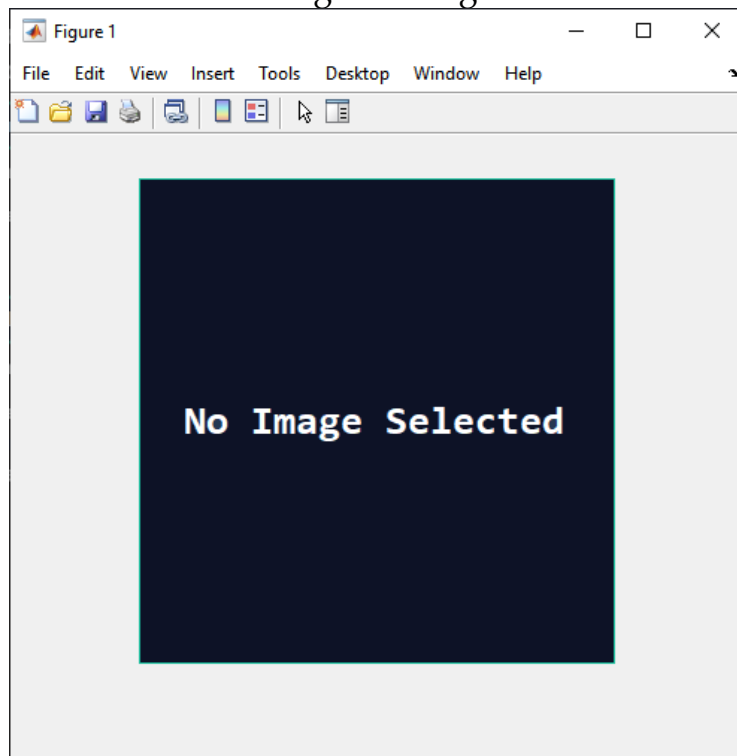
Image Processing Tab:



Εικόνα 59: Image Processing – Καρτέλα προ-επεξεργασίας εικόνων

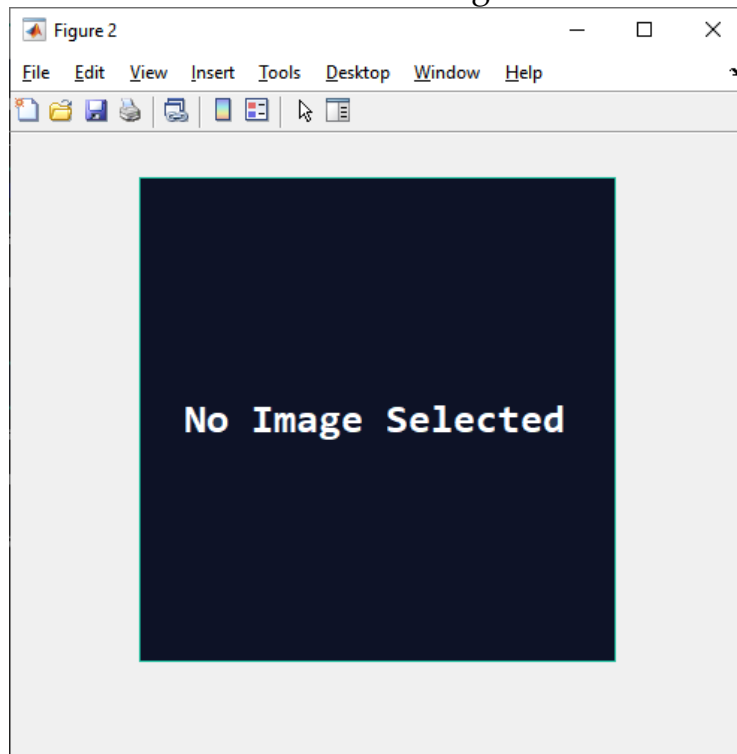
Κάνοντας κλικ με το κέρσορα στο χώρο (στη προκειμένη περίπτωση, στην εικόνα No Image Selected) των 2 Image_Box_Component, εκτελείται μια διαδικασία μεγέθυνσης της εικόνας σε ξεχωριστό παράθυρο (εικόνες 60 και 61):

Original Image



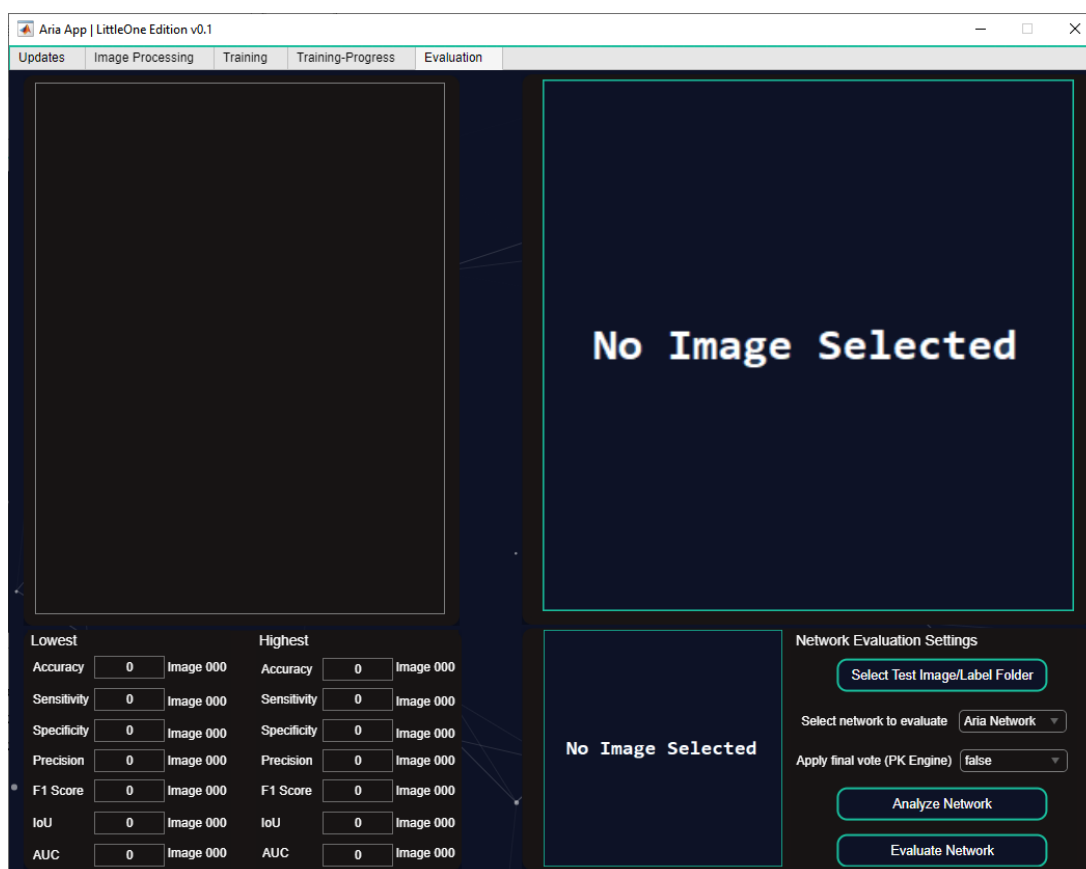
Εικόνα 60: Image Processing – Ξεχωριστό παράθυρο παρουσίασης Original Image

Processed Image



Εικόνα 61: Image Processing – Ξεχωριστό παράθυρο παρουσίασης Processed Image

Evaluation Tab:



Εικόνα 62: Evaluation – Καρτέλα αξιολόγησης δικτύου Aria Network, et al

Κάνοντας κλικ με το κέρσορα (στη προκειμένη περίπτωση, στην εικόνα No Image Selected) των 2 Image_Box_Component, εκτελείται μια διαδικασία ξεχωριστή για κάθε μία:

Μεγαλύτερου μεγέθους Image Box Component:

Παρέχει τη δυνατότητα μεταξύ εναλλαγής των εικόνων που έχουν αξιολογηθεί από το δίκτυο, κάνοντας κλικ αριστερά ή δεξιά εντός του χώρου, αυξομειώνοντας έτσι τον δείκτη προβολής εικόνας ενώ παράλληλα γίνεται και η αντίστοιχη εναλλαγή στο σημείο του μικρότερου σε μέγεθος Image_Box_Component, για να είναι αντίστοιχη η μάσκα με την εικόνα που προβάλλεται στο μεγαλύτερο σε μέγεθος Image_Box_Component.

Μικρότερου μεγέθους Image Box Component:

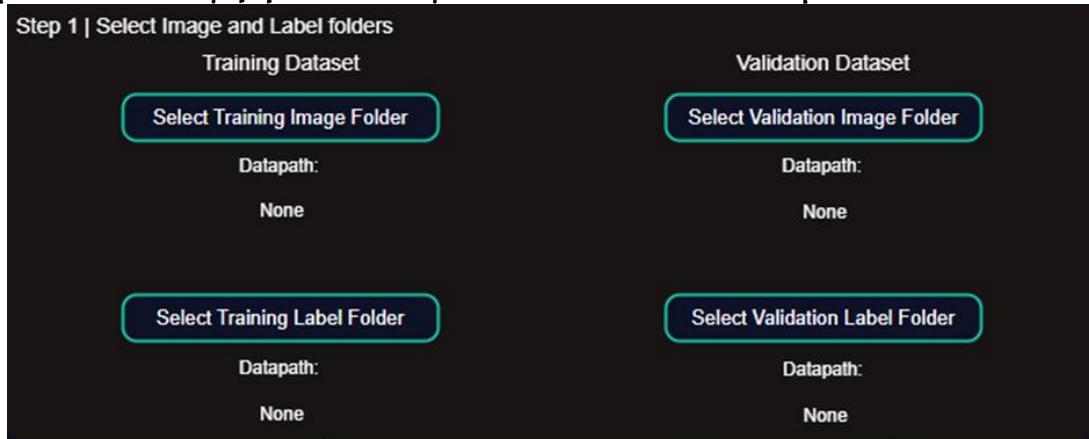
Παρέχει τη δυνατότητα της μεγέθυνσης της εικόνας σε ξεχωριστό παράθυρο, έχοντας την ίδια συμπεριφορά με αυτή που αναλύθηκε στη προηγούμενη καρτέλα Image Processing Tab.

Κεφάλαιο 6: Aria App - Backend

Στην παρούσα υπό-ενότητα, θα επεξηγηθεί το κομμάτι του back-end της εφαρμογής, ξεκινώντας από το την αρχική καρτέλα Image Processing.

6.1 Aria App: Backend – Επεξήγηση του backend των βημάτων

Βήμα 1 | Επιλογή φακέλων για τις εικόνες και τις μάσκες



Εικόνα 63: Image Processing – Βήμα 1 της προ-επεξεργασίας, κουμπιά επιλογής φακέλων

Στο βήμα αυτό, όπως παρουσιάστηκε στην προηγούμενη υπό-ενότητα του front-end, παρουσιάζονται 4 κουμπιά τα οποία αντιστοιχούν σε 2 φακέλους για τις εικόνες (φωτογραφίες) του αμφιβληστροειδή και 2 φακέλους για τις αντίστοιχες μάσκες των εικόνων (φωτογραφιών αυτών), χωρισμένοι στη training και validation dataset.

Η επιλογή των φακέλων γίνεται με τη χρήση της συνάρτησης `uigetdir()` [41] η οποία ανοίγει το παράθυρο επιλογής φακέλου των Windows. Στη περίπτωση της επιτυχής επιλογής φακέλου ή της ακύρωσης της επιλογής, έχει τοποθετηθεί μια δικλείδα ασφαλείας, η οποία κοιτάει αν υπάρχει δηλωμένη διεύθυνση – περιεχόμενο εντός της μεταβλητής ή όχι.

Παράδειγμα:

```
selectedPath = uigetdir();
```

```
if ~isempty(selectedPath)
```

```
% Success
```

```
% Insert code that updates DataPath Label with the folder location
```

```
else
```

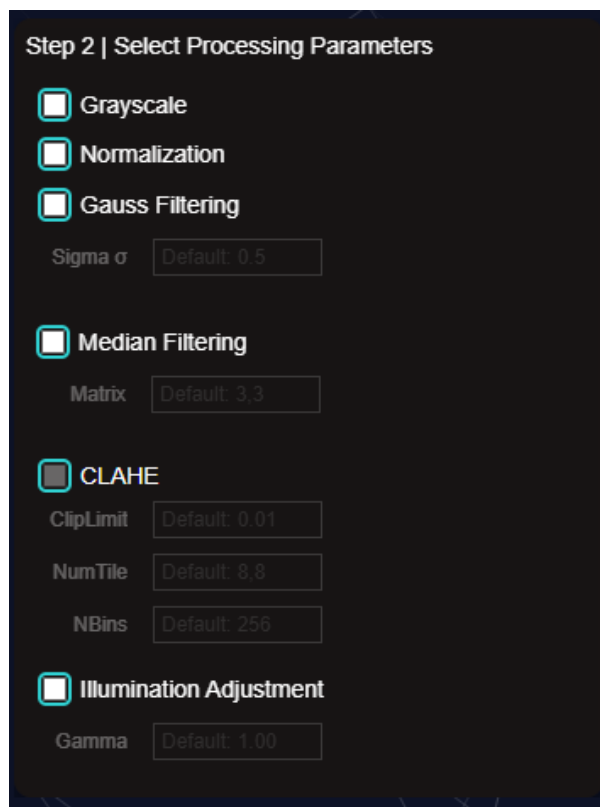
```
% Fail
```

```
% Insert code that updates DataPath Label with the user cancellation activity
```

```
end
```

Παράδειγμα 6.1

Βήμα 2 | Επιλογή παραμέτρων – μεθόδων προεπεξεργασίας



Εικόνα 64: Image Processing – Βήμα 2 της προ-επεξεργασίας, τεχνικές/μέθοδοι προ-επεξεργασίας

Στο δεύτερο βήμα της καρτέλας Image Processing, γίνεται η επιλογή των τεχνικών – μεθόδων προ-επεξεργασίας τις οποίες μπορεί να εφαρμόσει ο χρήστης στο dataset. Σε αυτό το βήμα, η σειρά επιλογής των τεχνικών – μεθόδων, θα είναι και η σειρά με την οποία θα εφαρμοστούν πάνω στις εικόνες. Αυτό επιτυγχάνεται με την αποθήκευση των τεχνικών στην σειρά που προστίθενται – αφαιρούνται μέσα από ένα cell array και στη συνέχεια εφαρμόζοντας αυτές μέσω μιας for loop.

Παράδειγμα:

```
preprocess_params = {};  
  
if Grayscale_CheckBox.Value == 1  
preprocess_params{end+1} = "grayscale";  
end  
  
>> preprocess_params = {'grayscale'}
```

Παράδειγμα 6.2

Αναλυτικά η κάθε τεχνική – μέθοδος λειτουργεί ως εξής:

Grayscale: Grayscale

Εφόσον επιλεχθεί η μετατροπή των εικόνων (φωτογραφιών) από 3 κανάλια (RGB) σε 1 κανάλι (γκρι), ακολουθούνται τα παρακάτω στάδια:

1. Μετατροπή των τιμών της εικόνας σε double precision (διπλής ακρίβειας – κινητής υποδιαστολής) κάνοντας χρήση της συνάρτησης **im2double()** [41] φέροντας τις τιμές των εικόνων από εύρος 0..255 σε 0..1.
2. Μετατροπή της εικόνας από 3 κανάλια σε 1 κανάλι (RGB to Gray) κάνοντας χρήση της εντολής **rgb2gray()** [41]. Η συνάρτηση **rgb2gray()** μετατρέπει τις τιμές RGB σε τιμές κλίμακας του γκρι, δημιουργώντας έναν συντελεστή αθροίσματος βάρους των στοιχείων **R**, **G** και **B** [41]:

$$(0.2989 * R) + (0.5870 * G) + (0.1140 * B) \quad \text{Τύπος 6.3}$$

Normalization: Normalization

Εφόσον επιλεχθεί η κανονικοποίηση των εικόνων (φωτογραφιών), ακολουθούνται οι εξής διαδικασίες από την εξατομικευμένη συνάρτηση **img_normalization()**:

1. Κανονικοποίηση (Z-Score Normalization):

Αρχικά, η εικόνα μετατρέπεται σε διπλή ακρίβεια, συμβολίζεται ως **imgnormalized**. Υπολογίζονται η μέση τιμή (μ) και η τυπική απόκλιση (σ) των τιμών των pixel στο **imgnormalized**. Η κανονικοποίηση Z-Score εφαρμόζεται στην εικόνα με τον ακόλουθο τύπο:

$$imgnormalized = \frac{(imgnormalized - \mu)}{\sigma} \quad \text{Τύπος 6.4}$$

2. Κανονικοποίηση Ελάχιστου-Μέγιστου (Min-Max Normalization):

Οι τιμές των pixel στο **imgnormalized** κανονικοποιούνται περαιτέρω στο εύρος [0,1] με τη μέθοδο της κανονικοποίησης Ελάχιστου-Μέγιστου:

$$imgnormalized = \frac{(imgnormalized - \min(imgnormalized))}{\max(imgnormalized) - \min(imgnormalized)} \quad \text{Τύπος 6.5}$$

3. Κλιμάκωση στο Εύρος [0, 255]:

Οι τιμές των pixel στο *img*normalized κλιμακώνονται από το εύρος [0, 1] στο εύρος [0, 255] με τον ακόλουθο τύπο:

$$img_{normalized} = img_{normalized} * 255$$

Τύπος 6.6

4. Μετατροπή σε uint8:

Τελικώς, το *img*normalized μετατρέπεται σε δεδομένα uint8, διασφαλίζοντας ότι οι τιμές των pixel αναπαριστώνται ως ακέραιοι αριθμοί στο εύρος από 0 έως 255.

Gauss Filtering: Gauss Filtering

Εφόσον επιλεχθεί η επιλογή για το γκαουσιανό φιλτράρισμα των εικόνων (φωτογραφιών), ενεργοποιείται και η επιλογή εισαγωγής εξατομικευμένης τιμής σ . Για το φιλτράρισμα γίνεται χρήση της συνάρτησης **imgaussfilt()** [41], με τιμή σ (τυπικής απόκλισης) που θέτει ο χρήστης ή αλλιώς με τιμή 0.5.

Η εφαρμογή του γκαουσιανού φίλτρου βοηθά στην απομάκρυνση του υψηλής συχνότητας θορύβου, κάνοντας τις ακμές και τα αντικείμενα να φαίνονται πιο ομαλά. Επιπλέον, το φίλτρο διατηρεί τα χαμηλής συχνότητας χαρακτηριστικά της εικόνας, ώστε να μην χαθούν σημαντικές πληροφορίες κατά τη διάρκεια της εξομάλυνσης.

Median Filtering: Median Filtering

Εφόσον επιλεχθεί η επιλογή για το φιλτράρισμα μεσαίας τιμής των εικόνων (φωτογραφιών), ενεργοποιείται και η επιλογή εισαγωγής εξατομικευμένης τιμής m για το μέγεθος του φίλτρου σε $[m,n]$. Για το φιλτράρισμα γίνεται χρήση της συνάρτησης **medfilt2()** [41], με τιμή m (μέγεθος φίλτρου – πίνακας m,n) που θέτει ο χρήστης ή αλλιώς με τιμή [3,3].

CLAHE: CLAHE

Εφόσον επιλεχθεί η επιλογή για τη προσαρμοστική εξίσωση ιστογράμματος περιορισμένης αντίθεσης των εικόνων (φωτογραφιών), ενεργοποιείται και η επιλογή εισαγωγής εξατομικευμένης τιμής για τις παραμέτρους ClipLimit, NumTile και NBins. Η επιλογή CLAHE είναι διαθέσιμη μόνο εφόσον επιλεχθεί η επιλογή Grayscale, και εφαρμόζεται

κάνοντας χρήση της συνάρτησης **adapthisteq()** [41] με τις τιμές που θέτει ο χρήστης για τις παραμέτρους ClipLimit, NumTile και NBins, αλλιώς:

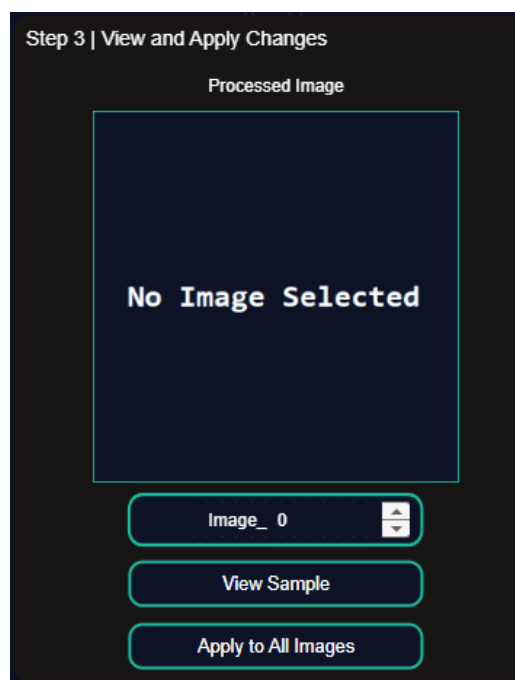
- ClipLimit = 0.01
- NumTile = 8,8
- NBins = 256

Η συνάρτηση **adapthisteq()** είναι χρήσιμη για εικόνες που περιέχουν περιοχές με διαφορετικά επίπεδα φωτεινότητας, όπως ιατρικές εικόνες, εικόνες σε συνθήκες ασθενούς φωτισμού ή εικόνες με σκιές.

Illumination Adjustment: **Illumination Adjustment**

Εφόσον επιλεχθεί η επιλογή για της προσαρμογής των τιμών της έντασης των εικόνων (φωτογραφιών), ενεργοποιείται και η επιλογή εισαγωγής εξατομικευμένης τιμής για τη παράμετρο Gamma. Η εφαρμογή αυτής της τεχνικής γίνεται μέσω της χρήσης της συνάρτησης **imadjust()** [41] με την τιμή Gamma που καθορίζει ο χρήστης, αλλιώς η τιμή είναι 1.00, όπου δεν υπάρχει καμία αλλαγή στην εικόνα. Η χρήση της συνάρτησης **imadjust()** είναι χρήσιμη για τη βελτίωση της ορατότητας στις περιοχές της εικόνας με χαμηλή αντίθεση ή περιορισμένη δυναμική εύρους τιμών. Παρέχει τη δυνατότητα προσαρμογής των τιμών των εικονοστοιχείων ώστε να αναδείξει λεπτομέρειες που δυσκολεύονται να διακριθούν αρχικά.

Βήμα 3 – Προβολή και Εφαρμογή Αλλαγών (Τεχνικών Επεξεργασίας)



Εικόνα 65: Image Processing – Βήμα 3 της καρτέλας προ-επεξεργασίας

Στο τρίτο βήμα της καρτέλας Image Processing, γίνεται μια γρήγορη προβολή των τεχνικών – μεθόδων προ-επεξεργασίας που επιλέχθηκαν, εφαρμόζοντας τις τεχνικές αυτές με τη σειρά που έγινε η επιλογή τους, πάνω σε μια εικόνα τη φορά η οποία εμφανίζεται βάσει του δείκτη Image_. Με αυτό το τρόπο, η προβολή – εφαρμογή διαφορετικών φίλτρων γίνεται ταχύτερα, αφού η διαδικασία εφαρμόζεται σε μια εικόνα τη φορά, η οποία εμφανίζεται και στον χώρο του Processed Image. Εάν ο χρήστης είναι ευχαριστημένος από το αποτέλεσμα, μπορεί να προχωρήσει στην εφαρμογή των τεχνικών αυτών σε όλο το dataset μέσω το Apply to All Images.

Δείκτης Image_:

Ο δείκτης Image_ αντικατοπτρίζει την εικόνα που θα προβληθεί, ξεκινώντας από τον αριθμό 1 με το μέγιστο να είναι αριθμός των εικόνων στο φάκελο στο σύνολο τους.

View Sample & Apply to All Images:

Στην περίπτωση των View Sample & Apply to All Images κουμπιών, τόσο για την εφαρμογή και προβολή ενός δείγματος (View Sample) αλλά και στην εφαρμογή σε όλες τις εικόνες (Apply to All Images), ακολουθείται η ίδια διαδικασία, με την διαφορά ότι στο View Sample, η εφαρμογή και προβολή γίνεται μια εικόνα την φορά.

Η διαδικασία που ακολουθείται είναι εξής:

1. Προετοιμασία και αποθήκευση των εικόνων σε κελιά (Cell Array)
2. Προβολή/εφαρμογή των τεχνικών που βρίσκονται εντός του Cell Array **preprocess_params**, με την σειρά που εμφανίζονται μέσα σε αυτή τη λίστα.
3. Στην περίπτωση του View Sample, η εφαρμογή των τεχνικών γίνεται για μια εικόνα με δείκτη Image_ ενώ στο Apply to All Images γίνεται για όλες τις εικόνες που βρίσκονται στη λίστα.

Και η εφαρμογή των παραπάνω γίνεται κάνοντας χρήση της συνάρτησης **cellfun()** [41] η οποία επιτρέπει την εφαρμογή διάφορων συναρτήσεων σε κάθε κελί με βελτιστοποιημένο τρόπο, χωρίς να υπάρχει η ανάγκη χρήσης του παραδοσιακού τρόπου for loop, όπως εμφανίζεται στο παράδειγμα 6.7:

Παράδειγμα:

```
>> preprocess_params = {'Grayscale', 'Normalization'}

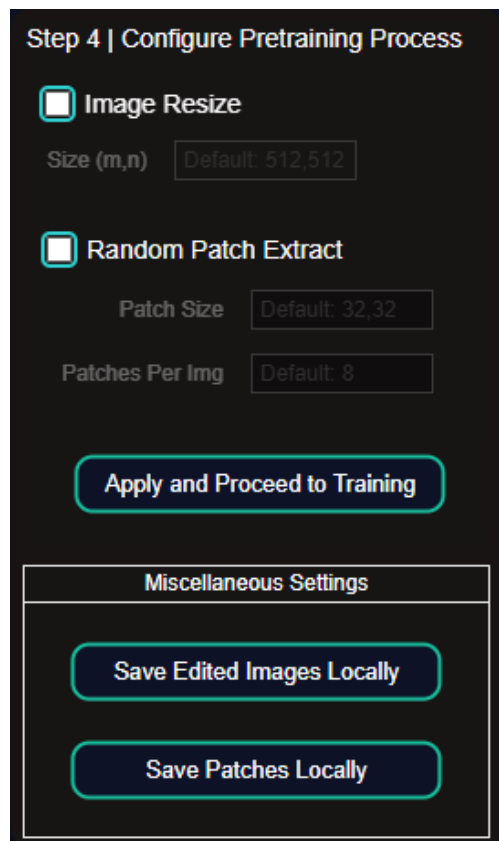
for i=1:numel(preprocess_params)
current_param = preprocess_params{i};

    switch current_param
    case 'Grayscale'
        image_trainingDB = cellfun(@im2double, training_imageDB, 'UniformOutput', false);
        image_trainingDB = cellfun(@rgb2gray, training_imageDB, 'UniformOutput', false);

    case 'Normalization'
        image_trainingDB = cellfun(@img_normalization, training_imageDB,
        'UniformOutput', false);
    end
end
```

Παράδειγμα 6.7

Βήμα 4 – Επιπλέον ρύθμιση της διαδικασίας προ-εκπαιδύσης



Εικόνα 66: Image Processing – Βήμα 4 της καρτέλας προ-επεξεργασίας

Στο τέταρτο βήμα της καρτέλας Image Processing, βρίσκεται η διαδικασία Image Resize και Random Patch Extract, οι οποίες εφόσον επιλεχθούν,

ενεργοποιούν την εισαγωγή εξατομικευμένης τιμής για το μέγεθος της νέας εικόνας $Size(m,n)$ σε μορφή πίνακα ή/και εφόσον το επιθυμεί ο χρήστης, να δημιουργήσει υπό-τμήματα – περιοχές (Patch Extract) των εικόνων αυτών με ή χωρίς την χρήση του Image Resize, δίνοντας του δυνατότητα να θέσει το μέγεθος των υπό-τμημάτων – περιοχών σε πίνακα και τον αριθμό των τυχαίων υπό-τμημάτων στο μέγεθος που έχει επιλέξει, που θα εξαχθούν από την κάθε εικόνα.

Image Resize:

Εφόσον επιλεχθεί η επιλογή αλλαγής μεγέθους των εικόνων (φωτογραφιών), τότε ενεργοποιείται και η δυνατότητα εισαγωγής εξατομικευμένης τιμής για το νέο μέγεθος της εικόνας $Size(m,n)$ και η εφαρμογή αυτής της τεχνικής γίνεται μέσω της συνάρτησης **imresize()** [41] στην οποία περνάει η τιμή m,n του πίνακα για το νέο μέγεθος της εικόνας. Εάν ο χρήστης δεν εισάγει κάποια τιμή m,n (π.χ. 256,256), αυτομάτως η τιμή τίθεται ως 512,512.

Random Patch Extract:

Εφόσον επιλεχθεί η επιλογή τυχαίας εξαγωγής τμημάτων – περιοχών από τις εικόνες (φωτογραφίες), τότε ενεργοποιείται και η δυνατότητα εισαγωγής εξατομικευμένης τιμής για το μέγεθος του τμήματος που θα εξαχθεί από την εικόνα $patch_size(m,n)$ και ο αριθμός των τυχαίων τμημάτων – περιοχών που πρέπει να εξαχθούν από την κάθε εικόνα $npatches$. Σε περίπτωση που ο χρήστης δεν εισάγει κάποια τιμή στις 2 παραπάνω παραμέτρους, οι τιμές τίθενται αυτομάτως σε $patch_size = 64,64$ και $npatches = 8$. Για την εφαρμογή αυτής της διαδικασίας, γίνεται χρήση της συνάρτησης **randomPatchExtractionDatastore()** [41]

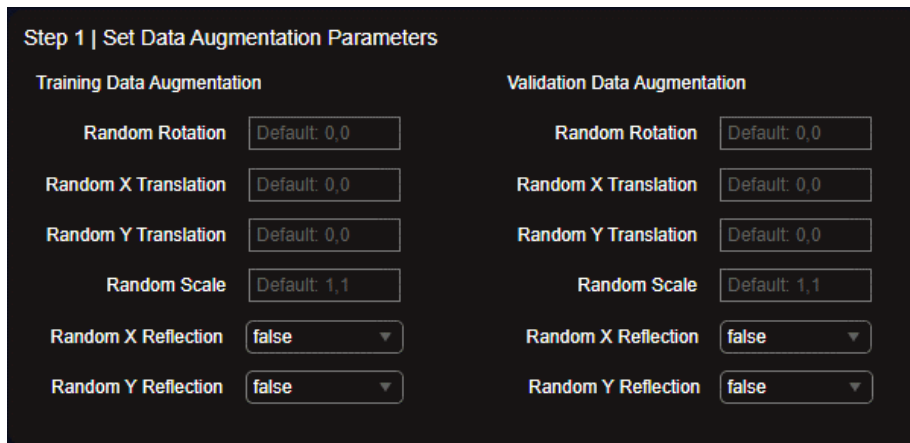
Apply and Proceed to Training:

Το κουμπί είναι το τελευταίο στάδιο της καρτέλας για το κομμάτι της προετοιμασίας, προβολής και εφαρμογής των εικόνων (φωτογραφιών) και έχει την ίδια συμπεριφορά με τα κουμπιά του βήματος 3, όπου η εφαρμογή των αλλαγών γίνεται μέσω της συνάρτησης **cellfun()** [41]. Στη προκειμένη περίπτωση, μόλις οι αλλαγές έχουν εφαρμοστεί πλήρως και επιτυχώς, γίνεται αυτομάτως η μετάβαση στην επόμενη καρτέλα Training, έχοντας πλέον στη μνήμη το νέο – προ-επεξεργασμένο dataset που δημιουργήθηκε μέσω της καρτέλας Image Processing.

6.2 Aria App: Backend – Training Tab

Στην παρούσα υπό-ενότητα, θα επεξηγηθεί το κομμάτι του back-end της εφαρμογής, στη δεύτερη καρτέλα Training.

Βήμα 1 | Παράμετροι επαύξησης δεδομένων



Εικόνα 67: Training – Βήμα 1 της καρτέλας εκπαίδευσης

Στο βήμα αυτό, γίνεται η επαύξηση των δεδομένων (data augmentation) μέσω εφαρμογής διαφόρων επιλογών όπως εμφανίζονται στο παραπάνω στιγμιότυπο με εφαρμογή στα δύο διαφορετικά datasets: Training και Validation. Το data augmentation είναι προαιρετικό και σε περίπτωση που δεν επιλέξει ο χρήστης να εισάγει/μεταβάλλει τις παραπάνω επιλογές, τότε οι τιμές αυτομάτως είναι στο 0 ή false (με εξαίρεση το Random Scale, όπου το 1,1 είναι η τιμή όπου δεν εφαρμόζονται αλλαγές). Η επίτευξη των παραπάνω γίνεται μέσω συνάρτησης `imageDataAugmenter()` [41].

Random Rotation:

Τυχαία περιστροφή της εικόνας. Καθορίζει το εύρος των γωνιών περιστροφής που μπορεί να εφαρμοστεί στις εικόνες. Π.χ.: -45, 45.

Random X Translation:

Τυχαίες μετατοπίσεις της εικόνας στον άξονα X, βασισμένο σε αριθμό pixels. Πχ: -5, 5.

Random Y Translation:

Τυχαίες μετατοπίσεις της εικόνας στον άξονα Y, βασισμένο σε αριθμό pixels. Πχ: -5, 5

Random Scale:

Τυχαία αλλαγή της κλίμακας της εικόνας. Καθορίζει το εύρος των κλιμακώσεων που μπορεί να εφαρμοστεί στις εικόνες. Πχ: 0.8, 1.2

Random X Reflection:

Ανάκλαση της εικόνας στον άξονα Χ. Εάν είναι ενεργοποιημένη η επιλογή (true), οι εικόνες μπορεί να ανακλαστούν οριζόντια με πιθανότητα 50%.

Random Y Reflection:

Ανάκλαση της εικόνας στον άξονα Υ. Εάν είναι ενεργοποιημένη η επιλογή (true), οι εικόνες μπορεί να ανακλαστούν κάθετα με πιθανότητα 50%.

Παράδειγμα:

% Create DataAugmentationStore and add transformations to Training Data

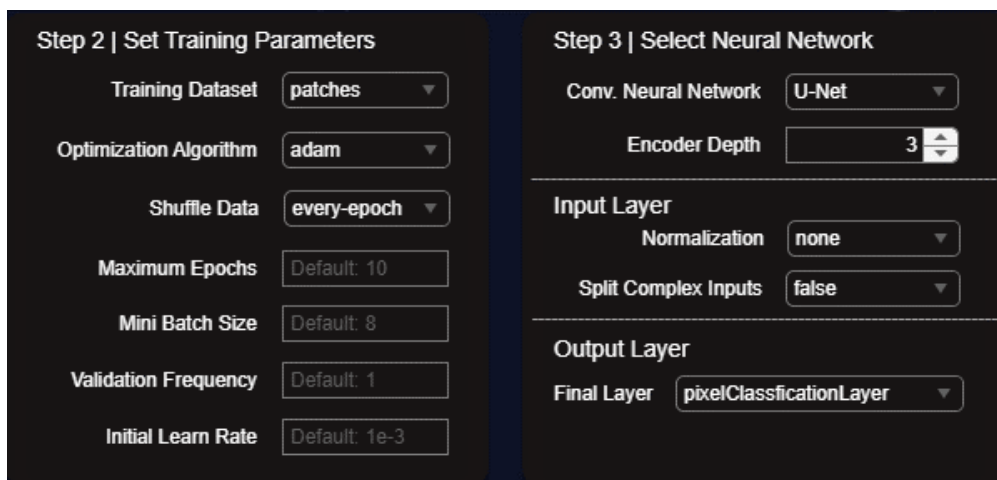
```
training_augmenter = imageDataAugmenter( ...  
"RandRotation", [-45 45], ...  
'RandXTranslation', [-5 5], ...  
'RandYTranslation', [-10 10], ...  
'RandScale', [0.8, 1.2], ...  
'RandXReflection', true, ...  
'RandYReflection', true);
```

% Create DataAugmentationStore and add transformations to Validation Data

```
validation_augmenter = imageDataAugmenter( ...  
"RandRotation", [-15 15], ...  
'RandXTranslation', [-2 2], ...  
'RandYTranslation', [-5 5], ...  
'RandScale', [0.9, 1.1], ...  
'RandXReflection', true, ...  
'RandYReflection', false);
```

Παράδειγμα 6.8

Βήμα 2 και 3 | Επιλογή δικτύου και παραμέτρων εκπαίδευσης



Εικόνα 68: Training – Βήμα 2 και Βήμα 3 της καρτέλας εκπαίδευσης

Στο βήμα 2, γίνεται η επιλογή των παραμέτρων εκπαίδευσης του δικτύου που θα επιλεχθεί στο βήμα 3 (στη παρούσα εργασία, το δίκτυο είναι το U-Net). Οι παράμετροι εκπαίδευσης είναι στο σύνολο 7, και γίνεται η χρήση αυτών μέσω της συνάρτησης **trainingOptions()** [41] και αναλύονται ως εξής:

Training Dataset:

Σε αυτή τη παράμετρο εκπαίδευσης, γίνεται η επιλογή του dataset στην οποία το μοντέλο θα εκπαιδευτεί. Οι επιλογές είναι a) patches και b) full_size, όπου δημιουργήθηκαν στην προηγούμενη καρτέλα, Image Processing.

Optimization Algorithm:

Οι αλγόριθμοι βελτιστοποίησης που είναι διαθέσιμοι περιλαμβάνουν τους παρακάτω:

1. **SGDM (Stochastic Gradient Descent with Momentum)**
2. **ADAM (Adaptive Moment Estimation)**
3. **RMSProp (Root Mean Square Propagation)**

Shuffle Data:

Η παράμετρος shuffle data χρησιμοποιείται για να καθορίσει εάν τα δεδομένα εκπαίδευσης θα μπαίνουν σε μια τυχαία σειρά σε κάθε επανάληψη της εκπαίδευσης (epoch-to-epoch). Οι διαθέσιμες τιμές για τη συγκεκριμένη παράμετρο είναι οι εξής:

1. **every-epoch**

Στη τιμή αυτή, τα δεδομένα παίρνουν τυχαία σειρά σε κάθε νέα προσπάθεια εκπαίδευσης (epoch).

2. **once**

Στη τιμή αυτή, τα δεδομένα παίρνουν τυχαία σειρά μια φορά στην αρχή της εκπαίδευσης και βρίσκονται στην ίδια σειρά σε κάθε νέα προσπάθεια εκπαίδευσης (epoch).

3. **none**

Στη τιμή αυτή, τα δεδομένα κρατούν την ίδια σειρά στην οποία βρίσκονται, χωρίς να αλλάζει η σειρά τους.

Mini Batch Size:

Η παράμετρος minibatchsize παρέχει την δυνατότητα στον χρήστη να θέσει τον αριθμό των δεδομένων (εικόνων) που θα διοχετεύεται για την ολοκλήρωση μιας προσπάθειας (epoch). Η τιμή αυτή είναι σημαντική, καθώς εξαρτάται πλήρως από τα χαρακτηριστικά του υπολογιστή, ειδικότερα στο κομμάτι της κάρτας γραφικών που επεξεργάζεται τα δεδομένα κατά τη διάρκεια της εκπαίδευσης. Μια καλή τιμή έναρξης είναι minibatchsize = 32.

Initial Learn Rate:

Η παράμετρος InitialLearnRate αναφέρεται στον αρχικό ρυθμό μάθησης που χρησιμοποιείται κατά την έναρξη της εκπαίδευσης ενός μοντέλου μηχανικής μάθησης, όπως ένα νευρωνικό δίκτυο. Ο ρυθμός αυτός καθορίζει πόσο γρήγορα το μοντέλο "μαθαίνει" από τα δεδομένα εκπαίδευσης και προσαρμόζει τα βάρη του κατάλληλα. Η τιμή αυτή μπορεί να έρθει μέσα από πειραματισμό και μια καλή τιμή έναρξης είναι 1e-03.

Maximum Epochs / Validation Frequency

Η παράμετρος `maximum epochs` καθορίζει τον μέγιστο αριθμό των επαναλήψεων που θα εκτελεστεί η εκπαίδευση. Ένα epoch αναφέρεται στο πέρασμα όλων των δεδομένων εκπαίδευσης μέσα από το μοντέλο. Η παράμετρος αυτή περιορίζει τον αριθμό των epoch που θα εκτελεστεί η εκπαίδευση, βοηθώντας να αποφευχθεί η υπερεκπαίδευση (overfitting) του μοντέλου στα δεδομένα εκπαίδευσης.

Η παράμετρος `validation frequency` καθορίζει πόσα epochs περνούν πριν από την εκτέλεση αξιολόγησης (validation) στο μοντέλο. Κατά τη διάρκεια της εκπαίδευσης, γίνεται αξιολόγηση του μοντέλου για να παρακολουθούμε την απόδοσή του και να αποφασίζουμε αν πρέπει να σταματήσουμε την εκπαίδευση ή να κάνουμε προσαρμογές. Η παράμετρος αυτή ελέγχει πόσο συχνά θα εκτελούνται αυτές οι αξιολογήσεις, και μπορεί να εξοικονομήσει χρόνο κατά τη διάρκεια της εκπαίδευσης, αλλά και να βοηθήσει στην προσαρμογή των υπερπαραμέτρων του μοντέλου.

Παράδειγμα:

```
options = trainingOptions('adam', ...  
    'MaxEpochs', MaximumEpochs, ...  
    'MiniBatchSize', MiniBatchSz, ...  
    'Shuffle', 'every-epoch', ...  
    'ValidationFrequency', 10, ...  
    'ValidationData', validation_imageDB, ...  
    'InitialLearnRate', Initial_LR);
```

Παράδειγμα 6.9

Αφού εισαχθούν όλες οι παραπάνω τιμές στις παραμέτρους και δημιουργηθεί η μεταβλητή `options`, το επόμενο βήμα είναι η επιλογή κάποιων ρυθμίσεων – επιλογών που βρίσκονται στο βήμα 3 και είναι οι εξής:

Encoder Depth:

Η παράμετρος `"encoderDepth"` [41] στο μοντέλο Unet αναφέρεται στο βάθος του κωδικοποιητή (encoder) του δικτύου. Το στάδιο encoder είναι η πρώτη μισή του δικτύου Unet και είναι υπεύθυνη για την εξαγωγή χαρακτηριστικών από την είσοδο. Καθορίζει τον αριθμό των συνελίξεων (convolutions) που θα εφαρμοστούν στο στάδιο αυτό. Κάθε συνέλιξη

αποτελείται από φίλτρα που εκπαιδεύονται για την ανίχνευση διαφορετικών χαρακτηριστικών στην εικόνα. Όσο μεγαλύτερο είναι το "encoderDepth," τόσο περισσότερα στάδια συνελίξεων θα εφαρμοστούν.

Αυξημένο βάθος μπορεί να οδηγήσει σε καλύτερη αναπαράσταση των χαρακτηριστικών, αλλά απαιτεί επίσης περισσότερους υπολογιστικούς πόρους και μεγαλύτερο χρόνο εκπαίδευσης. Συνολικά, η παράμετρος "encoderDepth" είναι σημαντική για την προσαρμογή του μοντέλου U-Net και η τιμή του βασίζεται σε σημαντικό βαθμό στα χαρακτηριστικά του υπολογιστή και συγκεκριμένα στη κάρτα γραφικών του. Μια τιμή έναρξης, θα μπορούσε να είναι η τιμή 3, όπως φαίνεται στο στιγμιότυπο.

(Image) Input Layer:

Το **imageInputLayer** [41] παρέχει τον τρόπο διαμόρφωσης στην είσοδο των εικόνων του δικτύου και συγκεκριμένα μέσω της εφαρμογή δίνεται η δυνατότητα να κανονικοποίησης (Normalization) αυτών προτού εισέλθουν στο στάδιο της εκπαίδευσης και την επιλογή Split Complex Inputs, η οποία σε ορισμένες περιπτώσεις, οι εικόνες μπορεί να περιέχουν πληροφορία για το πραγματικό μέρος (π.χ. φωτεινότητα) και το φανταστικό μέρος (π.χ. χρώματα), και τα χωρίζει σε πραγματικό και φανταστικό μέρος. Οι παραπάνω επιλογές βρίσκονται ως μέρος της συνάρτησης **imageInputLayer()** [41].

- **Normalization:**

1. none
2. zerocenter
3. zscore
4. rescale-symmetric
5. rescale-zero-one

- **Split Complex Inputs:**

1. true
2. false

Παράδειγμα Input Layer:

```
newInputLayer = imageInputLayer(imageSize, ...  
"Normalization", zscore", ...  
"SplitComplexInputs", false);
```

```
lgraph_unet = replaceLayer(lgraph_unet, "ImageInputLayer", newInputLayer);
```

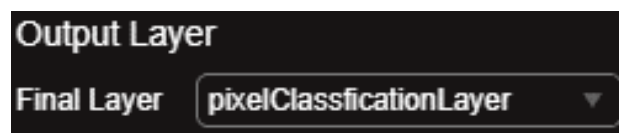
Παράδειγμα 7.1

Output Layer:

Το outputLayer καθορίζει τα χαρακτηριστικά της εξόδου που παράγεται από το δίκτυο. Το outputLayer είναι συνήθως το τελευταίο επίπεδο του δικτύου και είναι υπεύθυνο για τη δημιουργία των τελικών προβλέψεων ή εξόδων με βάση τα δεδομένα εισόδου. Στη συγκεκριμένη περίπτωση της κατάτμησης των αγγείων, υπάρχουν διαθέσιμοι ορισμένοι τύποι επιπέδων εξόδου που χρησιμοποιούνται συχνά.

Τύποι επιπέδων (1/3):

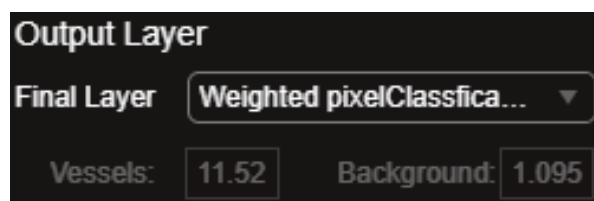
- **pixelClassificationLayer:** Σε αυτό το επίπεδο κάθε εικονοστοιχείο στην εικόνα εκχωρείται μια ετικέτα ή κλάση. Το pixelClassificationLayer απαιτεί τον αριθμό των κλάσεων ως είσοδο. Υπολογίζει το pixel-wise cross-entropy loss κατά τη διάρκεια της εκπαίδευσης [41].



Εικόνα 69: Training – Στάδιο επιλογής final output layer

Τύποι επιπέδων (2/3):

- **weightedPixelClassificationLayer:** Αυτό το επίπεδο επιτρέπει την ανάθεση διαφορετικών βαρών σε διάφορες κλάσεις κατά τον υπολογισμό της συνάρτησης απώλειας. Αυτό μπορεί να είναι χρήσιμο όταν αντιμετωπίζετε ανισορροπία κλάσεων. Στην συγκεκριμένη περίπτωση, τα βάρη για τις εικόνες υπολογίζονται αυτόματα μέσω μίας εξατομικευμένης συνάρτησης η οποία θα παρουσιαστεί παρακάτω στο παράδειγμα [41].



Εικόνα 70: Training – Παράδειγμα εμφάνισης classWeights μέσω της επιλογής Weighted final output layer

Τύποι επιπέδων (2/3):

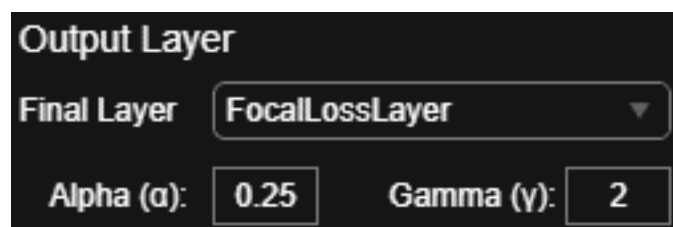
- **dicePixelClassificationLayer:** Το `dicePixelClassificationLayer` ενσωματώνει τη συνάρτηση απώλειας Dice ως μέρος της διαδικασίας εκπαίδευσης. Στοχεύει να βελτιώσει την απόδοση του δικτύου βελτιστοποιώντας απευθείας τον συντελεστή Dice [41].



Εικόνα 71: Training – Επιλογή `dicePixelClassificationLayer` ως final output layer

Τύποι επιπέδων (3/3):

- **focalLossLayer:** Η συνάρτηση απώλειας focal έχει σχεδιαστεί για να αντιμετωπίσει το πρόβλημα της ανισορροπίας των κλάσεων σε εργασίες ταξινόμησης. Αυξάνει το βάρος στα δείγματα εκείνα που ανήκουν στις δύσκολες κλάσεις, ενώ δίνει λιγότερη βαρύτητα στα εύκολα. Το `focalLossLayer` είναι ιδιαίτερα χρήσιμο όταν αντιμετωπίζετε ανισορροπία κλάσεων ή όταν ορισμένες κλάσεις είναι σπάνιες. Η παραμετροποίηση του γίνεται μέσω των τιμών α (α) και σ (σ) [41].



Εικόνα 72: Training – επιλογή `FocalLossLayer` ως final output layer

Στην επόμενη σελίδα, θα δοθούν παραδείγματα του κώδικα για το κάθε επίπεδο (layer) που αναφέρθηκε νωρίτερα.

Παράδειγμα των ξεχωριστών layers:

1. pixelClassificationLayer:

```
newOutputLayer = pixelClassificationLayer('Classes', {'vessel', 'background'},  
'Name', 'new_segmentation_layer');  
  
lgraph_unet = replaceLayer(lgraph_unet, "Segmentation-Layer", newOutputLayer);
```

Παράδειγμα 7.2

2. weightedPixelClassificationLayer:

```
numberPixels = sum(tbl.PixelCount);  
frequency = tbl.PixelCount / numberPixels;  
classWeights = 1 ./ frequency;
```

```
VesselsWeight = classWeights(1);  
BackgroundWeight = classWeights(2);
```

```
newOutputLayer = pixelClassificationLayer('Classes', {'vessel', 'background'},  
'ClassWeights', [classWeights], 'Name', 'new_segmentation_layer');  
lgraph_unet = replaceLayer(lgraph_unet, "Segmentation-Layer", newOutputLayer);
```

Παράδειγμα 7.3

3. dicePixelClassificationLayer:

```
newOutputLayer = dicePixelClassificationLayer('Classes', {'vessel', 'background'},  
'Name', 'new_segmentation_layer');
```

```
lgraph_unet = replaceLayer(lgraph_unet, "Segmentation-Layer", newOutputLayer);
```

Παράδειγμα 7.4

4. focalLossLayer:

```
newOutputLayer = focalLossLayer('Classes', {'vessel', 'background'}, 'Alpha',  
alpha_value, 'Gamma', gamma_value, 'Name', 'new_segmentation_layer');
```

```
lgraph_unet = replaceLayer(lgraph_unet, "Segmentation-Layer", newOutputLayer);
```

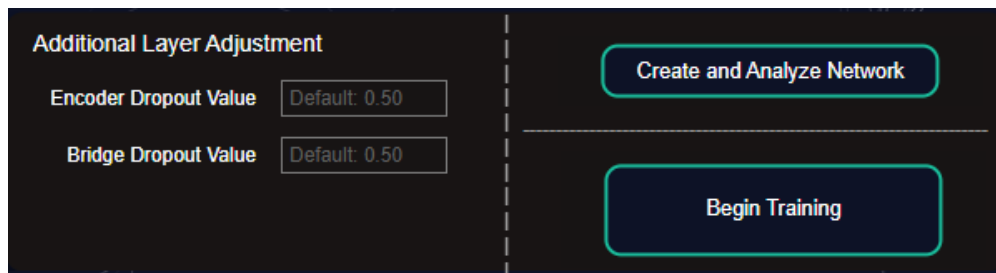
Παράδειγμα 7.5

Στο παράδειγμα 7.3, αρχικά υπολογίζεται ο συνολικός αριθμός των pixel που βρίσκονται στον πίνακα tbl, για κάθε κλάση {vessel, background}.

Ύστερα, διαιρούμε τον αριθμό όλων των pixel της εκάστοτε εικόνας, με τον αριθμό των pixel που ανήκουν σε κάθε κλάση, δίνοντας μας την συχνότητα εμφάνισης της κάθε κλάσης ανάμεσα σε όλα τα pixels.

Τελικώς, διαιρούμε τον αριθμό 1 με τον αριθμό της κάθε συχνότητας και υπολογίζουμε το βάρος της κάθε κλάσης. Όσο μεγαλύτερο το νούμερο, τόσο μικρότερη η εκπροσώπηση της εκάστοτε κλάσης.

Βήμα 4 | Προαιρετική ρύθμιση DropOut και εκκίνηση εκπαίδευσης



Εικόνα 73: Training – Επιπλέον πλαίσιο παραμετροποίησης DropOut layer και δημιουργίας δικτύου

Στο βήμα 4, στο αριστερό κομμάτι γίνεται η προαιρετική ρύθμιση των DropOut layers που βρίσκονται στο κομμάτι του encoder και bridge [41]. Και στο δεξί κομμάτι, βρίσκονται οι τελευταίες επιλογές της καρτέλας Training, τα οποία είναι η δημιουργία και ανάλυση του δικτύου (Create and Analyze Network) και η εκκίνηση της εκπαίδευσης (Begin Training).

DropOut Layer:

Το επίπεδο dropout layer είναι μια τεχνική που χρησιμοποιείται σε νευρωνικά δίκτυα για την αντιμετώπιση του προβλήματος του overfitting.

Κατά την εκπαίδευση, το επίπεδο απόρριψης τυχαία «απορρίπτει» (απενεργοποιεί) ένα προκαθορισμένο ποσοστό των νευρώνων σε κάθε επίπεδο, κάνοντας τα δεδομένα να μην είναι πάντα παρόντα σε κάθε επίπεδο του δικτύου, επιτρέποντας σε αυτό να μάθει πιο γενικά χαρακτηριστικά.

Στην πράξη, το επίπεδο απόρριψης λειτουργεί ως εξής:

1. Για κάθε επίπεδο νευρώνων, επιλέγουμε ένα ποσοστό πιθανότητας (π.χ. 0.5).
2. Κατά τη διάρκεια της εκπαίδευσης, κάθε νευρώνας απενεργοποιείται με την πιθανότητα που έχει οριστεί. Αυτό σημαίνει ότι η έξοδος του νευρώνα είναι μηδενική.

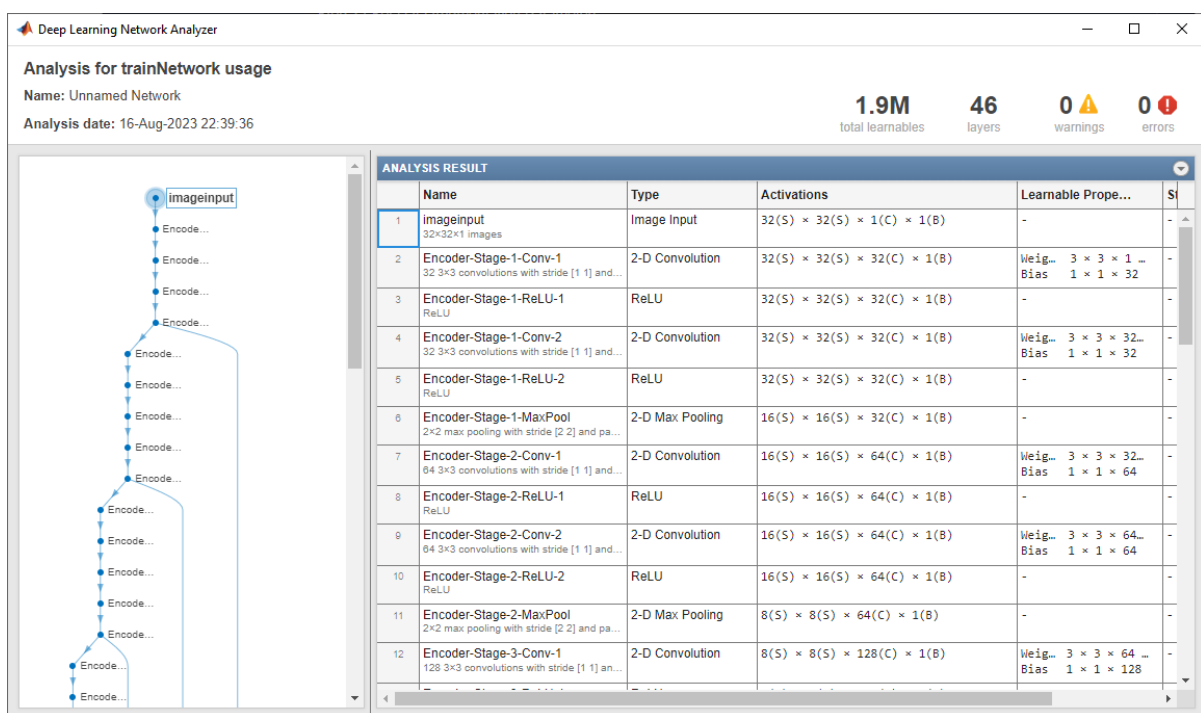
Παράδειγμα DropOut Layer στο κομμάτι του encoder:

```
value = 0.5;
newDropoutLayer = dropoutLayer(value, 'Name', 'Encoder-Stage-3-DropOut');
lgraph_unet = replaceLayer(lgraph_unet, "Encoder-Stage-3-DropOut", ...
newDropoutLayer);
```

Παράδειγμα 7.6

Create and Analyze Network:

Στο στάδιο αυτό, βάσει των προηγούμενων επιλογών της καρτέλας Training, τροποποιείται το τελικό δίκτυο **U-Net**, το οποίο έχει όνομα μεταβλητής **lgraph_unet**, το οποίο είναι τύπου **layerGraph()** [41], και επιτρέπει την προγραμματιστική επεξεργασία – μεταβολή της αρχιτεκτονικής του δικτύου και συγκεκριμένα των επιπέδων (layers) του. Μόλις γίνει η τελική αλλαγή – μεταβολή των επιπέδων αυτών, τότε χρησιμοποιείται η συνάρτηση **analyzeNetwork()** [41] η οποία αναλύει το δίκτυο και τις παραμέτρους του και προβάλλει την αρχιτεκτονική του σε ξεχωριστό παράθυρο:



Εικόνα 74: Training – Ξεχωριστό παράθυρο ανάλυσης της αρχιτεκτονικής του δικτύου

Στο σημείο αυτό, εφόσον ο χρήστης επιβεβαιώσει ότι το δίκτυο είναι όπως ήθελε να το σχεδιάσει και να παραμετροποιήσει, ξεκινά το στάδιο της εκπαίδευσης μέσω της επιλογής **Begin Training**, η οποία θα επεξηγηθεί πλήρως στην επόμενη σελίδα.

Begin Training:

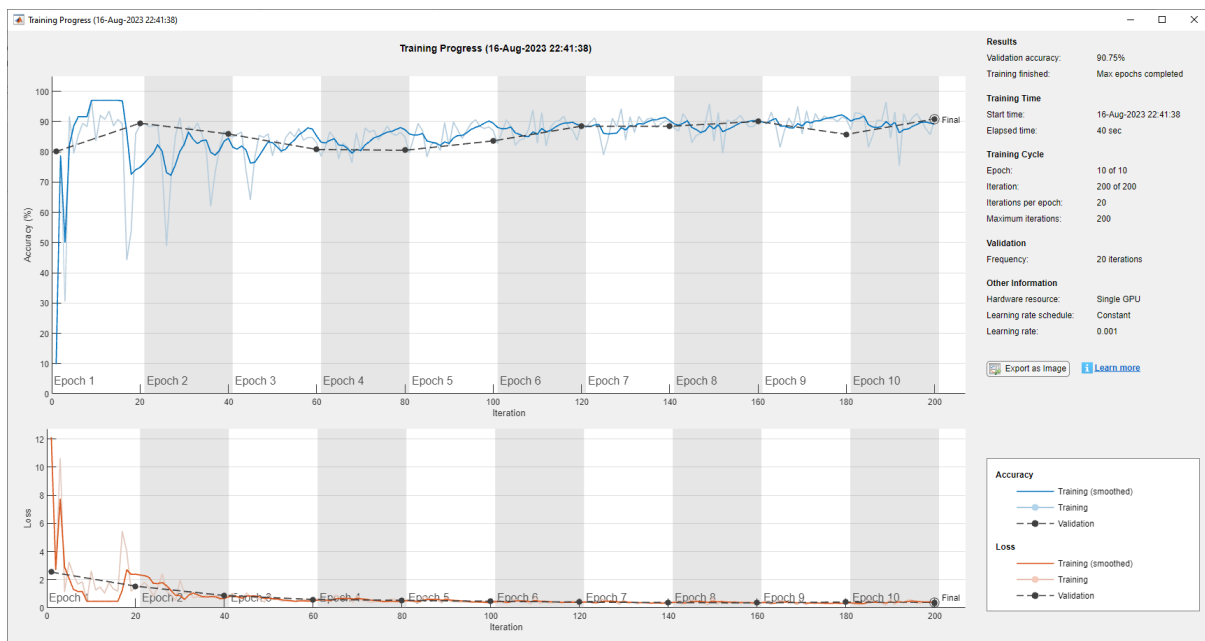
Η εκκίνηση της εκπαίδευσης γίνεται μέσω της συνάρτησης **trainNetwork()** [41] και χρησιμοποιεί τις παραμέτρους εκπαίδευσης που αναλύθηκαν προηγουμένως στην ίδια καρτέλα Training.

Παράδειγμα:

```
current_unet = trainNetwork(training_final_imds, lgraph_unet, options);
```

Παράδειγμα 7.7

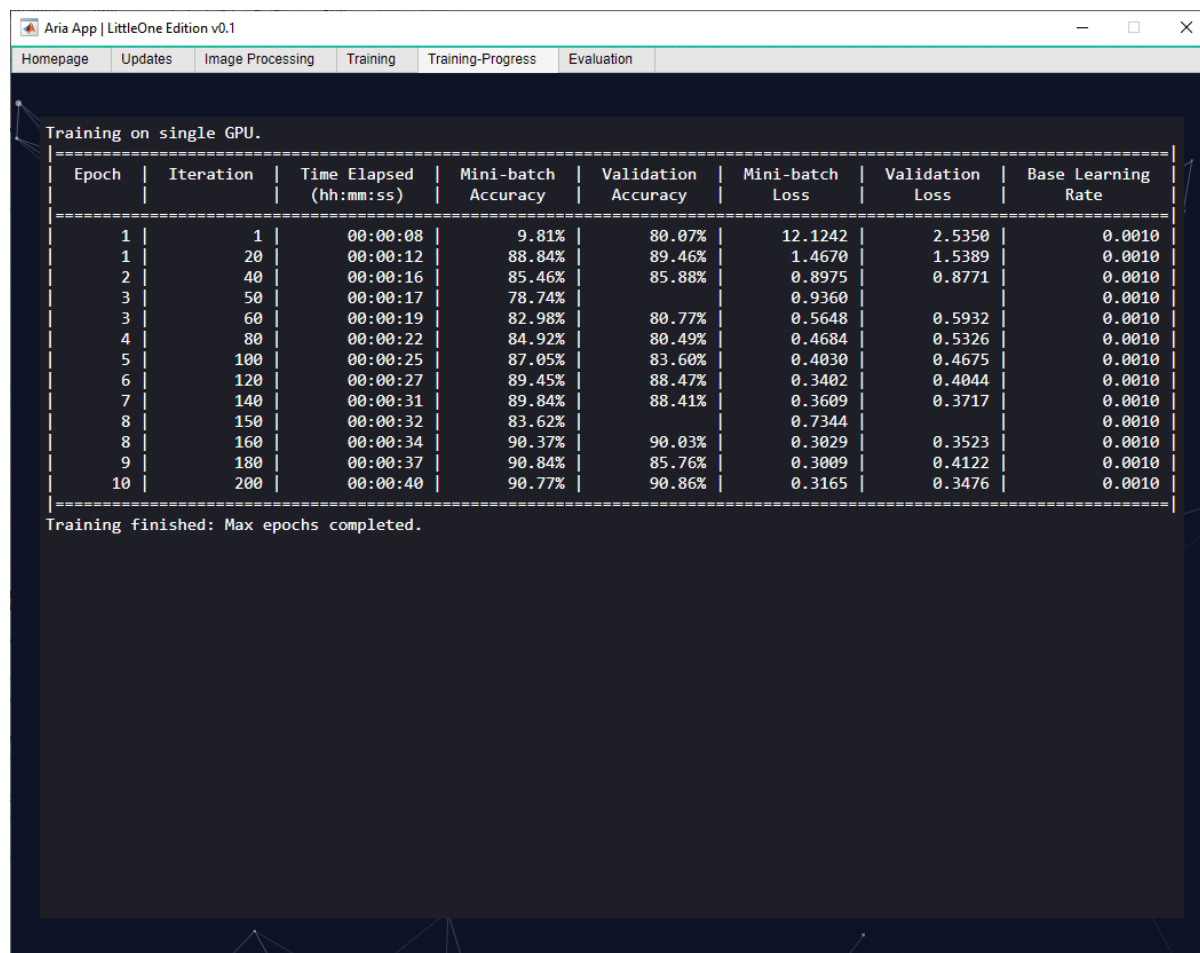
Όπου `training_final_imds` είναι η τελική έκδοση των εικόνων που θα χρησιμοποιηθεί για την εκπαίδευση (βάσει των παραμέτρων που αναλύθηκαν νωρίτερα, όπως πχ Data Augmentation), το `lgraph_unet` είναι η τελική έκδοση του δικτύου μέσω του Create and Analyze Network και το `options` είναι οι παράμετροι εκπαίδευσης που τέθηκε σε προηγούμενο βήμα, όπως αναλύθηκε παραπάνω. Εφόσον όλες οι παράμετροι έχουν εισαχθεί, τότε αμέσως ξεκινά η εκπαίδευση του μοντέλου σε ξεχωριστό παράθυρο:



Εικόνα 75: Training – Ξεχωριστό παράθυρο παρουσίασης προόδου εκπαίδευσης

6.3 Aria App: Backend – Training Progress Tab

Στην παρούσα υπό-ενότητα, θα επεξηγηθεί εν συντομία το κομμάτι του back-end της εφαρμογής, στη τρίτη καρτέλα Training Progress για τη πρόοδο της εκπαίδευσης του δικτύου.



The screenshot shows the 'Training Progress' tab in the Aria App. The window title is 'Aria App | LittleOne Edition v0.1'. The navigation bar includes 'Homepage', 'Updates', 'Image Processing', 'Training', 'Training-Progress', and 'Evaluation'. The main content area displays 'Training on single GPU.' followed by a table of training metrics. The table has 8 columns: Epoch, Iteration, Time Elapsed (hh:mm:ss), Mini-batch Accuracy, Validation Accuracy, Mini-batch Loss, Validation Loss, and Base Learning Rate. The training progresses through 10 epochs, with the final epoch (10) showing a Mini-batch Accuracy of 90.77% and a Validation Accuracy of 90.86%. Below the table, it states 'Training finished: Max epochs completed.'

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:08	9.81%	80.07%	12.1242	2.5350	0.0010
1	20	00:00:12	88.84%	89.46%	1.4670	1.5389	0.0010
2	40	00:00:16	85.46%	85.88%	0.8975	0.8771	0.0010
3	50	00:00:17	78.74%		0.9360		0.0010
3	60	00:00:19	82.98%	80.77%	0.5648	0.5932	0.0010
4	80	00:00:22	84.92%	80.49%	0.4684	0.5326	0.0010
5	100	00:00:25	87.05%	83.60%	0.4030	0.4675	0.0010
6	120	00:00:27	89.45%	88.47%	0.3402	0.4044	0.0010
7	140	00:00:31	89.84%	88.41%	0.3609	0.3717	0.0010
8	150	00:00:32	83.62%		0.7344		0.0010
8	160	00:00:34	90.37%	90.03%	0.3029	0.3523	0.0010
9	180	00:00:37	90.84%	85.76%	0.3009	0.4122	0.0010
10	200	00:00:40	90.77%	90.86%	0.3165	0.3476	0.0010

Εικόνα 76: Training Progress – Αναλυτική μορφή παρουσίασης προόδου εκπαίδευσης

Η back-end πλευρά βασίζεται εν μέρει σε HTML για την μορφοποίηση του πεδίου (TextBox) και την μορφοποίηση του κειμένου. Η πρόοδος της εκπαίδευσης λόγω περιορισμού της συνάρτησης **trainNetwork()** [41], μπορεί να παρουσιαστεί μόνο κατά τη διακοπή της εκπαίδευσης και για να είναι εφικτή η παρουσίαση αυτής, γίνεται χρήση της συνάρτησης **evalc()** [41] το οποίο «μεταφέρει» την πρόοδο από το command window του Matlab.

Παράδειγμα:

```
CommandWindowText = evalc('current_unet = trainNetwork(training_final_imds,  
lgraph_unet, options);');
```

Παράδειγμα 7.8

6.4 Aria App: Backend – Evaluation Tab

Στην παρούσα υπό-ενότητα, θα επεξηγηθεί το κομμάτι του back-end της εφαρμογής, στη τέταρτη καρτέλα Evaluation.

The screenshot displays the 'Evaluation' tab of the Aria App. It features a navigation bar with tabs for 'Homepage', 'Updates', 'Image Processing', 'Training', 'Training-Progress', and 'Evaluation'. The main content area is divided into several sections:

- Performance Metrics:** A list of metrics for 20 images, including Sensitivity (0.94482), Specificity (0.94203), Precision (0.64397), F1 Score (0.76591), Intersection over Union (0.62063), and Area under Curve (0.94342). Below this, a summary for 'Image: 20' shows Accuracy (0.93745), Sensitivity (0.91181), Specificity (0.94009), Precision (0.61057), F1 Score (0.73139), Intersection over Union (0.57652), and Area under Curve (0.92595). A final summary for 'Total Images: 20' provides Mean Accuracy (0.93700), Mean Sensitivity (0.95457), Mean Specificity (0.93546), Mean Precision (0.57781), Mean F1 Score (0.71868), Mean Intersection over Union (0.56164), and Mean Area under Curve (0.94501).
- Image Display:** A large circular fundus image labeled 'Image: 18' with blue vessel segmentation. Below it is a smaller image labeled 'Label: 18' showing the ground truth white vessel segmentation.
- Network Evaluation Settings:** A panel on the right with a 'Select Test Image/Label Folder' button, a 'Select network to evaluate' dropdown menu set to 'Aria Network', and an 'Apply final vote (PK Engine)' dropdown menu set to 'false'. It includes 'Analyze Network' and 'Evaluate Network' buttons.
- Performance Comparison:** A table at the bottom left comparing 'Lowest' and 'Highest' performance metrics across various images.

Lowest		Highest	
Accuracy	0.9268 Image: 1	Accuracy	0.9497 Image: 15
Sensitivity	0.908 Image: 6	Sensitivity	0.983 Image: 14
Specificity	0.9221 Image: 1	Specificity	0.949 Image: 15
Precision	0.5046 Image: 8	Precision	0.644 Image: 19
F1 Score	0.6603 Image: 8	F1 Score	0.7659 Image: 19
IoU	0.4929 Image: 8	IoU	0.6206 Image: 19
AUC	0.9257 Image: 9	AUC	0.9568 Image: 10

Εικόνα 77: Evaluation – Καρτέλα αξιολόγησης δικτύου με παραδείγματα

Ξεκινώντας από το κομμάτι του Network Evaluation Settings, υπάρχουν 2 επιλογές:

1. Select network to evaluate: Για την επιλογή του δικτύου που θέλει να αξιολογήσει ο χρήστης.
2. Apply Final Vote (PK Engine): Οι τιμές που λαμβάνει η επιλογή αυτή, είναι true ή false.

Select network to evaluate:

Οι διαθέσιμες επιλογές είναι οι εξής:

- **Aria Network:** Εφόσον έχει γίνει εκπαίδευση δικτύου μέσω της εφαρμογής, η επιλογή Aria Network εμφανίζεται πρώτη.
- **browse.:** Στην οποία μπορεί να γίνει επιλογή ξεχωριστού δικτύου, τύπου .mat. Στη περίπτωση αυτή, γίνεται χρήση της συνάρτησης **uigetfile()** [41] για την επιλογή του αρχείου που περιέχει το δίκτυο που θέλει ο χρήστης να αξιολογήσει. Το ξεχωριστό δίκτυο, μπορεί να προέρχεται και εκτός εφαρμογής.

Apply Final Vote (PK Engine):

Ο μηχανισμός PK χρησιμοποιεί τα αρχικά της MD Persila Koumati, αφού η ιδέα προήλθε από το τρόπο που αξιολογεί ιατρικά δεδομένα, «ρυθμίζοντας» διάφορες συνιστώσες, για την εύρεση διαφορών / ομοιοτήτων και σύγκρισης στο γενικότερα φάσμα.

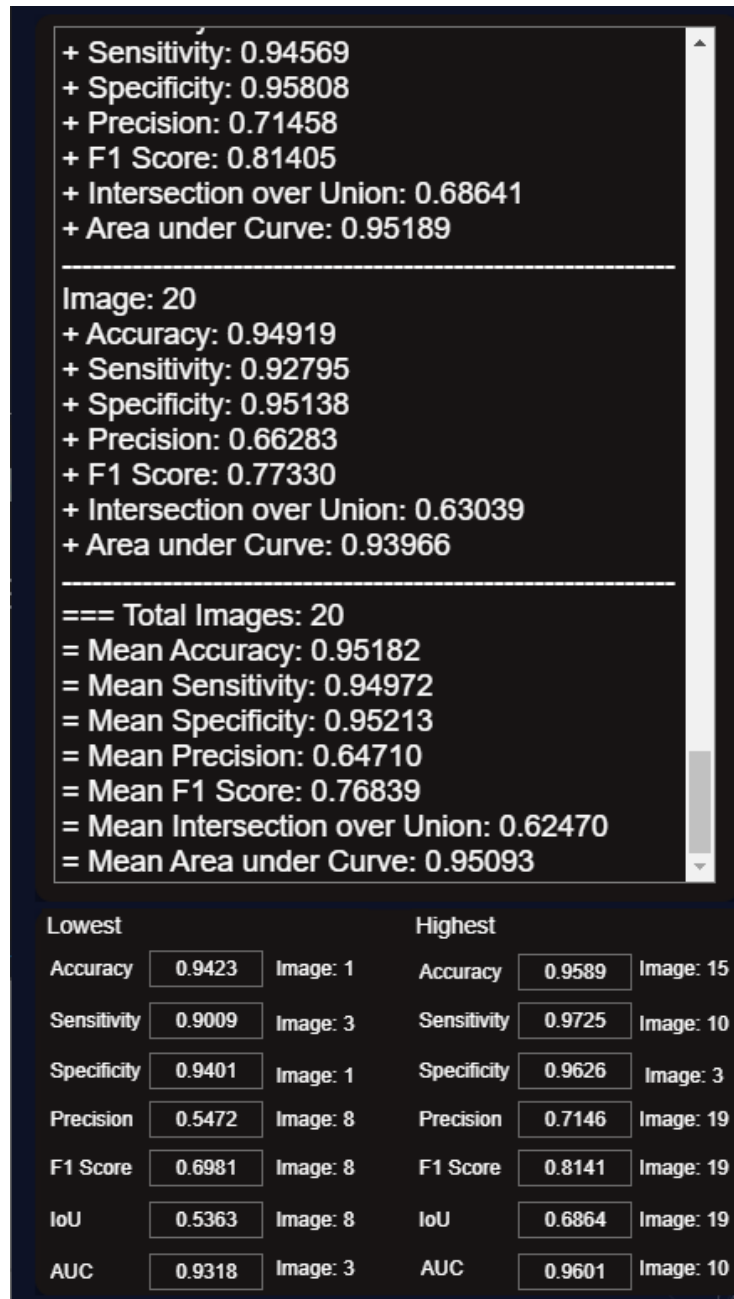
Ο μηχανισμός πλειοψηφίας, ενεργοποιείται εφόσον τεθεί η μεταβλητή στο true και αλλάζει τον τρόπο με τον οποίο παρουσιάζεται η τελική εικόνα. Αυτό γίνεται μέσω της αξιολόγησης της κάθε εικόνας 5 φορές στο σύνολο, με τον εξής τρόπο:

- 1^η αξιολόγηση: Καμία αλλαγή στην εικόνα πριν την αξιολόγηση
- 2^η αξιολόγηση: Εφαρμογή γκαουσιανού φίλτρου με τιμή $\sigma = 0.25$
- 3^η αξιολόγηση: Εφαρμογή γκαουσιανού φίλτρου με τιμή $\sigma = 0.50$
- 4^η αξιολόγηση: Εφαρμογή γκαουσιανού φίλτρου με τιμή $\sigma = 0.75$
- 5^η αξιολόγηση: Εφαρμογή γκαουσιανού φίλτρου με τιμή $\sigma = 1.00$

Όπου η ίδια εικόνα υπάρχει 5 φορές και αξιολογείται η κάθε μια από αυτές με τον τρόπο που εξηγείται παραπάνω. Εφόσον αξιολογηθεί η ίδια εικόνα 5 ξεχωριστές φορές, τότε δημιουργούνται 5 ξεχωριστές μάσκες της ίδιας εικόνας, με διαφορετικές τιμές σ του γκαουσιανού φίλτρου. Τελικώς, η μάσκα που εξάγεται προκύπτει από τη πλειοψηφία των ετικετών για κάθε ξεχωριστό pixel. Π.χ.: Αν για το pixel στην θέση [1,1] οι 3 από τις 5 μάσκες έχουν την ετικέτα vessel, τότε το pixel στη θέση [1,1] για τη τελική μάσκα, είναι vessel.

Συνεχίζοντας την υπό-ενότητα αυτή με την αριστερή πλευρά της καρτέλας όπου γίνεται η παρουσίαση των αποτελεσμάτων αξιολόγησης της κάθε εικόνας και ταυτόχρονα προβάλλεται η ελάχιστη και η μεγίστη

τιμή που καταγράφηκε από τις αξιολογήσεις, σε συνδυασμό με το νούμερο της εικόνα για την καταγραφή αυτή. Παρακάτω, θα αναλυθεί ο τρόπος με τον οποίο υπολογίζονται οι δείκτες αξιολόγησης – απόδοσης (evaluation/performance metrics) κατά την αξιολόγηση της απόδοσης του δικτύου για την κάθε εικόνα ξεχωριστά.



Εικόνα 78: Evaluation – Πλαίσιο προβολής αριθμητικών αποτελεσμάτων της απόδοσης του δικτύου

Lowest, Mean, Highest | Metrics:

Για τις μετρήσεις ελαχίστου και μεγίστου, έχουν τεθεί αρχικά κατώφλια για τον κάθε δείκτη, έτσι ώστε η πρώτη εικόνα που θα αξιολογηθεί να έχει μικρότερη τιμή από το ελάχιστο κατώφλι (9999) και μεγαλύτερη τιμή από το μέγιστο κατώφλι (0), ενώ ταυτόχρονα καταγράφεται και ο αριθμός της εικόνας για την κάθε μέτρηση ελαχίστου και μεγίστου, όπως φαίνεται στην παραπάνω εικόνα. Όλες οι τιμές, αποθηκεύονται στις σχετικές μεταβλητές, δημιουργώντας έναν πίνακα με όλες τις μετρήσεις. Για τον υπολογισμό της μέσης τιμής όπως φαίνεται στο TextBox, προστίθενται όλες οι τιμές των παραπάνω πινάκων και ύστερα διαιρείται τελική τιμή με το αριθμό των τιμών που υπάρχουν στον πίνακα (αριθμός των εικόνων που αξιολογήθηκαν και αποθηκεύτηκαν) μέσω της συνάρτησης **mean()** [41].

Παράδειγμα:

```
accuracy = [0.96, 0.85, 0.86, 0.92, 0.99];  
  
mean_accuracy = mean(mean(accuracy));  
  
>> mean_accuracy = 0.9160
```

Παράδειγμα 7.9

ενώ για την συγκράτηση του αριθμού της εικόνας όταν εντοπίζεται νέο ελάχιστο ή μέγιστο κατώφλι, αποθηκεύεται ως 2^η εγγραφή στην ίδια μεταβλητή που κρατάει το κατώφλι.

Παράδειγμα:

```
NumberOfImages = numel(evaluation_imageDB.Files);  
  
for i=1:NumberOfImages  
if accuracy(i) < min_accuracy(1)  
min_accuracy(1) = accuracy(i);  
min_accuracy(2) = i;  
end  
end
```

Παράδειγμα 8.1

Κώδικας υπολογισμού των δεικτών απόδοσης – αξιολόγησης

Για τον υπολογισμό των δεικτών χρησιμοποιείται εξατομικευμένη συνάρτηση με το όνομα **evaluateSegmentation()** με εισόδους την μάσκα αληθείας **groundTruth** και την μάσκα **predictedSegmentation** που δημιουργήθηκε από το δίκτυο.

Ο υπολογισμός των TP, TN, FP, FN γίνεται ως εξής:

% True Positive (TP): Vessel pixels correctly classified as vessel pixels

TP = sum(groundTruth(:) & predictedSegmentation(:));

Αληθώς Θετικό (TP): Τα εικονοστοιχεία που ανήκουν σε αγγείο και ταξινομούνται σωστά ως εικονοστοιχεία αγγείου. Το TP υπολογίζεται μέσω της συνάρτησης **sum()** [41] στα στοιχεία του πίνακα **groundTruth**, που αντιστοιχούν σε αγγείο, και του πίνακα **predictedSegmentation**, που δείχνουν ως εικονοστοιχεία αγγείου.

% True Negative (TN): Non-vessel pixels correctly classified as non-vessel pixels

TN = sum(~groundTruth(:) & ~predictedSegmentation(:));

Αληθώς Αρνητικό (TN): Τα εικονοστοιχεία που δεν ανήκουν σε αγγείο και ταξινομούνται σωστά ως εικονοστοιχεία που δεν ανήκουν σε αγγείο. Το TN υπολογίζεται μέσω της συνάρτησης **sum()** [41] στα στοιχεία του πίνακα **groundTruth**, που δεν αντιστοιχούν σε αγγείο, και του πίνακα **predictedSegmentation**, που δείχνουν ως στοιχεία που δεν ανήκουν σε αγγείο.

% False Positive (FP): Non-vessel pixels incorrectly classified as vessel pixels

FP = sum(~groundTruth(:) & predictedSegmentation(:));

Ψευδώς Θετικό (FP): Τα εικονοστοιχεία που δεν ανήκουν σε αγγείο και ταξινομούνται εσφαλμένα ως εικονοστοιχεία αγγείου. Το FP υπολογίζεται μέσω της συνάρτησης **sum()** [41] στα στοιχεία του πίνακα **groundTruth**, που δεν αντιστοιχούν σε αγγείο, και του πίνακα **predictedSegmentation**, που δείχνουν ως εικονοστοιχεία αγγείου.

% False Negative (FN): Vessel pixels incorrectly classified as non-vessel pixels

FN = sum(groundTruth(:) & ~predictedSegmentation(:));

Ψευδώς Αρνητικό (FN): Τα εικονοστοιχεία που ανήκουν σε αγγείο και ταξινομούνται εσφαλμένα ως εικονοστοιχεία που δεν ανήκουν σε αγγείο. Το FN υπολογίζεται μέσω της συνάρτησης **sum()** [41] στα στοιχεία του πίνακα **groundTruth**, που αντιστοιχούν σε αγγείο, και του πίνακα **predictedSegmentation**, που δείχνουν ως στοιχεία που δεν ανήκουν σε αγγείο.

Υστερα, βάσει των παραπάνω παραμέτρων TP, TN, FP, FN, υπολογίζονται οι δείκτες:

Accuracy:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad \text{Τύπος 8.2}$$

Sensitivity:

$$sensitivity = \frac{tp}{tp + fn} \quad \text{Τύπος 8.3}$$

Specificity:

$$specificity = \frac{tn}{tn + fp} \quad \text{Τύπος 8.4}$$

Precision:

$$precision = \frac{tp}{tp + fp} \quad \text{Τύπος 8.5}$$

F1 Score:

$$f1Score = \frac{2 * (precision * sensitivity)}{precision + sensitivity} \quad \text{Τύπος 8.6}$$

Intersection over Union:

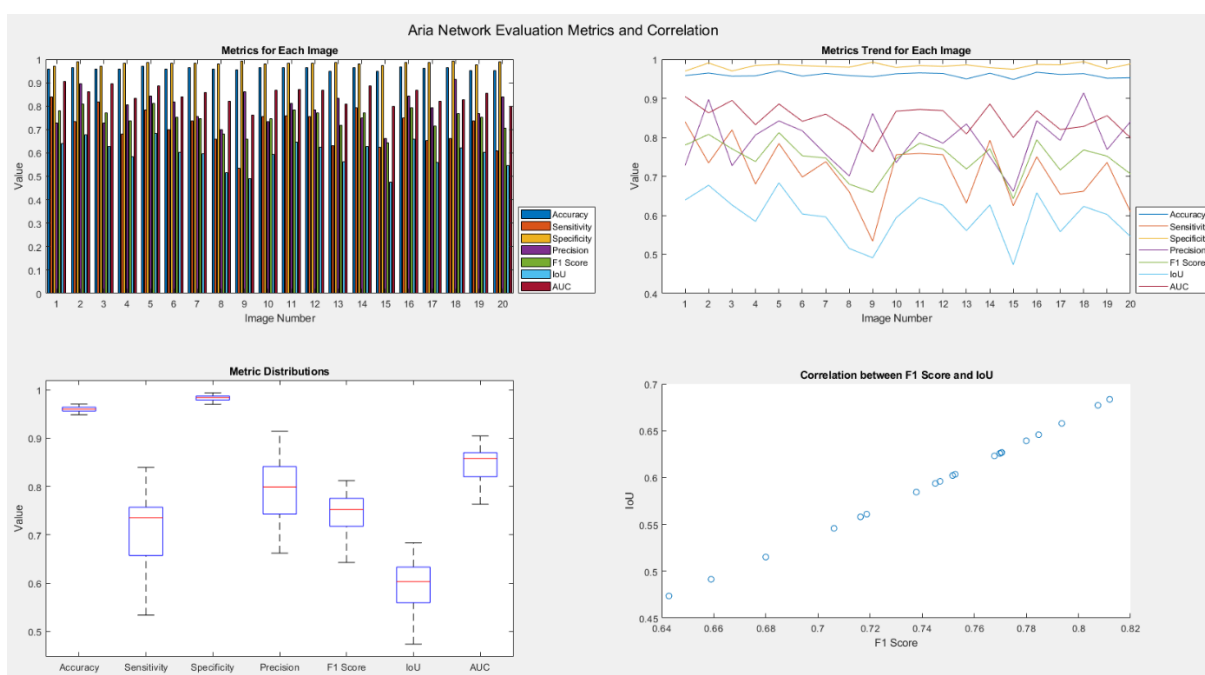
$$IoU = \frac{TP}{(TP + FN + FP)} \quad \text{Τύπος 8.7}$$

Στη περίπτωση του δείκτη AUC (Area Under the Curve) χρησιμοποιείται η συνάρτηση **perfcurve()** [41] ως εξής:

```
% Convert to scores (floating-point values)
scores = double(predictedSegmentation(:));
```

```
% ROC Curve (TPR, FPR) → AUC
[ tpr, fpr, ~, auc ] = perfcurve(groundTruth(:), scores, true);
```

Κατά το τέλος του υπολογισμού όλων των παραπάνω, η εφαρμογή παρουσιάζει τα εξής γραφήματα σε ξεχωριστό παράθυρο ως μέρος της περιγραφικής στατιστικής, οπτικοποιώντας τα αποτελέσματα:



Εικόνα 79: Evaluation – Ξεχωριστό παράθυρο οπτικοποίησης αποτελεσμάτων απόδοσης

Διάγραμμα Ομαδοποιημένων Ράβδων (Επάνω Αριστερά):

- Ο άξονας x αναπαριστά τους αριθμούς των εικόνων.
- Ο άξονας y αναπαριστά τις τιμές διαφορετικών δεικτών αξιολόγησης.
- Κάθε εικόνα αναπαρίσταται από ένα σύνολο ραβδίων, όπου κάθε ράβδος αντιστοιχεί σε ένα συγκεκριμένο δείκτη.
- Η λεζάντα υποδεικνύει ποιο χρώμα αντιστοιχεί σε κάθε δείκτη.

Γράφημα Περιοδικών Τάσεων Μετρικών (Επάνω Δεξιά):

- Ο άξονας x αναπαριστά τους αριθμούς των εικόνων.
- Ο άξονας y αναπαριστά τις τιμές διαφορετικών δεικτών αξιολόγησης.
- Κάθε γραμμή αντιπροσωπεύει την τάση ενός συγκεκριμένου δείκτη στο εύρος των εικόνων.
- Η λεζάντα υποδεικνύει ποιο χρώμα αντιστοιχεί σε κάθε δείκτη.

Θηκόγραμμα Κατανομής των Δεικτών (Κάτω Αριστερά):

- Κάθε κουτί αναπαριστά τη κατανομή των τιμών για ένα συγκεκριμένο δείκτη.
- Ο άξονας x δείχνει τα ονόματα των δεικτών.
- Ο άξονας y αναπαριστά τις τιμές των δεικτών.
- Το θηκόγραμμα παρέχει μια εικόνα της διασποράς και των κεντρικών τάσεων των τιμών του κάθε δείκτη.

Διάγραμμα Διασποράς για τη συσχέτιση δεικτών (Κάτω Δεξιά):

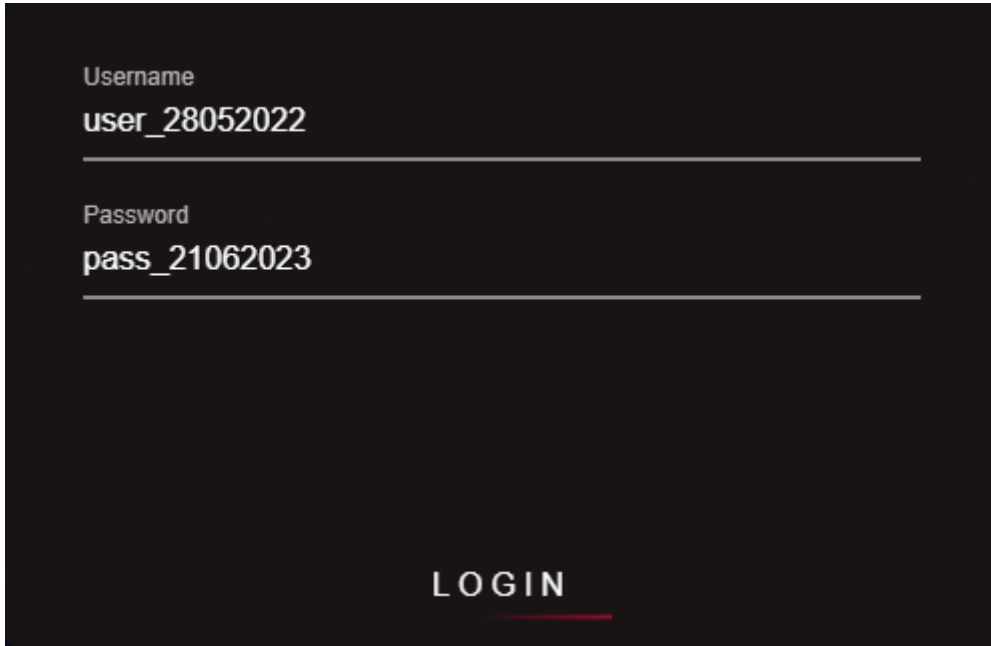
- Ο άξονας x αναπαριστά τις τιμές του F1 Score.
- Ο άξονας y αναπαριστά τις τιμές του IoU.
- Κάθε σημείο στο διάγραμμα διασποράς αντιπροσωπεύει την απόδοση μιας εικόνας όσον αφορά το F1 Score και το δείκτη IoU.
- Το διάγραμμα βοηθά να οπτικοποιηθεί η σχέση μεταξύ αυτών των δύο δεικτών.

Κεφάλαιο 7: Αποτελέσματα Aria App

Σε αυτό το κεφάλαιο θα γίνει η συνολική παρουσίαση της εφαρμογής, όπως θα παρουσιαζόταν σε πραγματικές συνθήκες.

7.1 Πειραματικό Στάδιο – Aria App in Action (AAA)

Logging In:

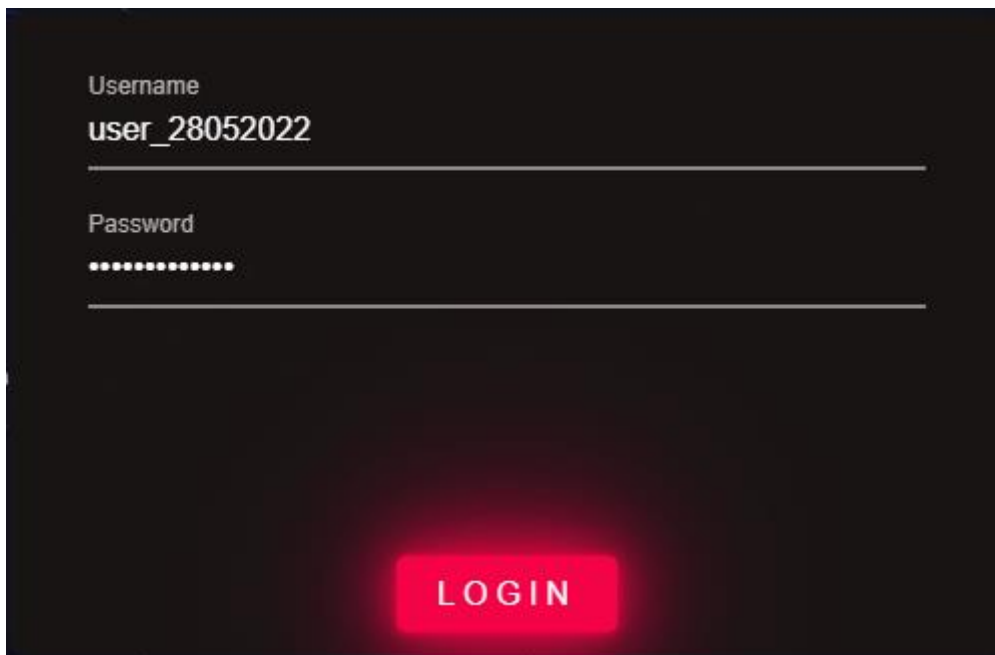


Username
user_28052022

Password
pass_21062023

LOGIN

Εικόνα 80: Homepage – Φόρμα σύνδεσης χρήστη

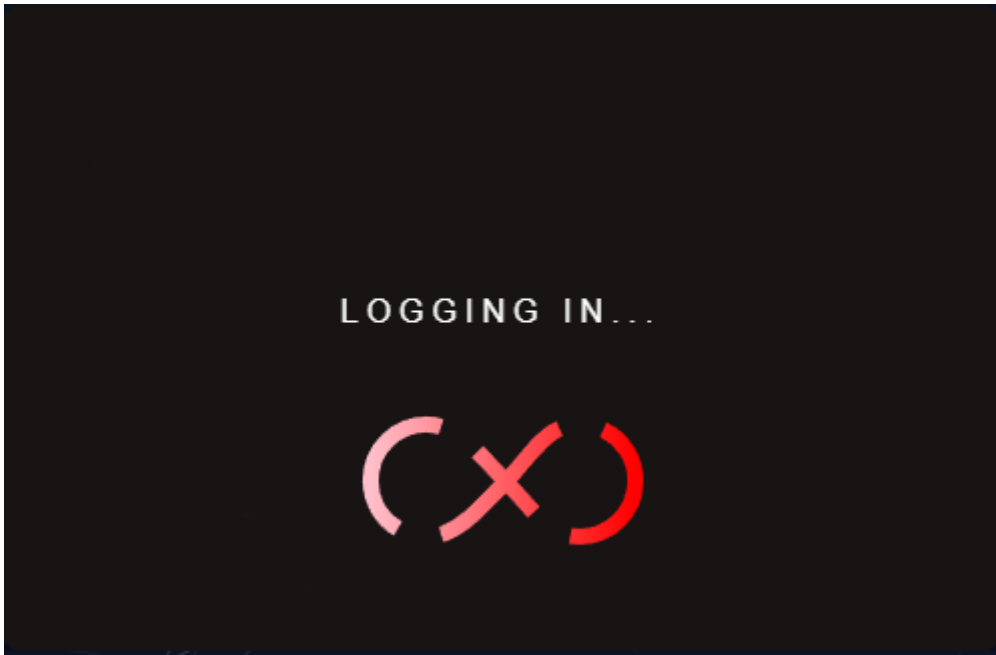


Username
user_28052022

Password
.....

LOGIN

Εικόνα 81: Homepage – Εκκίνηση επαλήθευσης των διαπιστευτηρίων του χρήστη (login)



Εικόνα 82: Homepage – Στάδιο επαλήθευσης (loading)

Μόλις η σύνδεση στο λογαριασμό είναι επιτυχής, η καρτέλα Login κλείνει και ανοίγει η καρτέλα Updates.

Updates

```
// Future releases, features and updates

=== Beta release v0.2 (01/01/1901)

- Include new app name
- Update the UI/UX of the app, include HTML, CSS, Javascript to dress panels, windows, labels, buttons, textboxes
- Change the colour palette to warmer choices for lower eye strain and combine it with a Dark theme in future releases

-----

// Change Log

=== Beta Release v0.1 (01/01/1901)

- Fixed a bug where loading non RGB images into ImageBox would cause the app to quit unexpectedly
- Fixed a bug where leaving TextBoxes empty wouldn't get them filled in automatically before proceeding to next step in the training tab
- Fixed a bug where Lowest_AUC would show the Highest_Accuracy instead

=== Alpha Release (01/01/1901)

- Develop first instances of the app
- Narrow down the number of tabs, features and the target of this app// Future releases, features and updates

-----

=== Licenses

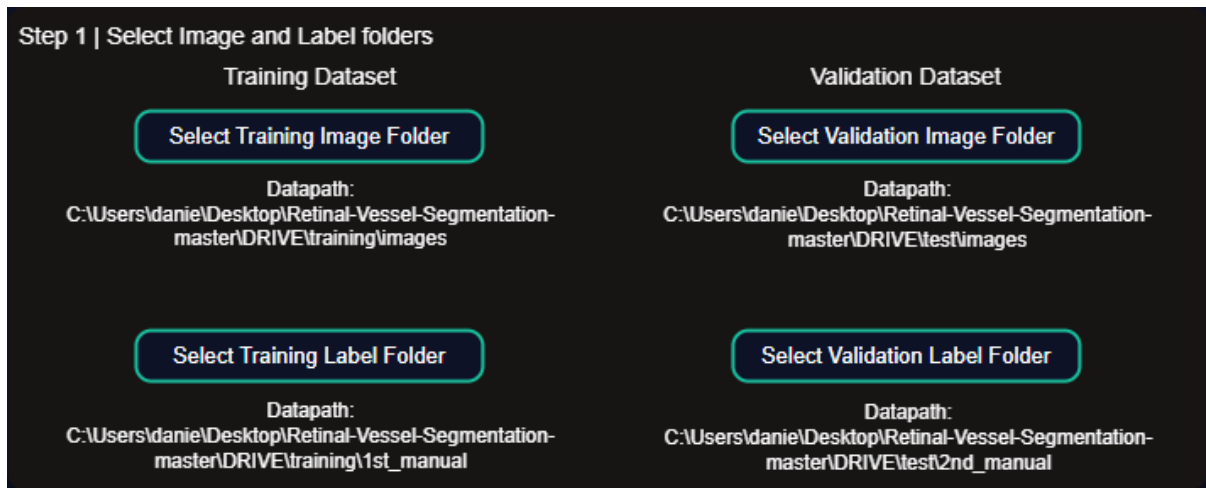
Developer: Dr. Vassilios G. ...
```

Εικόνα 83: Updates – Καρτέλα νέων ενημερώσεων

Εφόσον υπάρχει κάποια νέα ενημέρωση, αυτή θα φανεί στην καρτέλα Updates.

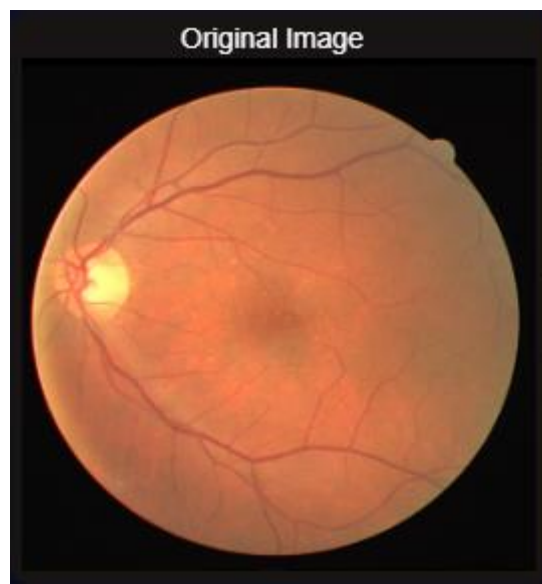
Image Processing

Βήμα 1: Επιλέγουμε τους φάκελους των εικόνων που θα χρησιμοποιηθούν στην εκπαίδευση του δικτύου:



Εικόνα 84: Image Processing - Βήμα 1 της καρτέλας προ-επεξεργασίας

Τους όποιους μόλις τους επιλέξουμε, θα παρουσιαστεί αυτομάτως ένα δείγμα εικόνας εντός της εφαρμογής Aria:



Εικόνα 85: Image Processing – Προβολή δείγματος από το φάκελο εικόνων προς εκπαίδευση

Εφόσον επιβεβαιώσουμε ότι οι φάκελοι και οι εικόνες είναι αυτές που θέλουμε να δουλέψουμε τόσο στο κομμάτι της προ-επεξεργασίας αλλά και μετέπειτα στο στάδιο της εκπαίδευσης του δικτύου, τότε μπορούμε να προχωρήσουμε στο βήμα 2.

Βήμα 2: Στο σημείο αυτό επιλέγουμε αρχικά μόνο την επιλογή “Grayscale” για να αποκτήσουμε την πιο βασική μορφή της εικόνας (από RGB σε Γκρι) και σε δεύτερο στάδιο να προβάλλουμε 2 δείγματα εικόνων προτού προβούμε σε περισσότερες αλλαγές – τεχνικές επεξεργασίας:

Step 2 | Select Processing Parameters

- Grayscale
- Normalization
- Gauss Filtering
Sigma σ
- Median Filtering
Matrix
- CLAHE
ClipLimit
NumTile
NBins
- Illumination Adjustment
Gamma

Original Image

Processed Image

Image_1

View Sample

Εικόνα 86: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Προβολή του δεύτερου δείγματος εικόνας από RGB σε γκρι

Step 2 | Select Processing Parameters

- Grayscale
- Normalization
- Gauss Filtering
Sigma σ
- Median Filtering
Matrix
- CLAHE
ClipLimit
NumTile
NBins
- Illumination Adjustment
Gamma

Original Image

Processed Image

Image_ 2

View Sample

Εικόνα 87: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Υστερα επιλέγουμε τις παρακάτω επιλογές βάσει του σκοπού – στόχου της εργασίας και προβάλλουμε ένα δείγμα (Image_3):

The screenshot displays a software interface for image processing. At the top, a panel titled "Step 2 | Select Processing Parameters" contains several options:

- Grayscale
- Normalization
- Gauss Filtering
Sigma σ : 0.50
- Median Filtering
Matrix: Default: 3,3
- CLAHE
ClipLimit: 0.02
NumTile: 16,16
NBins: 128
- Illumination Adjustment
Gamma: 0.95

Below the parameter panel, two circular images are shown side-by-side:

- Original Image:** A color fundus image of a retina, showing a warm orange-red color palette.
- Processed Image:** The same fundus image after processing, rendered in grayscale to highlight the vascular network.

At the bottom of the interface, there are two buttons: "Image_ 3" with a dropdown arrow and "View Sample".

Εικόνα 88: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Προβολή διαφορετικού δείγματος με τις ίδιες επιλογές (Image_4):

The screenshot displays a software interface for image processing. At the top, a panel titled "Step 2 | Select Processing Parameters" contains several settings:

- Grayscale
- Normalization
- Gauss Filtering
 - Sigma σ : 0.50
- Median Filtering
 - Matrix: Default: 3,3
- CLAHE
 - ClipLimit: 0.02
 - NumTile: 16,16
 - NBins: 128
- Illumination Adjustment
 - Gamma: 0.95

Below the parameter panel, two circular images are shown side-by-side:

- Original Image:** A color fundus photograph of a retina, showing a network of blood vessels and a bright optic disc on the right side.
- Processed Image:** A grayscale version of the same fundus photograph, where the vessels and optic disc are clearly visible against a dark background.

At the bottom of the interface, there is a dropdown menu showing "Image_ 4" and a "View Sample" button.

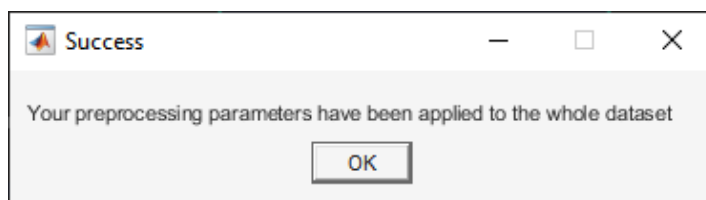
Εικόνα 89: Image Processing – Προβολή τεχνικών προ-επεξεργασίας και οπτικοποίηση αποτελεσμάτων

Βήμα 3: Εφόσον είμαστε ευχαριστημένοι με το τελικό αποτέλεσμα των φωτογραφιών βάσει των παραπάνω επιλογών που κάναμε, μπορούμε να εφαρμόσουμε τις επιλογές αυτές σε όλη τη βάση των εικόνων (φωτογραφικών) – dataset:



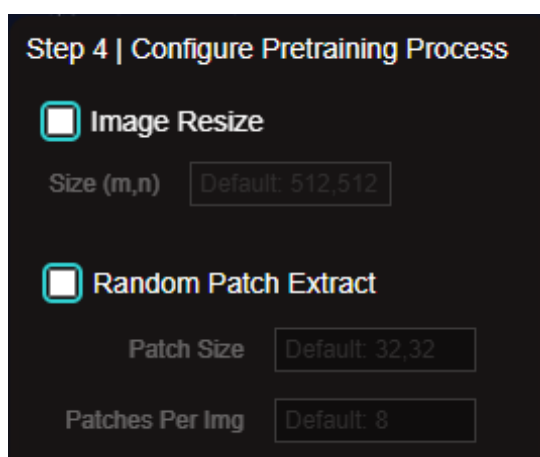
Εικόνα 90: Image Processing – Εφαρμογή όλων των επιλεγμένων τεχνικών προ-επεξεργασίας στο dataset

όπου και λαμβάνουμε μια ειδοποίηση – ενημέρωση για την επιτυχή εφαρμογή των παραπάνω τεχνικών – μεθόδων (επιλογών) σε όλο το dataset που έχει επιλεγθεί:



Εικόνα 91: Image Processing – Εμφάνιση μηνύματος επιτυχούς εφαρμογής τεχνικών στο dataset

Βήμα 4: Μπορούμε πλέον να προχωρήσουμε στο τελευταίο βήμα (4) της καρτέλας Image Processing, όπου επιλέγουμε αν θέλουμε να αλλάξουμε τις διαστάσεις της κάθε εικόνας (Image Resizing) ή/και να παράγουμε τμήματα «μπαλώματα» από τη κάθε φωτογραφία, επιλέγοντας τις διαστάσεις του τμήματος και τον αριθμό των τμημάτων που θα εξαχθεί από την κάθε φωτογραφία:



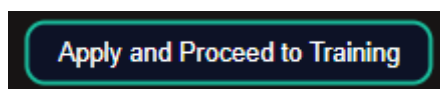
Εικόνα 92: Image Processing – Επιλογής διαμόρφωσης διαστάσεων και εξαγωγής τμημάτων εικόνας

Για το συγκεκριμένο παράδειγμα, θα επιλέξουμε Image Resizing: 512, 512 και θα ενεργοποιήσουμε την επιλογή της εξαγωγής τμημάτων της εικόνας με διαστάσεις 64,64 και 80 τυχαία τμήματα ανά εικόνα και προβάλλουμε τα αποτελέσματα:



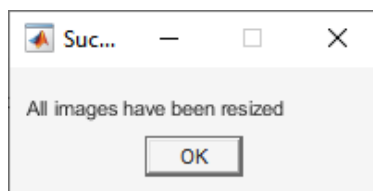
Εικόνα 93: Image Processing – Επιλογή resizing / patching και εισαγωγή τιμών στα πεδία

και εφαρμόζουμε τις νέες τιμές για αυτές τις ρυθμίσεις:

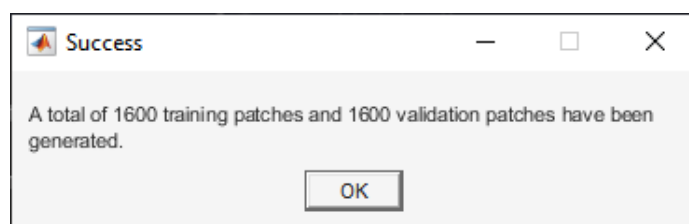


Εικόνα 94: Image Processing – Εφαρμογή resizing / patching και μετάβαση στη καρτέλα εκπαίδευσης

όπου και λαμβάνουμε μια ειδοποίηση – ενημέρωση για την επιτυχής εφαρμογή των παραπάνω ρυθμίσεων (επιλογών) σε όλο το dataset που έχει επιλεχθεί, μαζί με τον αριθμό των τμημάτων «μπαλωμάτων» που έχουν εξαχθεί:



Εικόνα 95: Image Processing – Εμφάνιση ειδοποίησης επιτυχούς αλλαγής διαστάσεων (resizing)



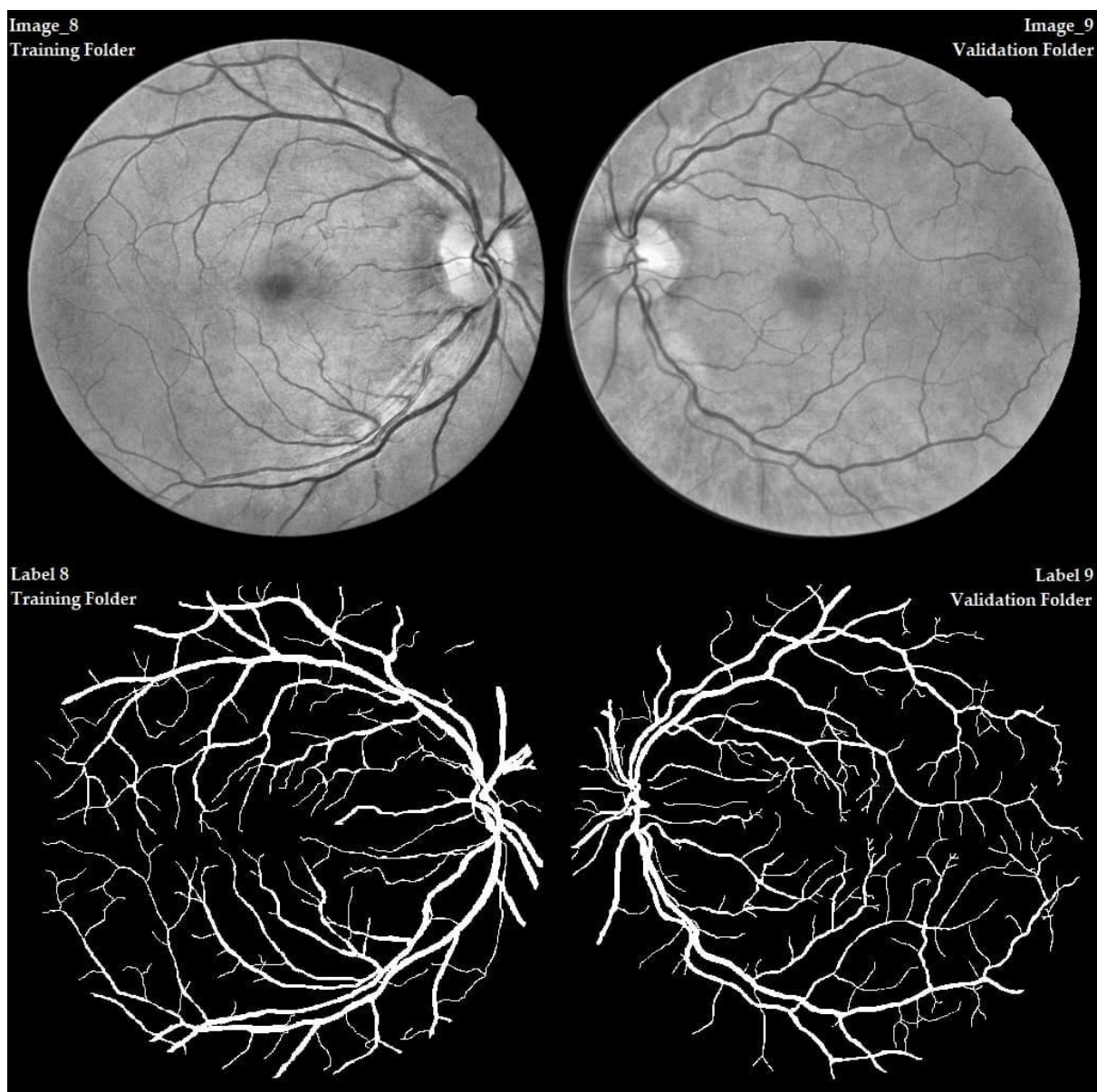
Εικόνα 96: Image Processing – Εμφάνιση ειδοποίησης επιτυχούς εξαγωγής patches από τις εικόνες

Οι τελικές εικόνες (φωτογραφίες) μπορούν να βρεθούν στους παρακάτω φακέλους:

DL_Temporary_Folder_Demo_1	File folder
1_demo_training_final_imds	File folder
1_demo_training_final_plimds	File folder
1_demo_validation_final_imds	File folder
1_demo_validation_final_plimds	File folder

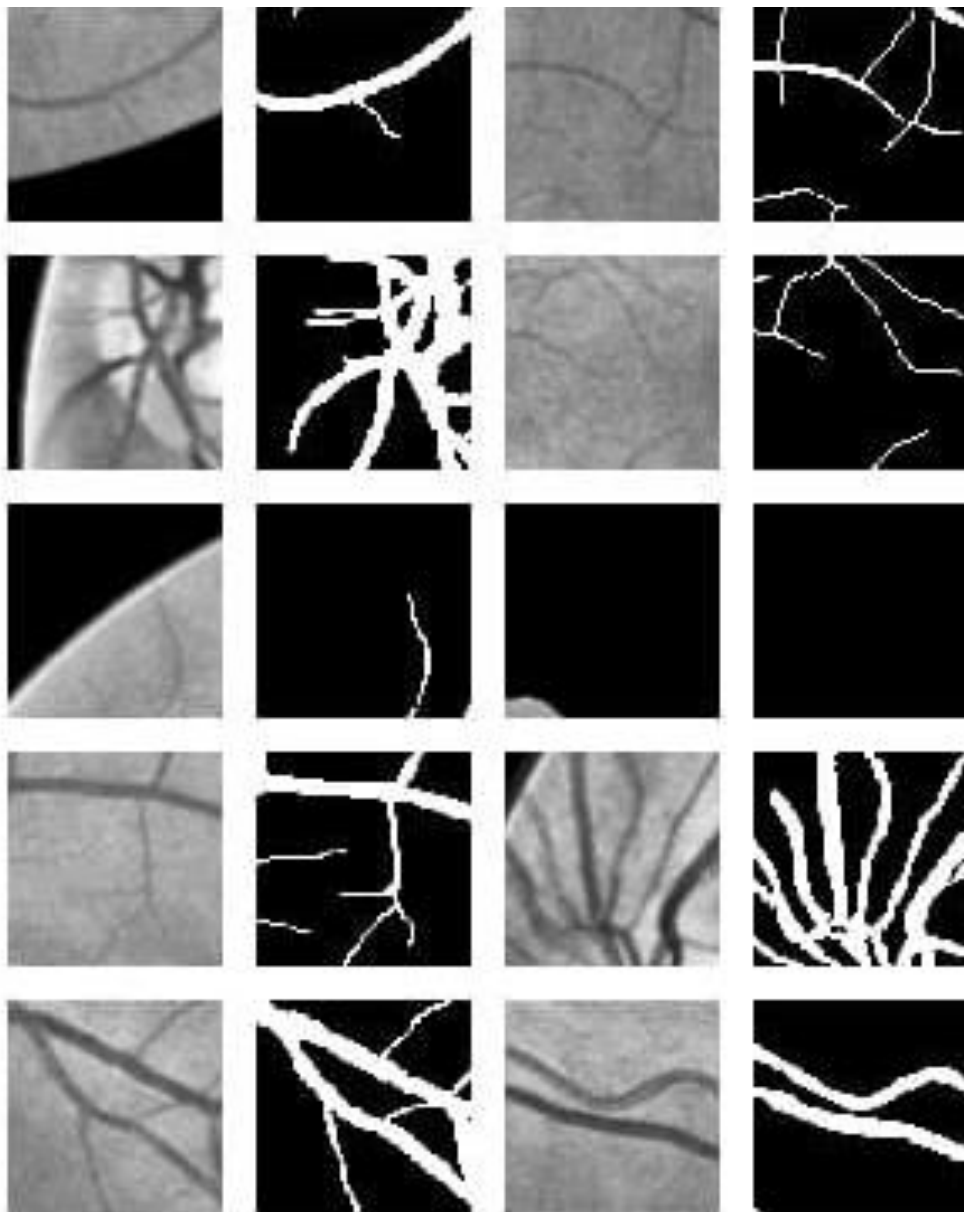
Εικόνα 97: Windows OS – Temporary Directory – Φάκελοι επεξεργασμένων εικόνων και patches

Στο φάκελο **DL_Temporary_Folder_Demo_1** βρίσκονται οι εικόνες 512x512:



Εικόνα 98: Windows OS – Temporary Directory – Προβολή επεξεργασμένων εικόνων

Στους φακέλους `1_demo_training_final_*` και `1_demo_validation_final_*` βρίσκονται τα τμήματα των εικόνων, με τις δύο πρώτες στήλες να ανήκουν στο training και τις 2 τελευταίες στο validation. Όλα τα τμήματα (εικόνες) είναι σε διαστάσεις 64x64, όπως είχε επιλεχθεί νωρίτερα:



Εικόνα 99: Windows OS – Temporary Directory – Προβολή δειγμάτων εξαγωγής random patches

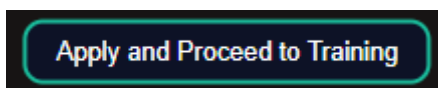
Και φτάνοντας σε αυτό το σημείο, πλέον έχουμε το νέο, επεξεργασμένο dataset τόσο σε πλήρη εικόνα μεγέθους 512x512 και τα τμήματα που εξάχθηκαν από τις εικόνες αυτές, σε μέγεθος 64x64.

Training / Validation Dataset: DRIVE | 20 images per dataset

Training / Validation Patches Dataset: 80 patches * 20 images per dataset

Training:

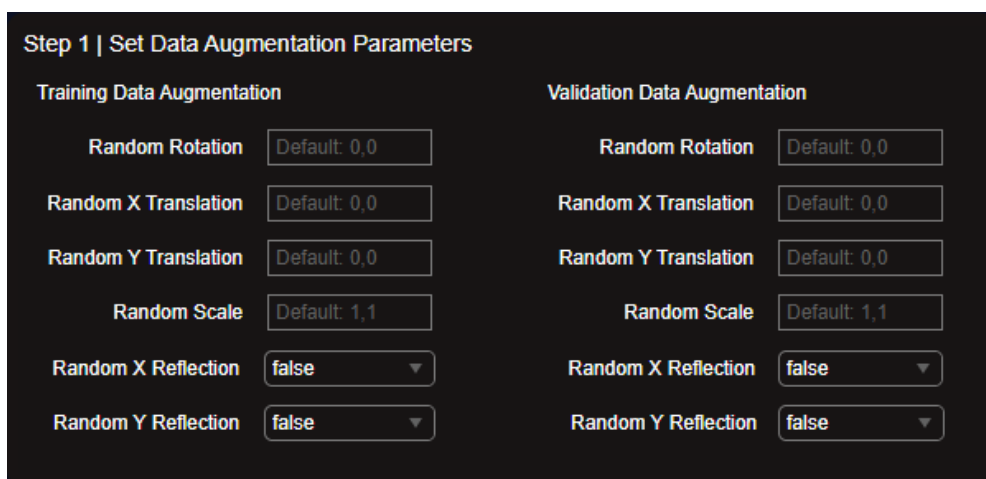
Στη καρτέλα Training φτάνουμε μόλις πατήσουμε το κουμπί **Apply and Proceed to Training** στο βήμα 4 της προηγούμενης καρτέλας, Image Processing:



Εικόνα 100: Image Processing – Εφαρμογή όλων των τεχνικών επεξεργασίας και μετάβαση στην εκπαίδευση

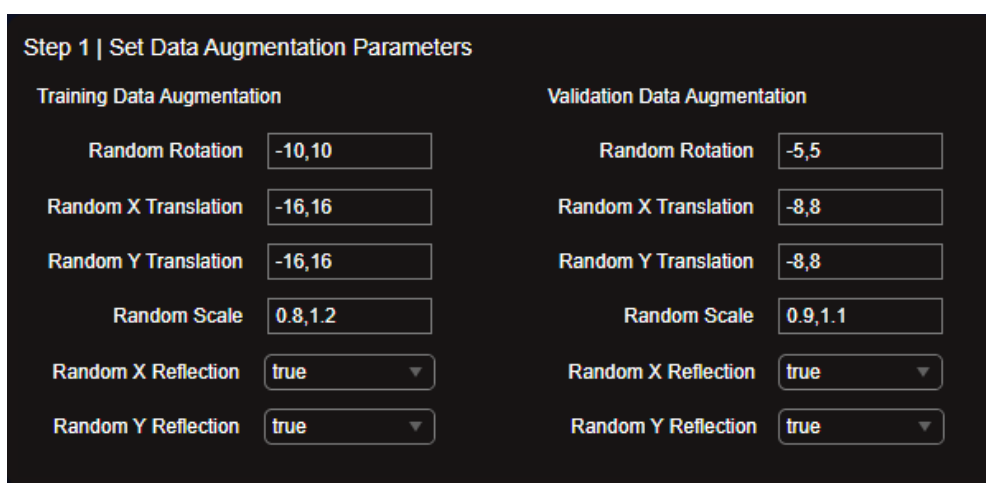
αφότου παραχθούν οι εικόνες (εφαρμογή των νέων επιλογών, παραλαβή των μηνυμάτων επιτυχής εφαρμογής των επιλογών), καταλήγουμε στα βήματα της καρτέλας Training:

Βήμα 1: Θέτουμε τις παραμέτρους για το data augmentation (επαύξηση δεδομένων), σαν έναν τρόπο αποφυγής του δικτύου από το overfitting



Step 1 Set Data Augmentation Parameters	
Training Data Augmentation	Validation Data Augmentation
Random Rotation	Random Rotation
Default: 0,0	Default: 0,0
Random X Translation	Random X Translation
Default: 0,0	Default: 0,0
Random Y Translation	Random Y Translation
Default: 0,0	Default: 0,0
Random Scale	Random Scale
Default: 1,1	Default: 1,1
Random X Reflection	Random X Reflection
false	false
Random Y Reflection	Random Y Reflection
false	false

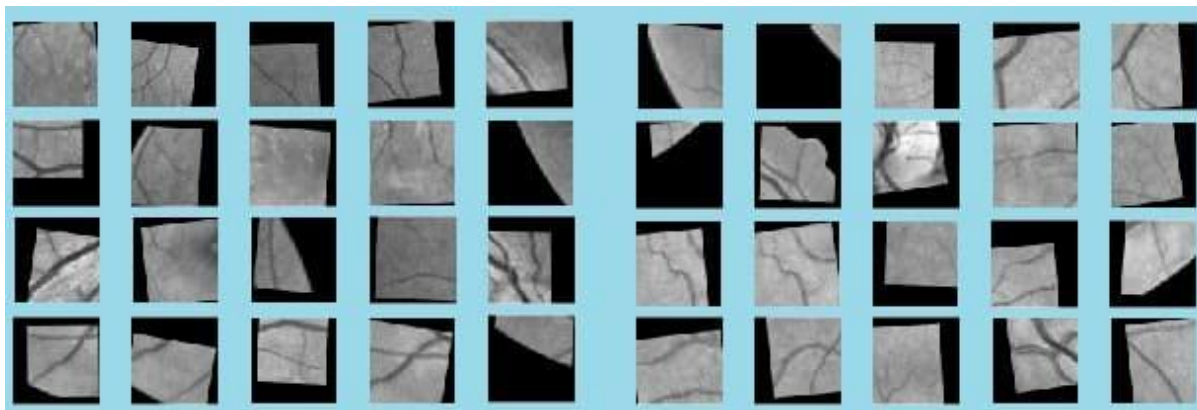
Εικόνα 101: Training – Παράμετροι Data Augmentation



Step 1 Set Data Augmentation Parameters	
Training Data Augmentation	Validation Data Augmentation
Random Rotation	Random Rotation
-10,10	-5,5
Random X Translation	Random X Translation
-16,16	-8,8
Random Y Translation	Random Y Translation
-16,16	-8,8
Random Scale	Random Scale
0.8,1.2	0.9,1.1
Random X Reflection	Random X Reflection
true	true
Random Y Reflection	Random Y Reflection
true	true

Εικόνα 102: Training - Παράδειγμα τιμών στις παραμέτρους του Data Augmentation

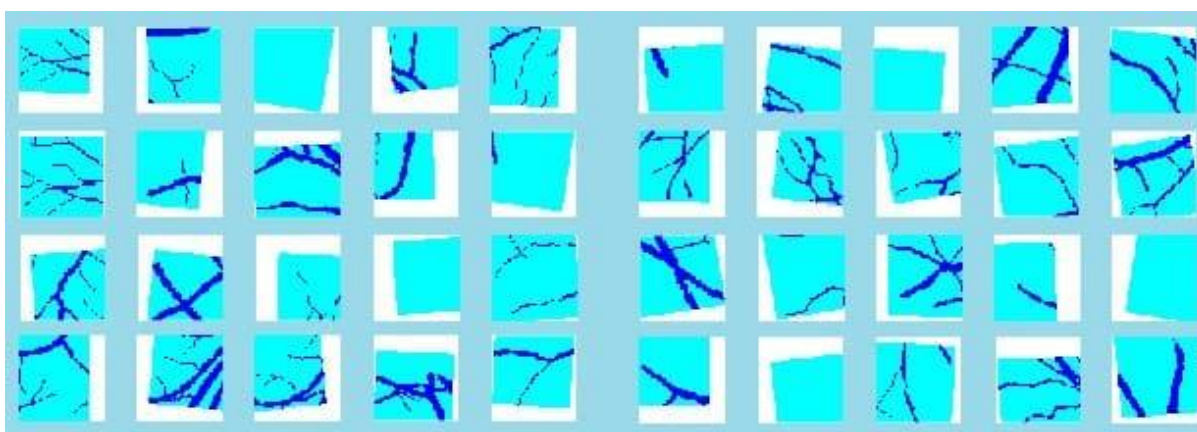
με τις οποίες τιμές που έχουν συμπληρωθεί παραπάνω για κάθε παράμετρο του data augmentation, παρατηρούμε ότι έχουν γίνει οι εξής αλλαγές στις εικόνες:



Εικόνα 103: Training – Παράδειγμα εφαρμογής data augmentation στα patches

όπου στα αριστερά οι 20 τυχαίες εικόνες που απεικονίζονται είναι από το training patches dataset και στα δεξιά οι τυχαίες 20 εικόνες που απεικονίζονται είναι από το validation patches dataset.

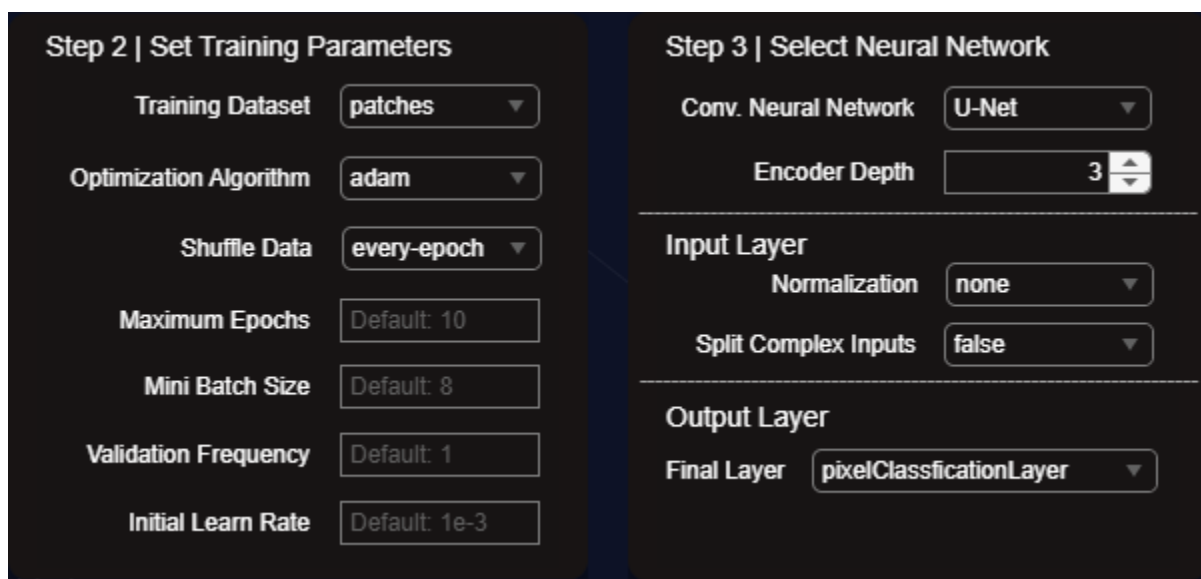
Αντίστοιχα, εφαρμόζονται οι αντίστοιχες οι τιμές των παραμέτρων του data augmentation και στις μάσκες των εικόνων αυτών και παρατηρούμε ότι έχουν οι εξής αλλαγές στις μάσκες:



Εικόνα 104: Training – Παράδειγμα εφαρμογής data augmentation στα αντίστοιχα labels των patches

όπου στα αριστερά οι 20 τυχαίες μάσκες που απεικονίζονται είναι από το training patches dataset και στα δεξιά οι τυχαίες 20 μάσκες που απεικονίζονται είναι από το validation patches dataset.

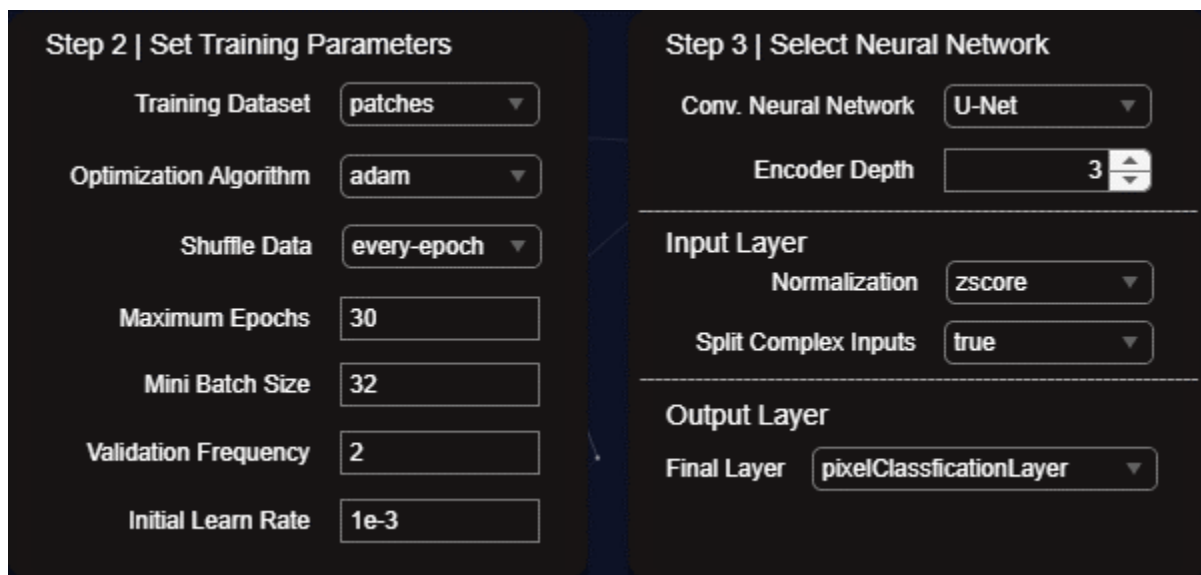
Βήμα 2 & 3: Στο πλαίσιο αυτό θέτουμε τις τελευταίες παραμέτρους για την εκπαίδευση του δικτύου.



Step 2 Set Training Parameters		Step 3 Select Neural Network	
Training Dataset	patches	Conv. Neural Network	U-Net
Optimization Algorithm	adam	Encoder Depth	3
Shuffle Data	every-epoch	Input Layer Normalization	none
Maximum Epochs	Default: 10	Split Complex Inputs	false
Mini Batch Size	Default: 8	Output Layer Final Layer	pixelClassificationLayer
Validation Frequency	Default: 1		
Initial Learn Rate	Default: 1e-3		

Εικόνα 105: Training – Βήμα 2 και βήμα 3 της καρτέλας Training

Για την παρούσα εργασία και συγκεκριμένα για το παράδειγμα – πειραματικό στάδιο, τίθενται οι παρακάτω τιμές:

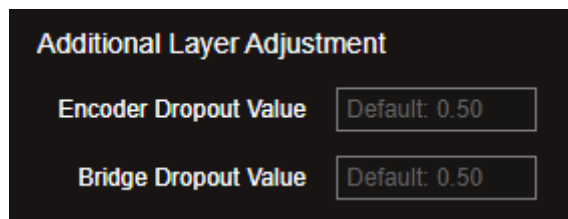


Step 2 Set Training Parameters		Step 3 Select Neural Network	
Training Dataset	patches	Conv. Neural Network	U-Net
Optimization Algorithm	adam	Encoder Depth	3
Shuffle Data	every-epoch	Input Layer Normalization	zscore
Maximum Epochs	30	Split Complex Inputs	true
Mini Batch Size	32	Output Layer Final Layer	pixelClassificationLayer
Validation Frequency	2		
Initial Learn Rate	1e-3		

Εικόνα 106: Training – Παράδειγμα τιμών στα βήματα 2 και 3 της καρτέλας εκπαίδευσης

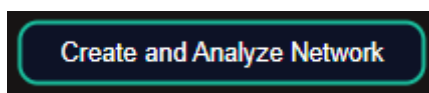
και ξεκινάμε τη πρώτη εκπαίδευση του δικτύου με τελικό layer το `pixelClassificationLayer` [41].

Βήμα 4: Στο σημείο αυτό, αφού έχουμε παραμετροποιήσει όλες τις επιλογές στη καρτέλα Training και εφόσον δεν θέλουμε να θέσουμε νέες τιμές στα DropOut layers μέσω:



Εικόνα 107: Training – Πρόσθετη ρύθμιση του DropOut layer στα στάδια encoder και bridge

τότε είμαστε έτοιμοι για τη δημιουργία και ανάλυση του δικτύου:



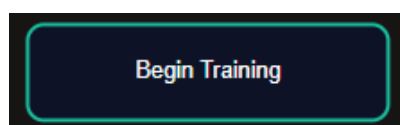
Εικόνα 108: Training – Κουμπί δημιουργίας και ανάλυσης του δικτύου με τις προηγούμενες παραμέτρους

με το δίκτυο να εμφανίζεται σε ξεχωριστό παράθυρο:

Name	Type	Activations	Learnable Properties	Status
1 imageinput	Image Input	64(S) × 64(S) × 2(C) × 1(B)	-	-
2 Encoder-Stage-1-Conv-1	2-D Convolution	64(S) × 64(S) × 64(C) × 1(B)	Weights: 3 × 3 × 2, Bias: 1 × 1 × 64	-
3 Encoder-Stage-1-ReLU-1	ReLU	64(S) × 64(S) × 64(C) × 1(B)	-	-
4 Encoder-Stage-1-Conv-2	2-D Convolution	64(S) × 64(S) × 64(C) × 1(B)	Weights: 3 × 3 × 64, Bias: 1 × 1 × 64	-
5 Encoder-Stage-1-ReLU-2	ReLU	64(S) × 64(S) × 64(C) × 1(B)	-	-
6 Encoder-Stage-1-MaxPool	2-D Max Pooling	32(S) × 32(S) × 64(C) × 1(B)	-	-
7 Encoder-Stage-2-Conv-1	2-D Convolution	32(S) × 32(S) × 128(C) × 1(B)	Weights: 3 × 3 × 64, Bias: 1 × 1 × 128	-
8 Encoder-Stage-2-ReLU-1	ReLU	32(S) × 32(S) × 128(C) × 1(B)	-	-
9 Encoder-Stage-2-Conv-2	2-D Convolution	32(S) × 32(S) × 128(C) × 1(B)	Weights: 3 × 3 × 128, Bias: 1 × 1 × 128	-
10 Encoder-Stage-2-ReLU-2	ReLU	32(S) × 32(S) × 128(C) × 1(B)	-	-
11 Encoder-Stage-2-MaxPool	2-D Max Pooling	16(S) × 16(S) × 128(C) × 1(B)	-	-
12 Encoder-Stage-3-Conv-1	2-D Convolution	16(S) × 16(S) × 256(C) × 1(B)	Weights: 3 × 3 × 128, Bias: 1 × 1 × 256	-

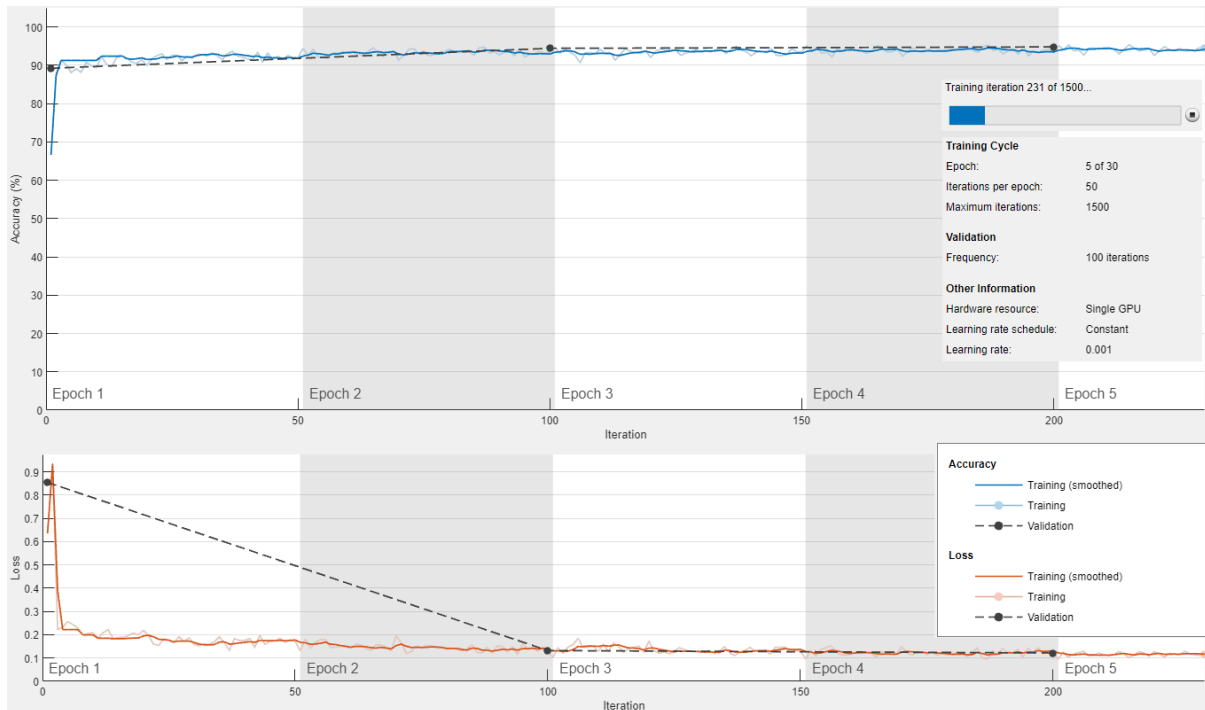
Εικόνα 109: Training – Αναλυτική παρουσίαση των layers του δικτύου σε ξεχωριστό παράθυρο (analyze)

και με την επιβεβαίωση ότι δεν υπάρχουν warnings / errors, ξεκινάμε την εκπαίδευση:



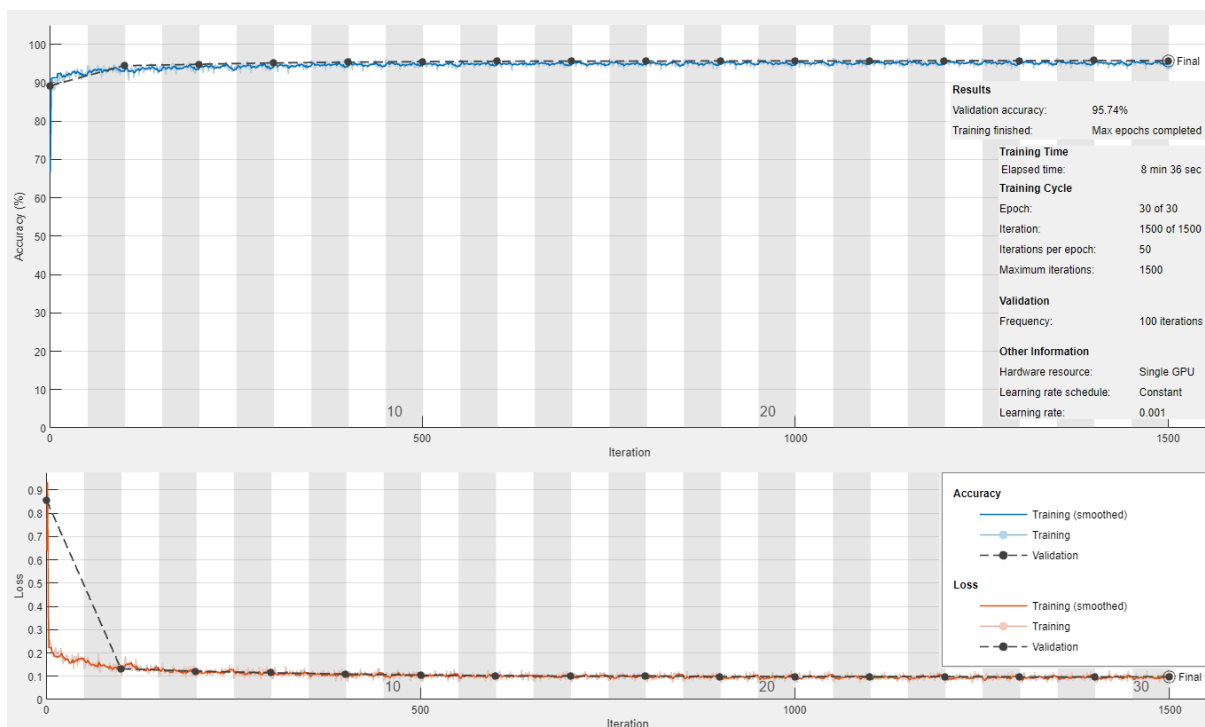
Εικόνα 110: Training – Κουμπί εκκίνησης εκπαίδευσης του δικτύου που δημιουργήθηκε

με το οποίο ξεκινάει η εκπαίδευση του δικτύου που δημιουργήσαμε και ταυτόχρονα ανοίγει ένα ξεχωριστό παράθυρο οπτικής παρουσίασης της προόδου της εκπαίδευσης:



Εικόνα 111: Training – Ξεχωριστό παράθυρο παρουσίασης προόδου εκπαίδευσης

και το τελικό αποτέλεσμα:



Εικόνα 112: Training – Ξεχωριστό παράθυρο παρουσίασης τελικού αποτελέσματος εκπαίδευσης

Training Progress:

Στη καρτέλα Training Progress, βλέπουμε τα αποτελέσματα της εκπαίδευσης σε πιο αναλυτική μορφή:

```
Training on single GPU.
Initializing input data normalization.
```

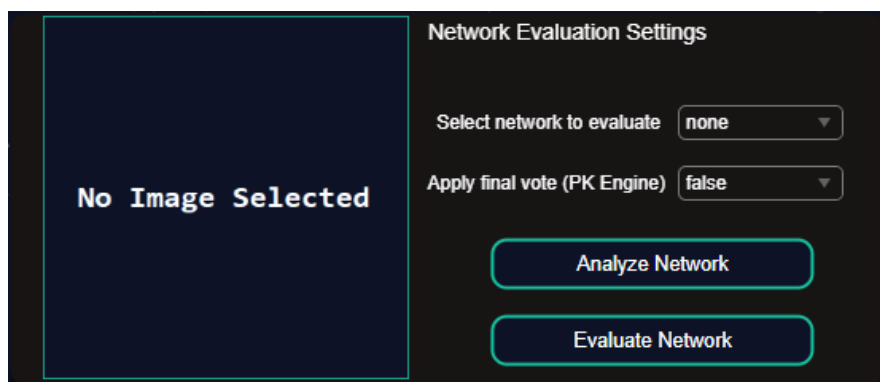
Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss	Base Learning Rate
1	1	00:00:09	66.59%	89.22%	0.6360	0.8544	0.0010
1	50	00:00:21	91.96%		0.1720		0.0010
2	100	00:00:47	93.36%	94.42%	0.1283	0.1321	0.0010
3	150	00:01:06	93.49%		0.1298		0.0010
4	200	00:01:28	93.84%	94.79%	0.1215	0.1218	0.0010
5	250	00:01:43	93.59%		0.1236		0.0010
6	300	00:02:03	94.19%	95.15%	0.1163	0.1149	0.0010
7	350	00:02:17	94.38%		0.1106		0.0010
8	400	00:02:38	94.29%	95.32%	0.1136	0.1097	0.0010
9	450	00:02:51	94.40%		0.1059		0.0010
10	500	00:03:10	94.49%	95.48%	0.1035	0.1045	0.0010
11	550	00:03:24	94.28%		0.0979		0.0010
12	600	00:03:44	94.65%	95.56%	0.1082	0.1026	0.0010
13	650	00:03:57	94.50%		0.1065		0.0010
14	700	00:04:17	94.82%	95.62%	0.0928	0.1004	0.0010
15	750	00:04:30	94.75%		0.1001		0.0010
16	800	00:04:49	94.41%	95.57%	0.1034	0.1014	0.0010
17	850	00:05:02	94.57%		0.1044		0.0010
18	900	00:05:21	94.68%	95.68%	0.0995	0.0987	0.0010
19	950	00:05:33	94.70%		0.1040		0.0010
20	1000	00:05:51	94.65%	95.67%	0.1005	0.0988	0.0010
21	1050	00:06:05	94.80%		0.1001		0.0010
22	1100	00:06:25	95.09%	95.67%	0.0900	0.0983	0.0010
23	1150	00:06:38	94.86%		0.1011		0.0010
24	1200	00:06:56	94.68%	95.71%	0.0984	0.0983	0.0010
25	1250	00:07:09	95.10%		0.0936		0.0010
26	1300	00:07:27	94.57%	95.73%	0.1014	0.0972	0.0010
27	1350	00:07:39	94.82%		0.1012		0.0010
28	1400	00:07:58	94.65%	95.75%	0.1012	0.0965	0.0010
29	1450	00:08:11	94.91%		0.0957		0.0010
30	1500	00:08:29	94.29%	95.74%	0.1029	0.0977	0.0010

```
Training finished: Max epochs completed.
```

Εικόνα 113: Training Progress – Αναλυτική μορφή παρουσίασης της προόδου εκπαίδευσης

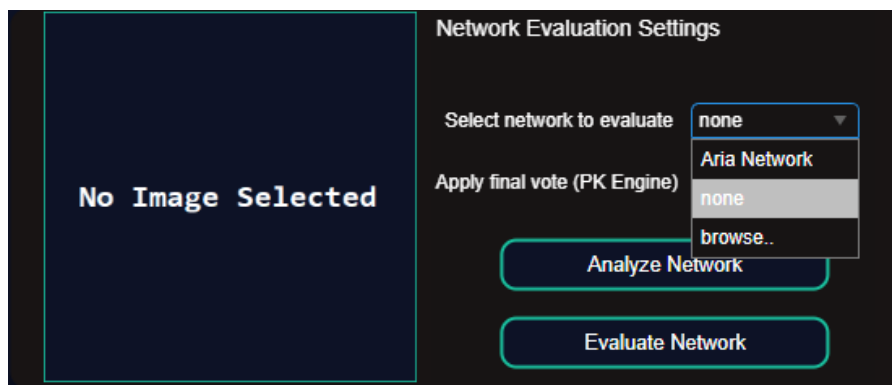
Evaluation:

Στη καρτέλα Evaluation, αξιολογούμε το δίκτυο που μόλις εκπαιδεύσαμε με το τελικό επίπεδο να είναι το **pixelClassificationLayer** [41].



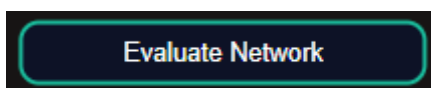
Εικόνα 114: Evaluation – Πεδία προβολής τελικού αποτελέσματος κατάτμησης και επιπλέον ρυθμίσεις

επιλέγοντας την δίκτυο **Aria Network**:



Εικόνα 115: Evaluation – Επιλογή δικτύου για αξιολόγηση

και ξεκινώντας την αξιολόγηση με το κουμπί Evaluate Network:



Εικόνα 116: Evaluation – Κουμπί εκκίνησης αξιολόγησης του δικτύου

και τα αποτελέσματα του δικτύου με 30 epochs, ρυθμό εκπαίδευσης $1e-3$ και μόλις ~9 λεπτά συνολικής εκπαίδευσης είναι τα εξής:

Lowest		Highest	
Accuracy	0.9648 Image: 13	Accuracy	0.975 Image: 5
Sensitivity	0.69 Image: 9	Sensitivity	0.8692 Image: 1
Specificity	0.9753 Image: 1	Specificity	0.9927 Image: 9
Precision	0.7734 Image: 1	Precision	0.9015 Image: 20
F1 Score	0.7786 Image: 9	F1 Score	0.8548 Image: 2
IoU	0.6374 Image: 9	IoU	0.7464 Image: 2
AUC	0.8414 Image: 9	AUC	0.9223 Image: 1

Εικόνα 117: Evaluation – Προβολή των αποτελεσμάτων αξιολόγησης του δικτύου Aria Network

Μέσος όρος δεικτών απόδοσης – αξιολόγησης:

```
=== Total Images: 20
= Mean accuracy: 0.97021
= Mean sensitivity: 0.80075
= Mean specificity: 0.98607
= Mean precision: 0.84246
= Mean F1 Score: 0.81900
= Mean Intersection over Union: 0.69393
= Mean Area under Curve: 0.89341
```

Εικόνα 118: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Ελάχιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Lowest
Accuracy 0.9648 Image: 13
Sensitivity 0.69 Image: 9
Specificity 0.9753 Image: 1
Precision 0.7734 Image: 1
F1 Score 0.7786 Image: 9
IoU 0.6374 Image: 9
AUC 0.8414 Image: 9
```

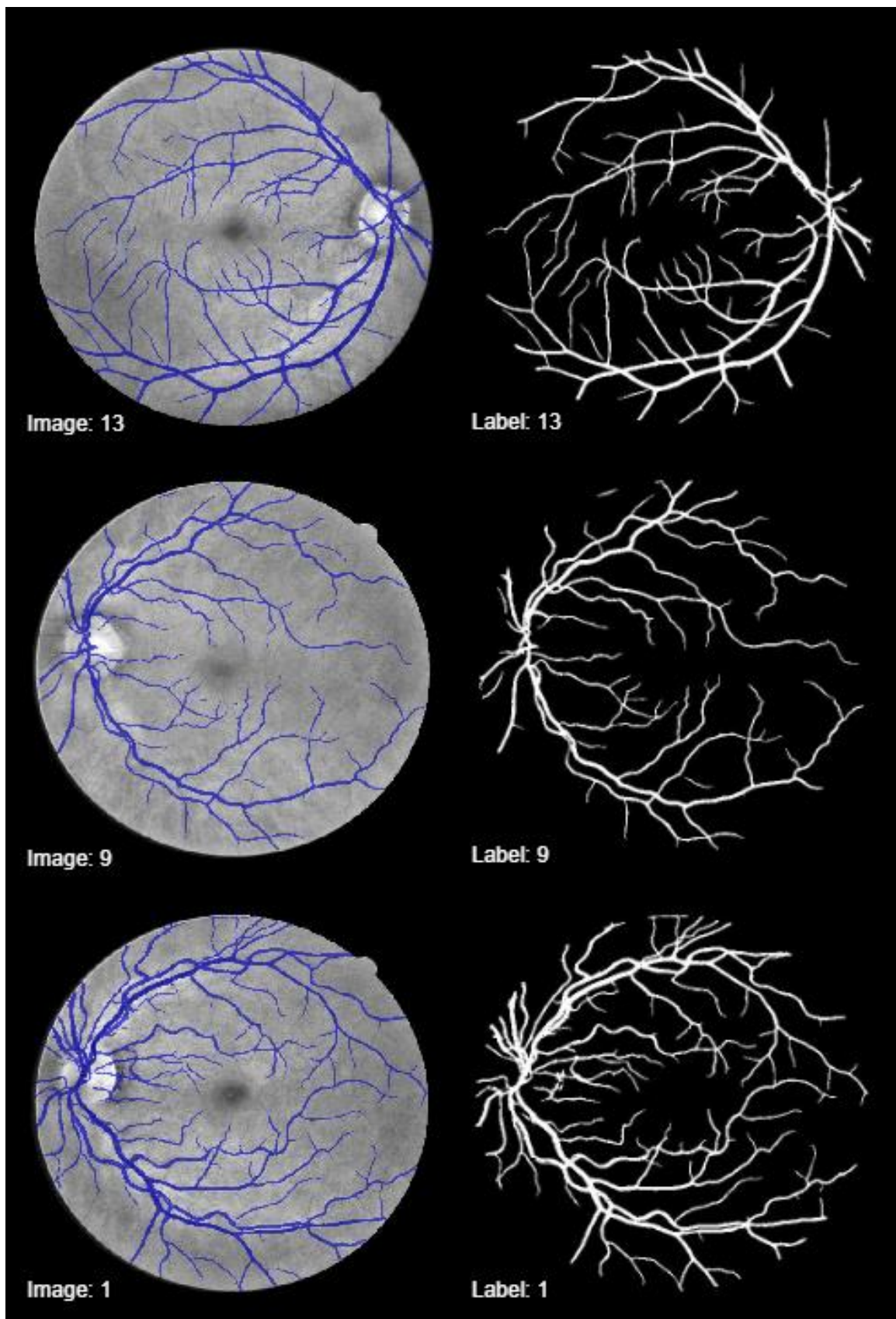
Εικόνα 119: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Μέγιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Highest
Accuracy 0.975 Image: 5
Sensitivity 0.8692 Image: 1
Specificity 0.9927 Image: 9
Precision 0.9015 Image: 20
F1 Score 0.8548 Image: 2
IoU 0.7464 Image: 2
AUC 0.9223 Image: 1
```

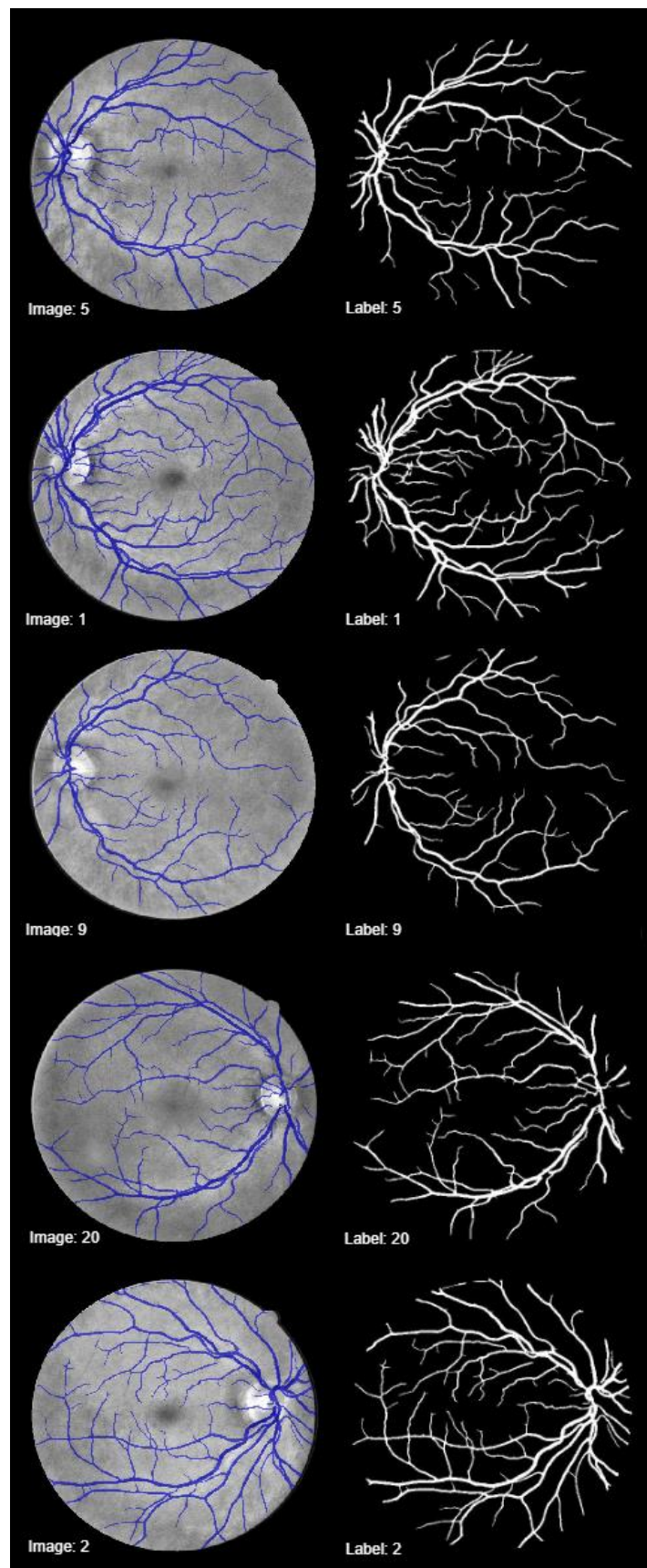
Εικόνα 120: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης

Εικόνες με την ελάχιστη τιμή για κάθε δείκτη:



Εικόνα 121: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνες με τη μέγιστη τιμή για κάθε δείκτη:

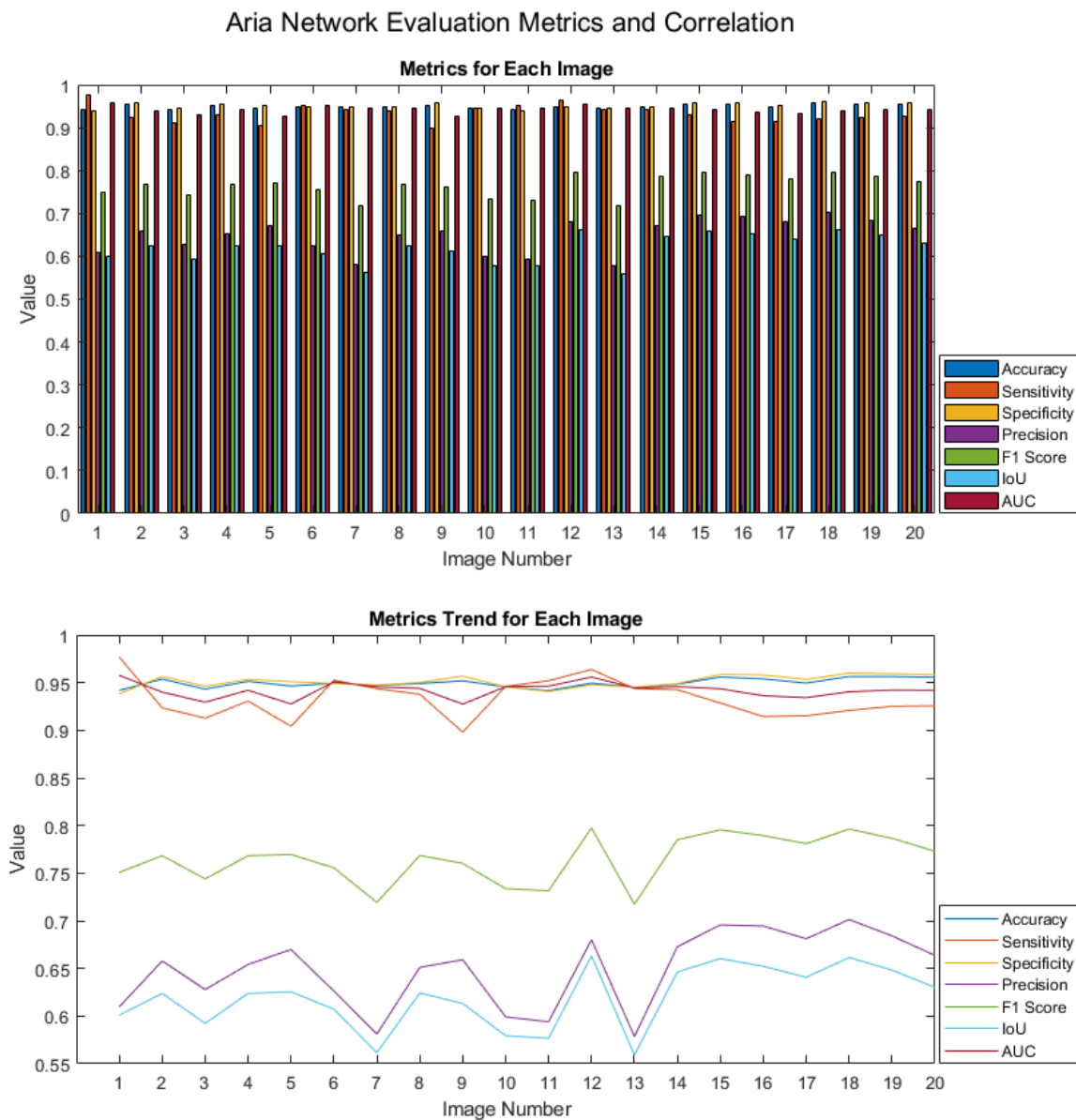


Εικόνα 122: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

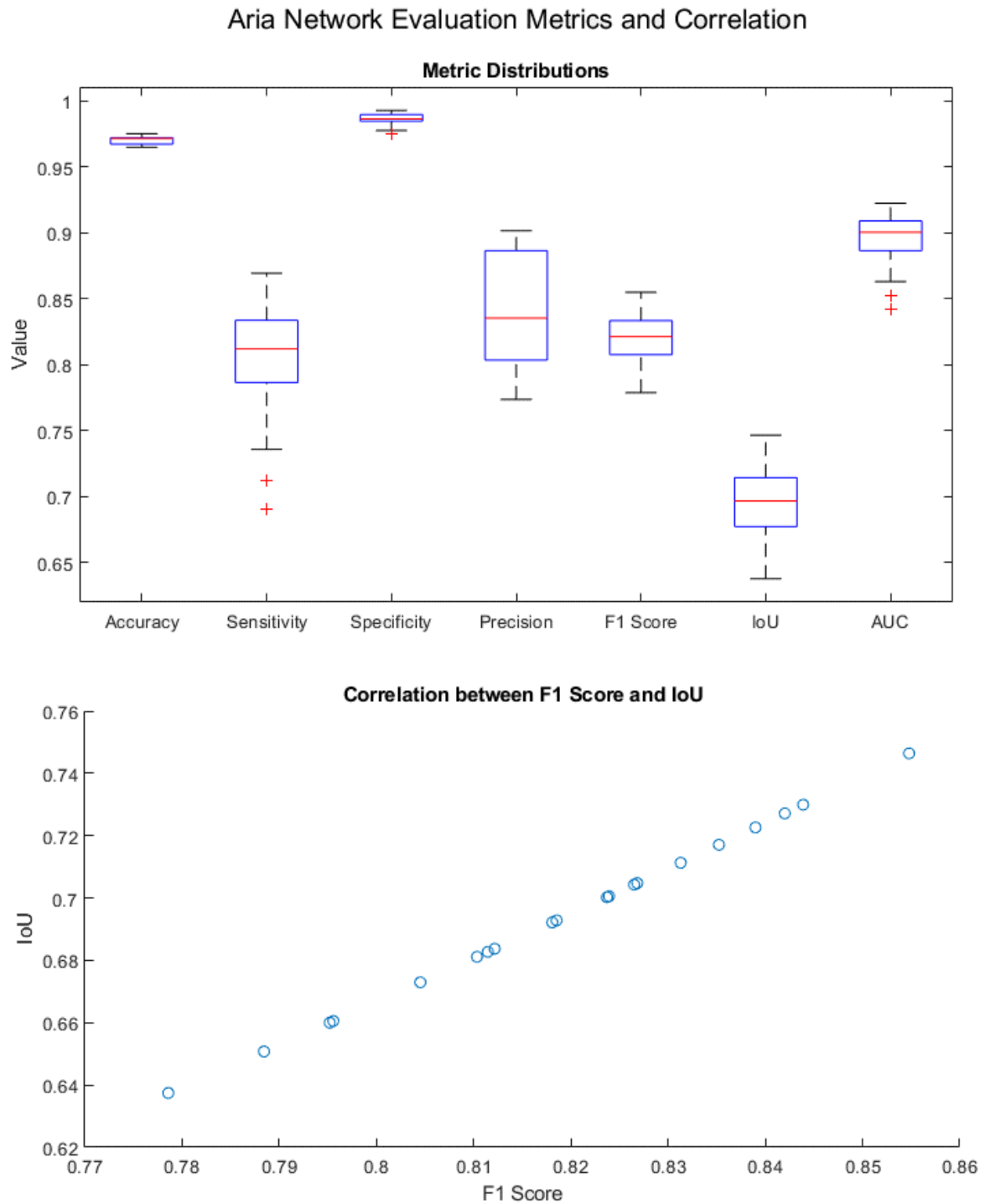
Με το τέλος της αξιολόγησης όλων των εικόνων και της παρουσίασης των αποτελεσμάτων στη καρτέλα Evaluation, σε ξεχωριστό παράθυρο εμφανίζονται και οι παρακάτω γραφικές παραστάσεις:

Γραφικές παραστάσεις και συσχέτιση:

(1/2)



Εικόνα 123: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

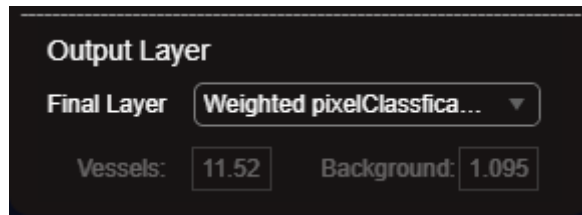


Εικόνα 124: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

7.2 Αξιολόγηση με Weighted pixelClassificationLayer

Evaluation – 2

Στη καρτέλα Evaluation, αξιολογούμε το δίκτυο που μόλις εκπαιδεύσαμε με το τελικό επίπεδο να είναι το **Weighted pixelClassificationLayer** [41], με **classWeights** που έχουν υπολογιστεί αυτόματα:



Εικόνα 125: Evaluation – Επιλογή Weighted pixelClassificationLayer ως final output layer

και τα αποτελέσματα του δικτύου με 30 epochs, ρυθμό εκπαίδευσης $1e-3$ και μόλις ~9 λεπτά συνολικής εκπαίδευσης είναι τα εξής:

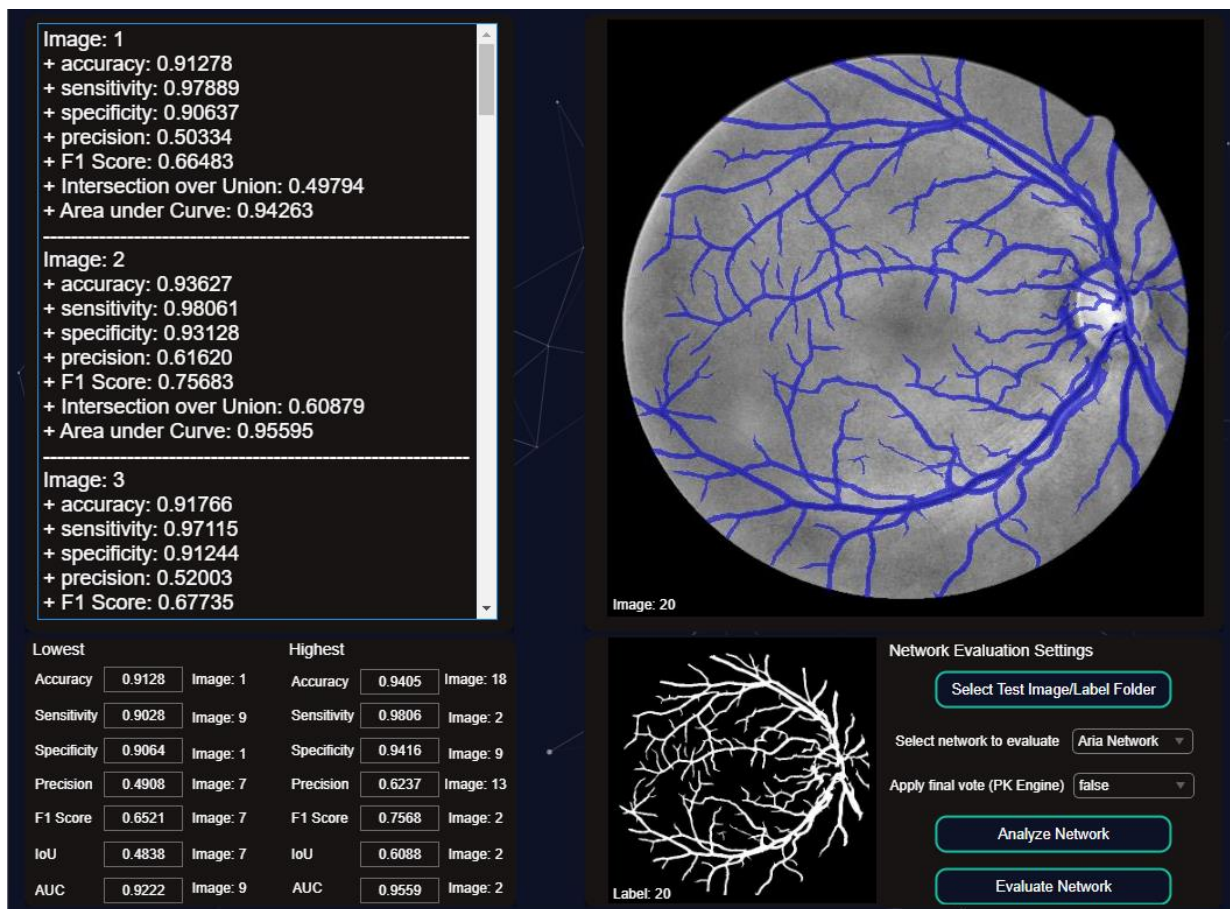


Image: 1

- + accuracy: 0.91278
- + sensitivity: 0.97889
- + specificity: 0.90637
- + precision: 0.50334
- + F1 Score: 0.66483
- + Intersection over Union: 0.49794
- + Area under Curve: 0.94263

Image: 2

- + accuracy: 0.93627
- + sensitivity: 0.98061
- + specificity: 0.93128
- + precision: 0.61620
- + F1 Score: 0.75683
- + Intersection over Union: 0.60879
- + Area under Curve: 0.95595

Image: 3

- + accuracy: 0.91766
- + sensitivity: 0.97115
- + specificity: 0.91244
- + precision: 0.52003
- + F1 Score: 0.67735

Lowest		Highest	
Accuracy	0.9128 Image: 1	Accuracy	0.9405 Image: 18
Sensitivity	0.9028 Image: 9	Sensitivity	0.9806 Image: 2
Specificity	0.9064 Image: 1	Specificity	0.9416 Image: 9
Precision	0.4908 Image: 7	Precision	0.6237 Image: 13
F1 Score	0.6521 Image: 7	F1 Score	0.7568 Image: 2
IoU	0.4838 Image: 7	IoU	0.6088 Image: 2
AUC	0.9222 Image: 9	AUC	0.9559 Image: 2

Network Evaluation Settings

- Select Test Image/Label Folder
- Select network to evaluate: Aria Network
- Apply final vote (PK Engine): false
- Analyze Network
- Evaluate Network

Εικόνα 126: Evaluation – Προβολή των αποτελεσμάτων αξιολόγησης του δικτύου Aria Network

Μέσος όρος δεικτών απόδοσης – αξιολόγησης:

```
=== Total Images: 20
= Mean accuracy: 0.93054
= Mean sensitivity: 0.95785
= Mean specificity: 0.92810
= Mean precision: 0.55255
= Mean F1 Score: 0.69918
= Mean Intersection over Union: 0.53860
= Mean Area under Curve: 0.94297
```

Εικόνα 127: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Ελάχιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Lowest
Accuracy 0.9128 Image: 1
Sensitivity 0.9028 Image: 9
Specificity 0.9064 Image: 1
Precision 0.4908 Image: 7
F1 Score 0.6521 Image: 7
IoU 0.4838 Image: 7
AUC 0.9222 Image: 9
```

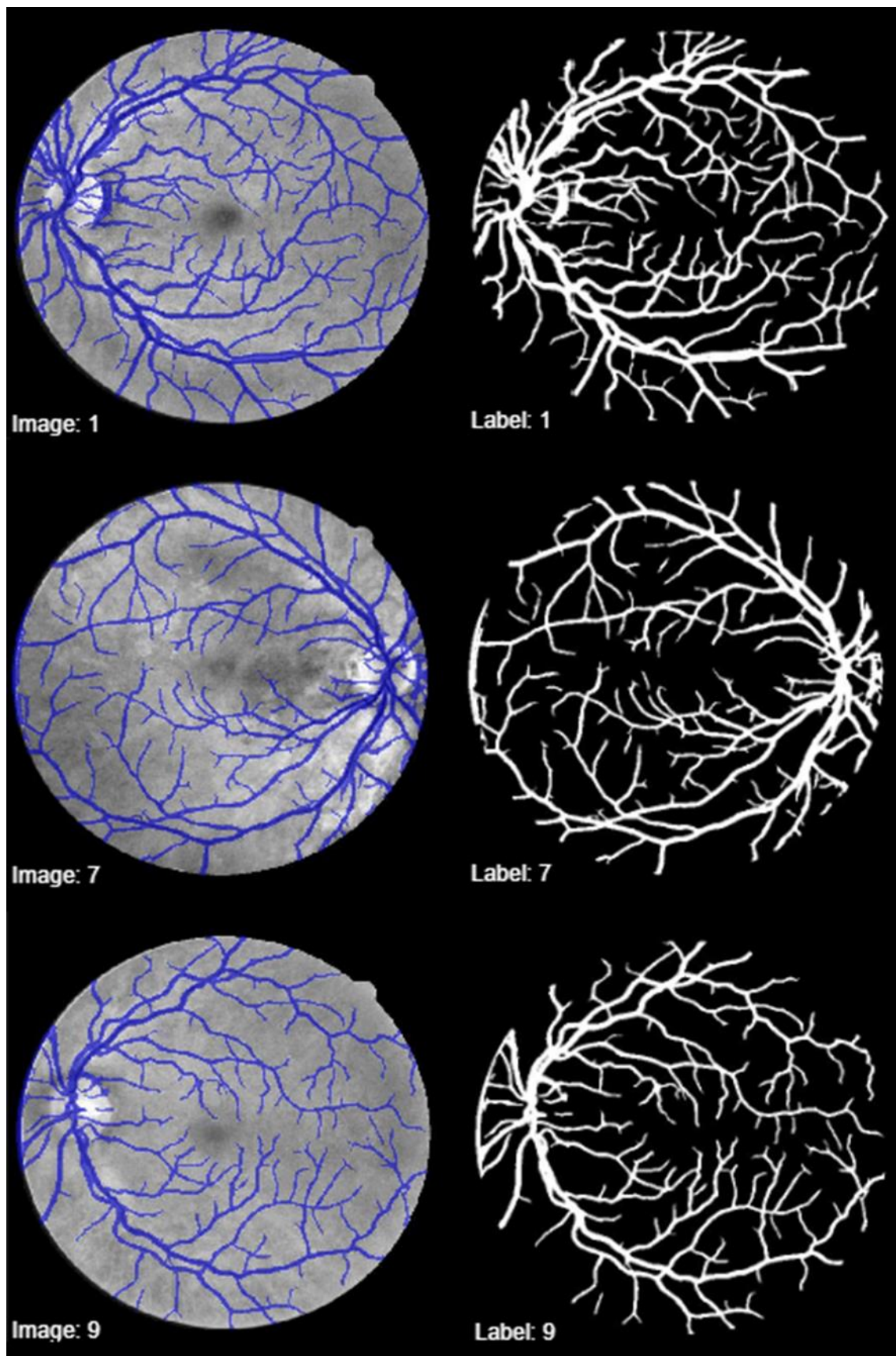
Εικόνα 128: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Μέγιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Highest
Accuracy 0.9405 Image: 18
Sensitivity 0.9806 Image: 2
Specificity 0.9416 Image: 9
Precision 0.6237 Image: 13
F1 Score 0.7568 Image: 2
IoU 0.6088 Image: 2
AUC 0.9559 Image: 2
```

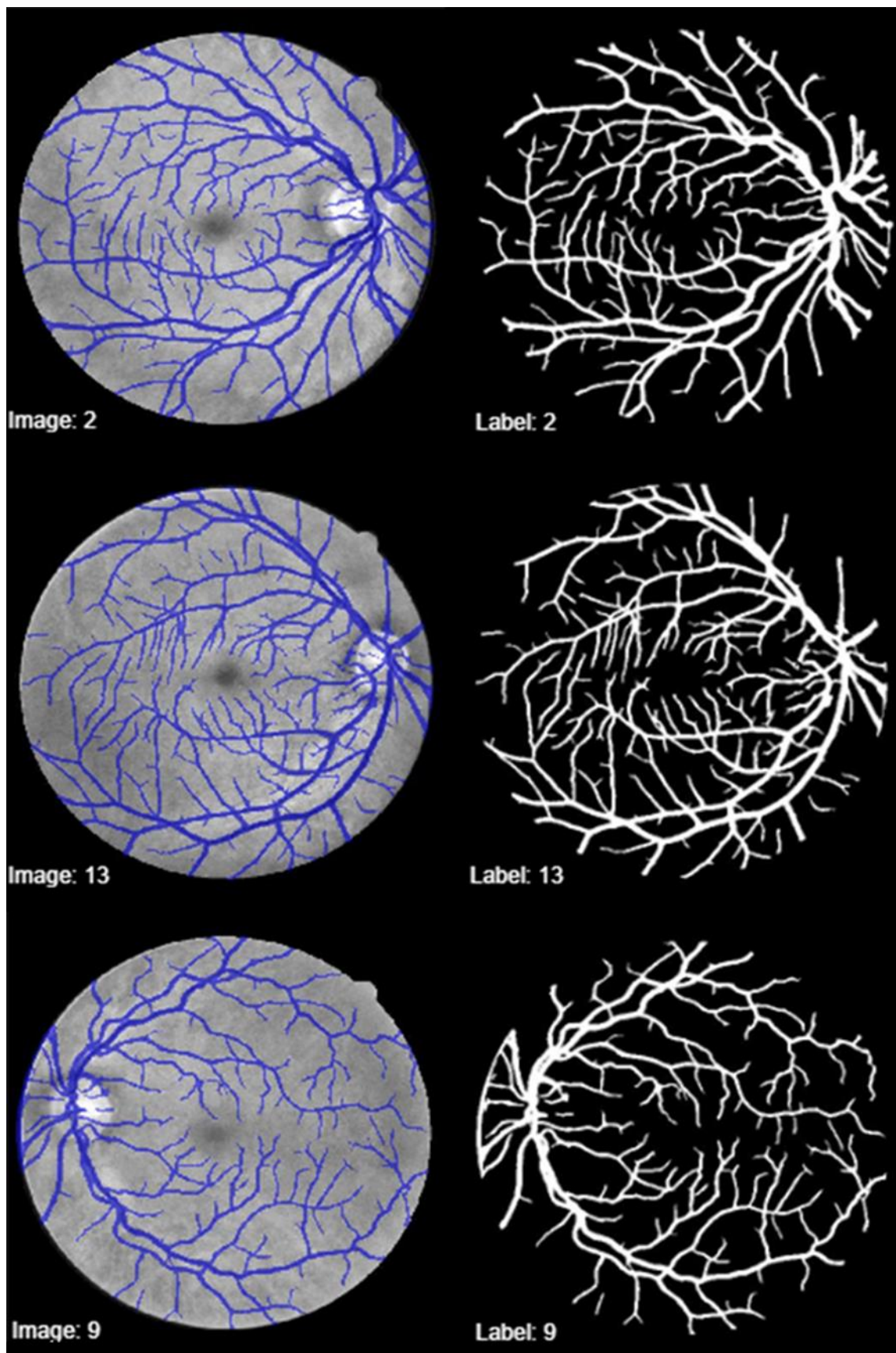
Εικόνα 129: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης

Εικόνες με την ελάχιστη τιμή για κάθε δείκτη:



Εικόνα 130: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνες με τη μέγιστη τιμή για κάθε δείκτη:

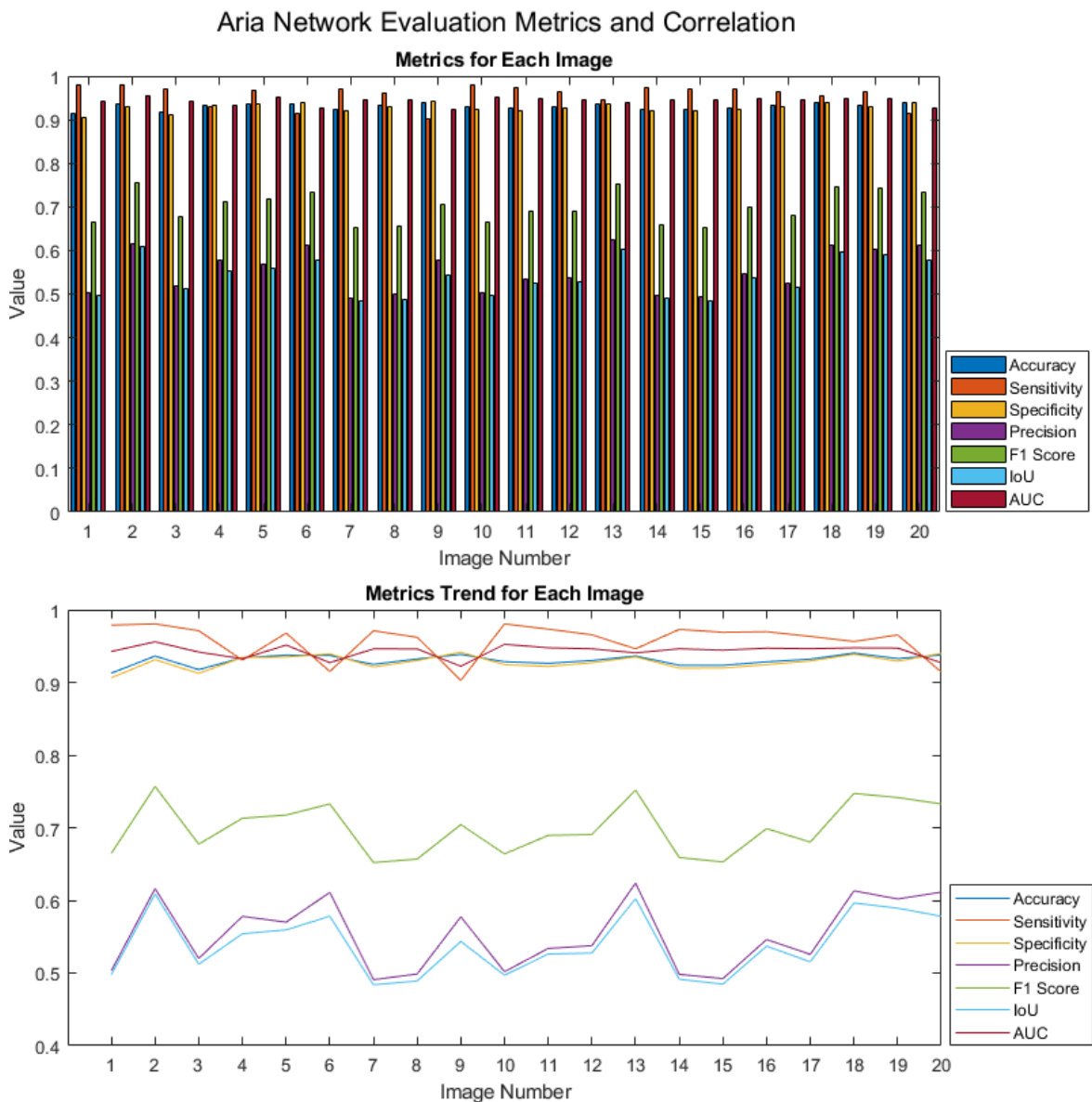


Εικόνα 131: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

Με το τέλος της αξιολόγησης όλων των εικόνων και της παρουσίασης των αποτελεσμάτων στη καρτέλα Evaluation, σε ξεχωριστό παράθυρο εμφανίζονται και οι παρακάτω γραφικές παραστάσεις:

Γραφικές παραστάσεις και συσχέτιση:

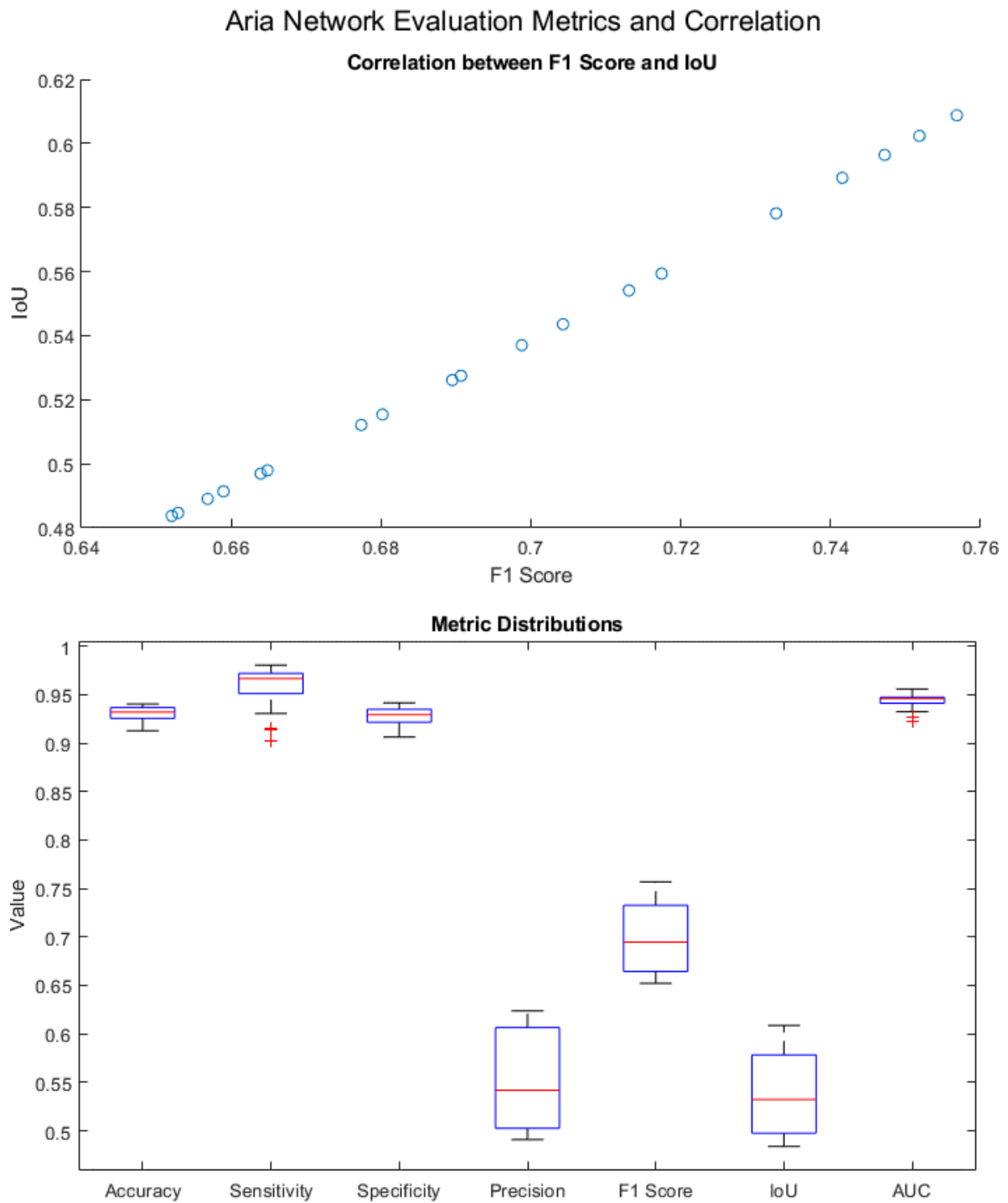
(1/2)



Εικόνα 132: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

Γραφικές παραστάσεις και συσχέτιση:

(2/2)

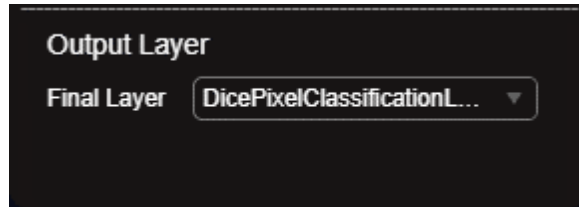


Εικόνα 133: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

7.3 Αξιολόγηση με DicePixelClassificationLayer

Evaluation – 3

Στη καρτέλα Evaluation, αξιολογούμε το δίκτυο που μόλις εκπαιδεύσαμε με το τελικό επίπεδο να είναι το **DicePixelClassificationLayer** [41]:



Εικόνα 134: Evaluation – Επιλογή DicePixelClassificationLayer ως final output layer

και τα αποτελέσματα του δικτύου με 30 epochs, ρυθμό εκπαίδευσης $1e-3$ και μόλις ~9 λεπτά συνολικής εκπαίδευσης είναι τα εξής:

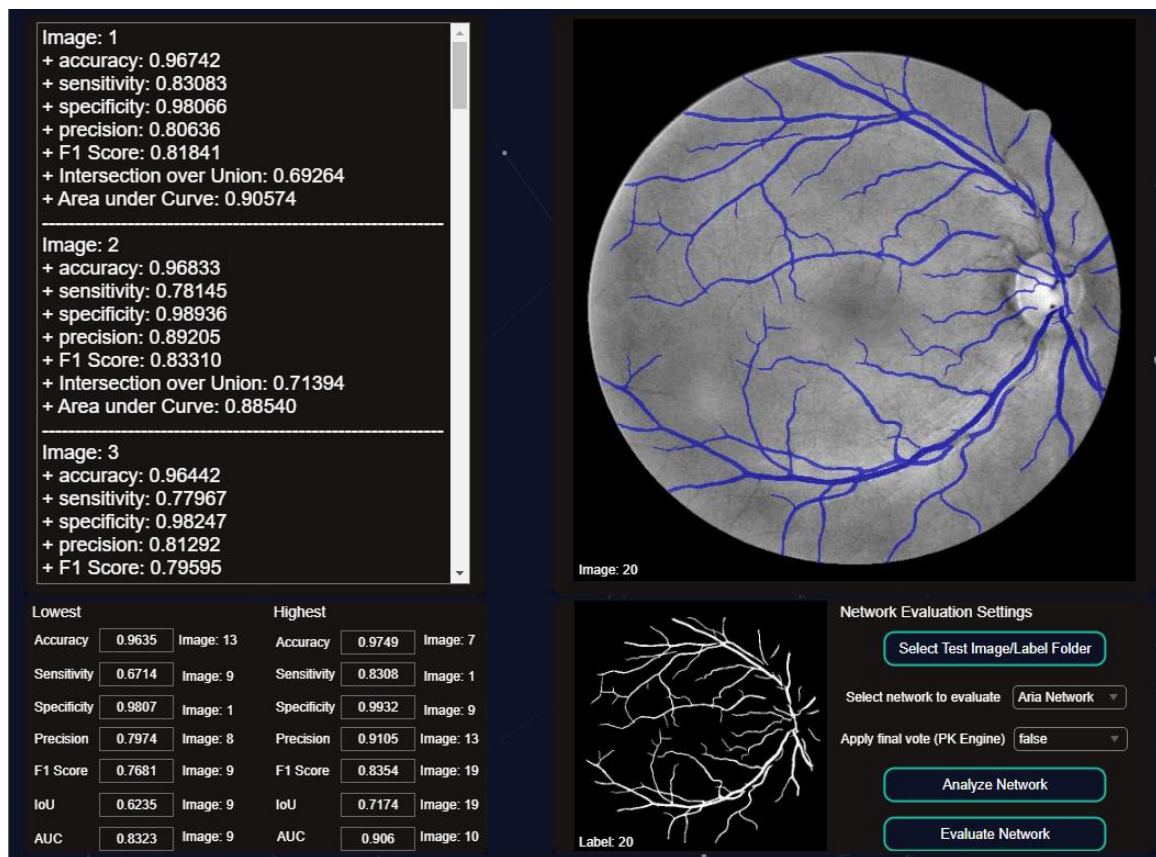


Image: 1
+ accuracy: 0.96742
+ sensitivity: 0.83083
+ specificity: 0.98066
+ precision: 0.80636
+ F1 Score: 0.81841
+ Intersection over Union: 0.69264
+ Area under Curve: 0.90574

Image: 2
+ accuracy: 0.96833
+ sensitivity: 0.78145
+ specificity: 0.98936
+ precision: 0.89205
+ F1 Score: 0.83310
+ Intersection over Union: 0.71394
+ Area under Curve: 0.88540

Image: 3
+ accuracy: 0.96442
+ sensitivity: 0.77967
+ specificity: 0.98247
+ precision: 0.81292
+ F1 Score: 0.79595

Lowest		Highest	
Accuracy	0.9635 Image: 13	Accuracy	0.9749 Image: 7
Sensitivity	0.6714 Image: 9	Sensitivity	0.8308 Image: 1
Specificity	0.9807 Image: 1	Specificity	0.9932 Image: 9
Precision	0.7974 Image: 8	Precision	0.9105 Image: 13
F1 Score	0.7681 Image: 9	F1 Score	0.8354 Image: 19
IoU	0.6235 Image: 9	IoU	0.7174 Image: 19
AUC	0.8323 Image: 9	AUC	0.906 Image: 10

Network Evaluation Settings

Select Test Image/Label Folder

Select network to evaluate: Aria Network

Apply final vote (PK Engine): false

Analyze Network

Evaluate Network

Εικόνα 135: Evaluation – Προβολή των αποτελεσμάτων αξιολόγησης του δικτύου Aria Network

Μέσος όρος δεικτών απόδοσης – αξιολόγησης:

```
=== Total Images: 20
= Mean accuracy: 0.96949
= Mean sensitivity: 0.77303
= Mean specificity: 0.98790
= Mean precision: 0.85507
= Mean F1 Score: 0.81017
= Mean Intersection over Union: 0.68140
= Mean Area under Curve: 0.88046
```

Εικόνα 136: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Ελάχιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Lowest
Accuracy 0.9635 Image: 13
Sensitivity 0.6714 Image: 9
Specificity 0.9807 Image: 1
Precision 0.7974 Image: 8
F1 Score 0.7681 Image: 9
IoU 0.6235 Image: 9
AUC 0.8323 Image: 9
```

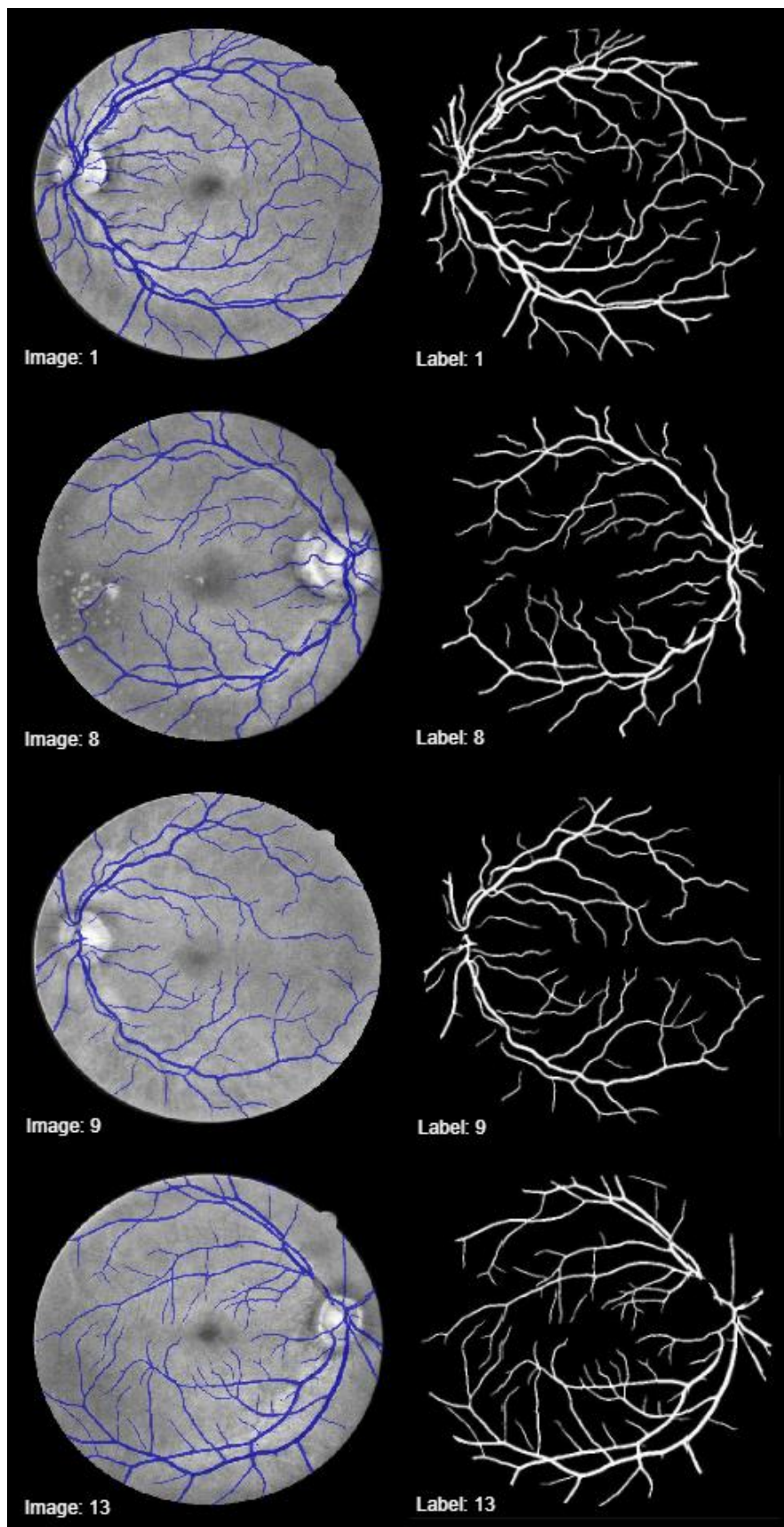
Εικόνα 137: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Μέγιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Highest
Accuracy 0.9749 Image: 7
Sensitivity 0.8308 Image: 1
Specificity 0.9932 Image: 9
Precision 0.9105 Image: 13
F1 Score 0.8354 Image: 19
IoU 0.7174 Image: 19
AUC 0.906 Image: 10
```

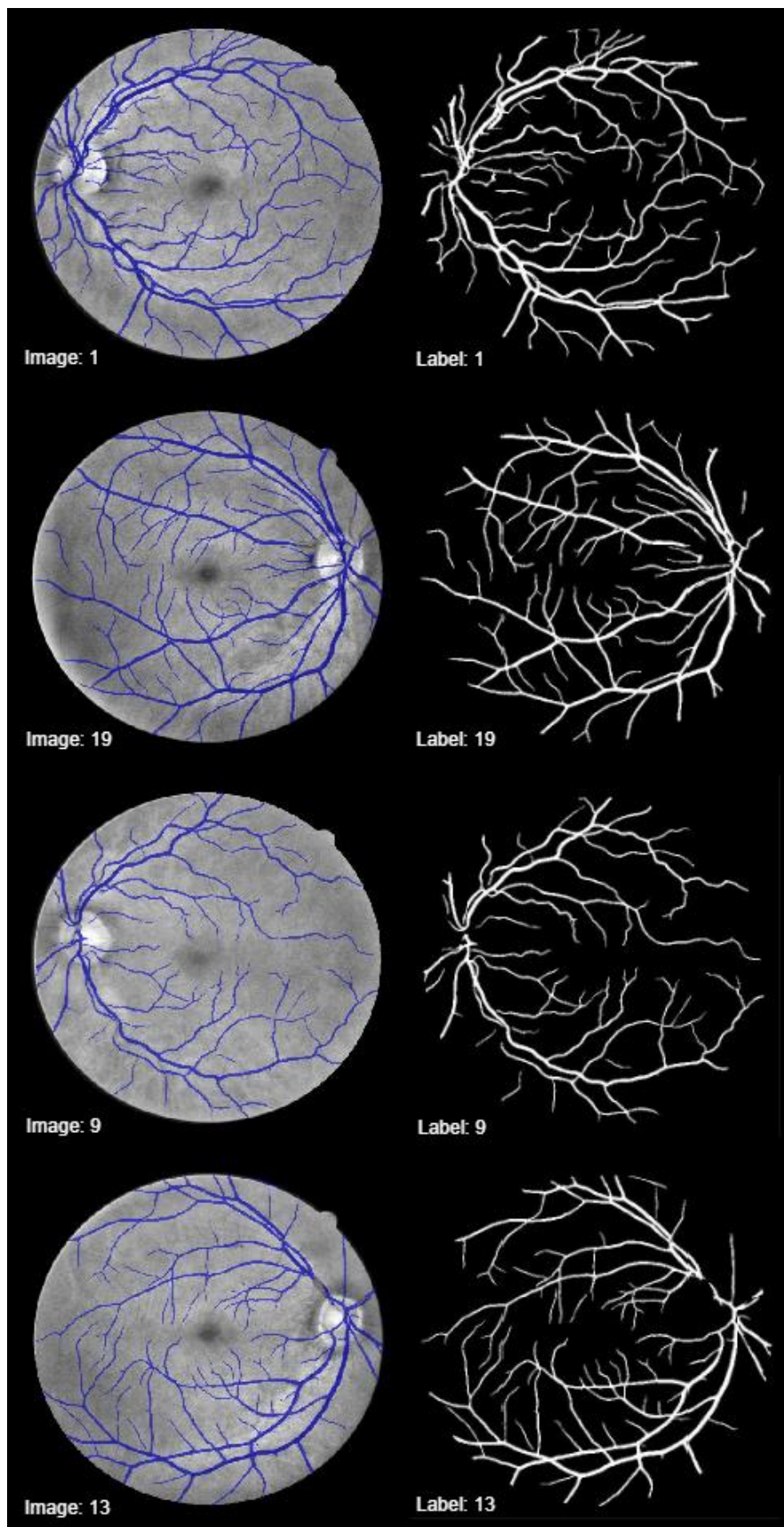
Εικόνα 138: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης

Εικόνες με την ελάχιστη τιμή για κάθε δείκτη:



Εικόνα 139: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνες με τη μέγιστη τιμή για κάθε δείκτη:

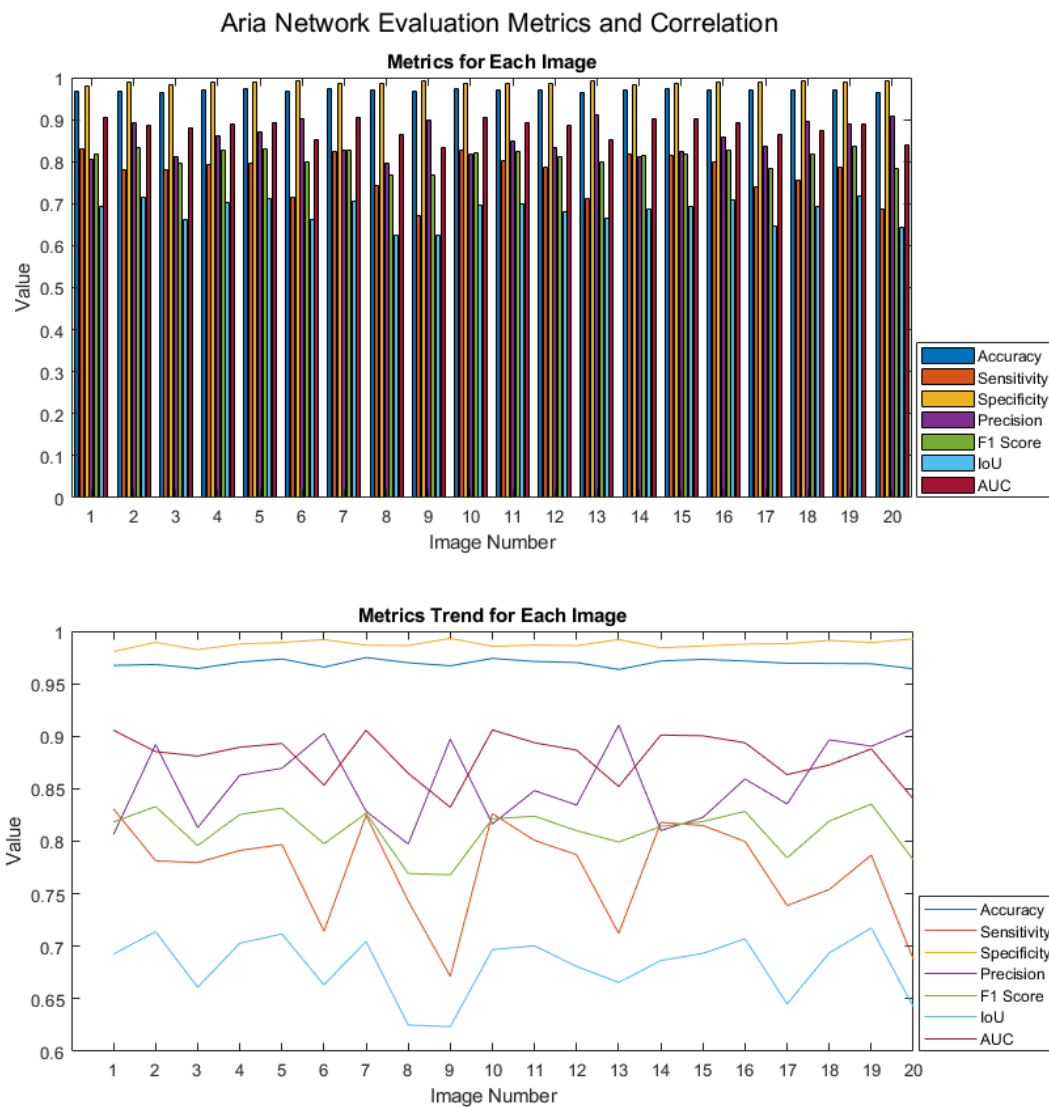


Εικόνα 140: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

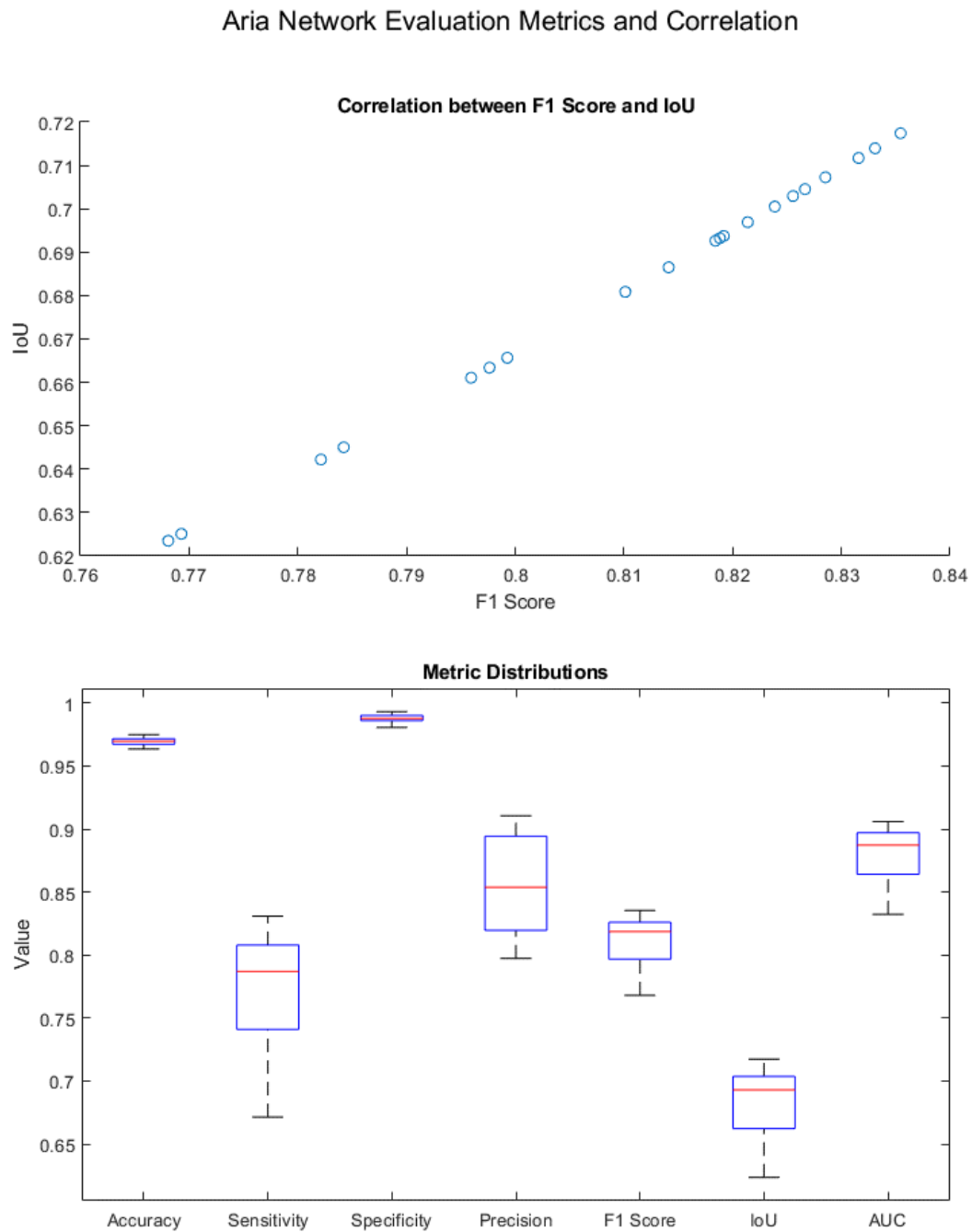
Με το τέλος της αξιολόγησης όλων των εικόνων και της παρουσίασης των αποτελεσμάτων στη καρτέλα Evaluation, σε ξεχωριστό παράθυρο εμφανίζονται και οι παρακάτω γραφικές παραστάσεις:

Γραφικές παραστάσεις και συσχέτιση:

(1/2)



Εικόνα 141: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

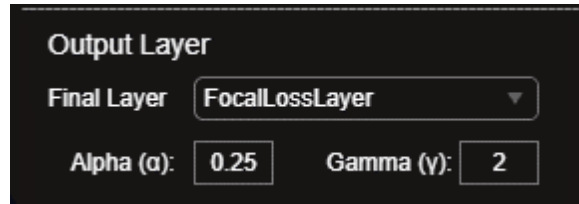


Εικόνα 142: Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

7.4 Αξιολόγηση με Focal Loss Layer

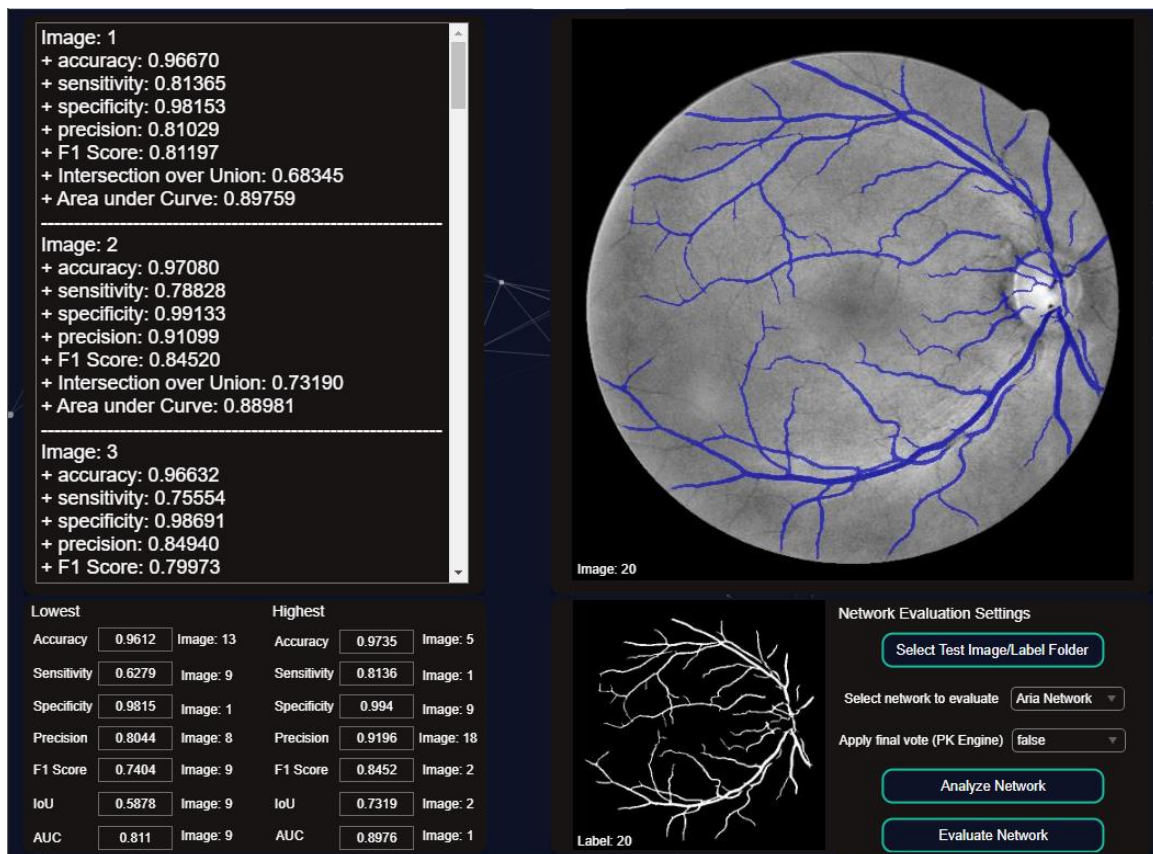
Evaluation – 4

Στη καρτέλα Evaluation, αξιολογούμε το δίκτυο που μόλις εκπαιδεύσαμε με το τελικό επίπεδο να είναι το **focalLossLayer** [41], με τιμές $\alpha = 0.25$, $\gamma = 2$:



Εικόνα 143: Επιλογή FocalLossLayer ως final output layer

και τα αποτελέσματα του δικτύου με 30 epochs, ρυθμό εκπαίδευσης $1e-3$ και μόλις ~9 λεπτά συνολικής εκπαίδευσης είναι τα εξής:



The screenshot displays the evaluation results for the Aria Network. On the left, a scrollable list shows performance metrics for three images:

- Image: 1**
 - + accuracy: 0.96670
 - + sensitivity: 0.81365
 - + specificity: 0.98153
 - + precision: 0.81029
 - + F1 Score: 0.81197
 - + Intersection over Union: 0.68345
 - + Area under Curve: 0.89759
- Image: 2**
 - + accuracy: 0.97080
 - + sensitivity: 0.78828
 - + specificity: 0.99133
 - + precision: 0.91099
 - + F1 Score: 0.84520
 - + Intersection over Union: 0.73190
 - + Area under Curve: 0.88981
- Image: 3**
 - + accuracy: 0.96632
 - + sensitivity: 0.75554
 - + specificity: 0.98691
 - + precision: 0.84940
 - + F1 Score: 0.79973

Below the list, a table compares the lowest and highest values for each metric across different images:

Metric	Lowest Value	Lowest Image	Highest Value	Highest Image
Accuracy	0.9612	Image: 13	0.9735	Image: 5
Sensitivity	0.6279	Image: 9	0.8136	Image: 1
Specificity	0.9815	Image: 1	0.994	Image: 9
Precision	0.8044	Image: 8	0.9196	Image: 18
F1 Score	0.7404	Image: 9	0.8452	Image: 2
IoU	0.5878	Image: 9	0.7319	Image: 2
AUC	0.811	Image: 9	0.8976	Image: 1

On the right, there are two network visualizations: 'Image: 20' showing a dense network of blue vessels and 'Label: 20' showing a sparser network of white vessels. The 'Network Evaluation Settings' panel includes a 'Select Test Image/Label Folder' button, a 'Select network to evaluate' dropdown set to 'Aria Network', an 'Apply final vote (FK Engine)' dropdown set to 'false', and 'Analyze Network' and 'Evaluate Network' buttons.

Εικόνα 144: Evaluation – Προβολή των αποτελεσμάτων αξιολόγησης του δικτύου Aria Network

Μέσος όρος δεικτών απόδοσης – αξιολόγησης:

```
=== Total Images: 20
= Mean accuracy: 0.96859
= Mean sensitivity: 0.74441
= Mean specificity: 0.98953
= Mean precision: 0.86775
= Mean F1 Score: 0.79941
= Mean Intersection over Union: 0.66656
= Mean Area under Curve: 0.86697
```

Εικόνα 145: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Ελάχιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Lowest
Accuracy 0.9612 Image: 13
Sensitivity 0.6279 Image: 9
Specificity 0.9815 Image: 1
Precision 0.8044 Image: 8
F1 Score 0.7404 Image: 9
IoU 0.5878 Image: 9
AUC 0.811 Image: 9
```

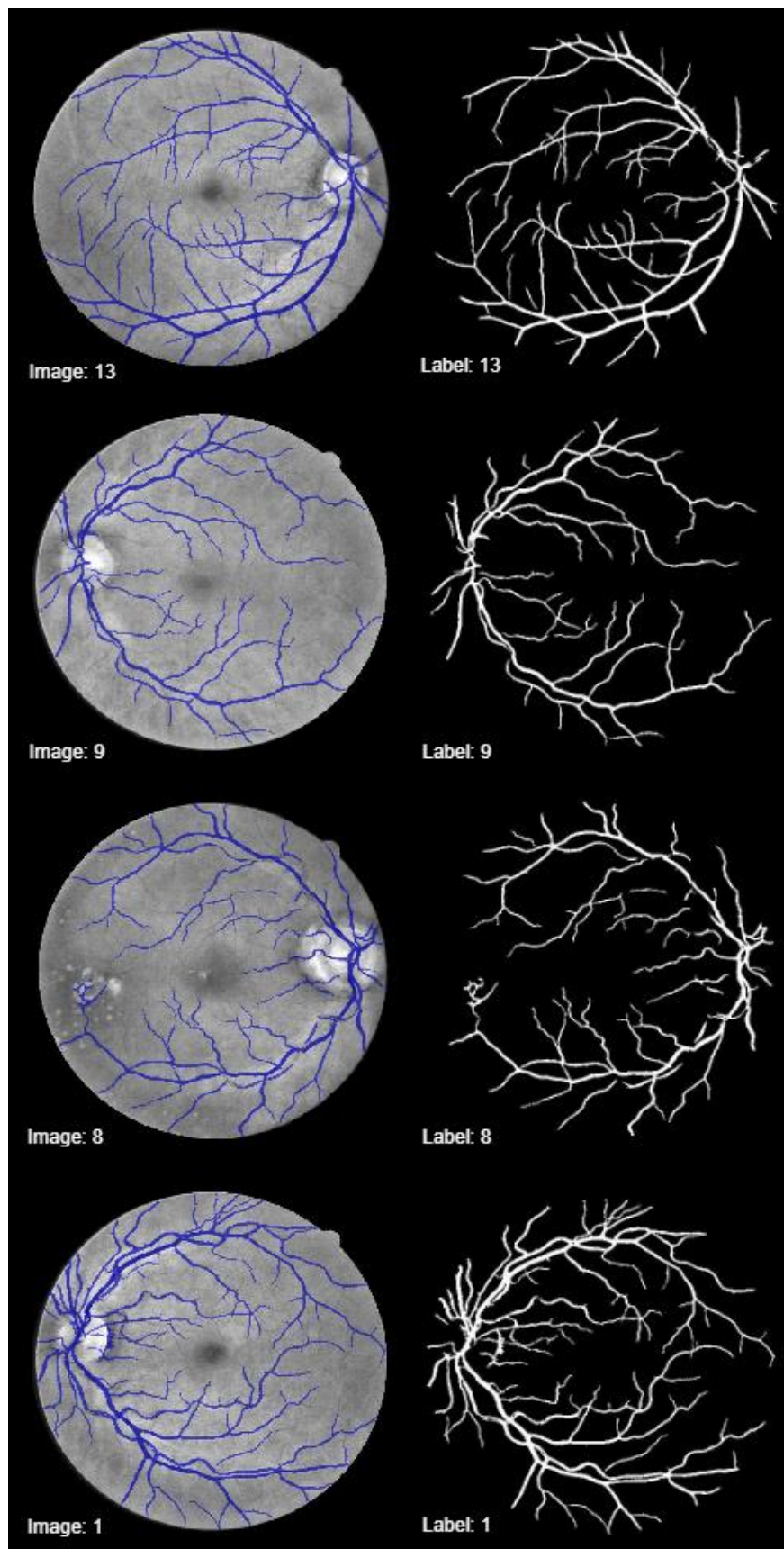
Εικόνα 146: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Μέγιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Highest
Accuracy 0.9735 Image: 5
Sensitivity 0.8136 Image: 1
Specificity 0.994 Image: 9
Precision 0.9196 Image: 18
F1 Score 0.8452 Image: 2
IoU 0.7319 Image: 2
AUC 0.8976 Image: 1
```

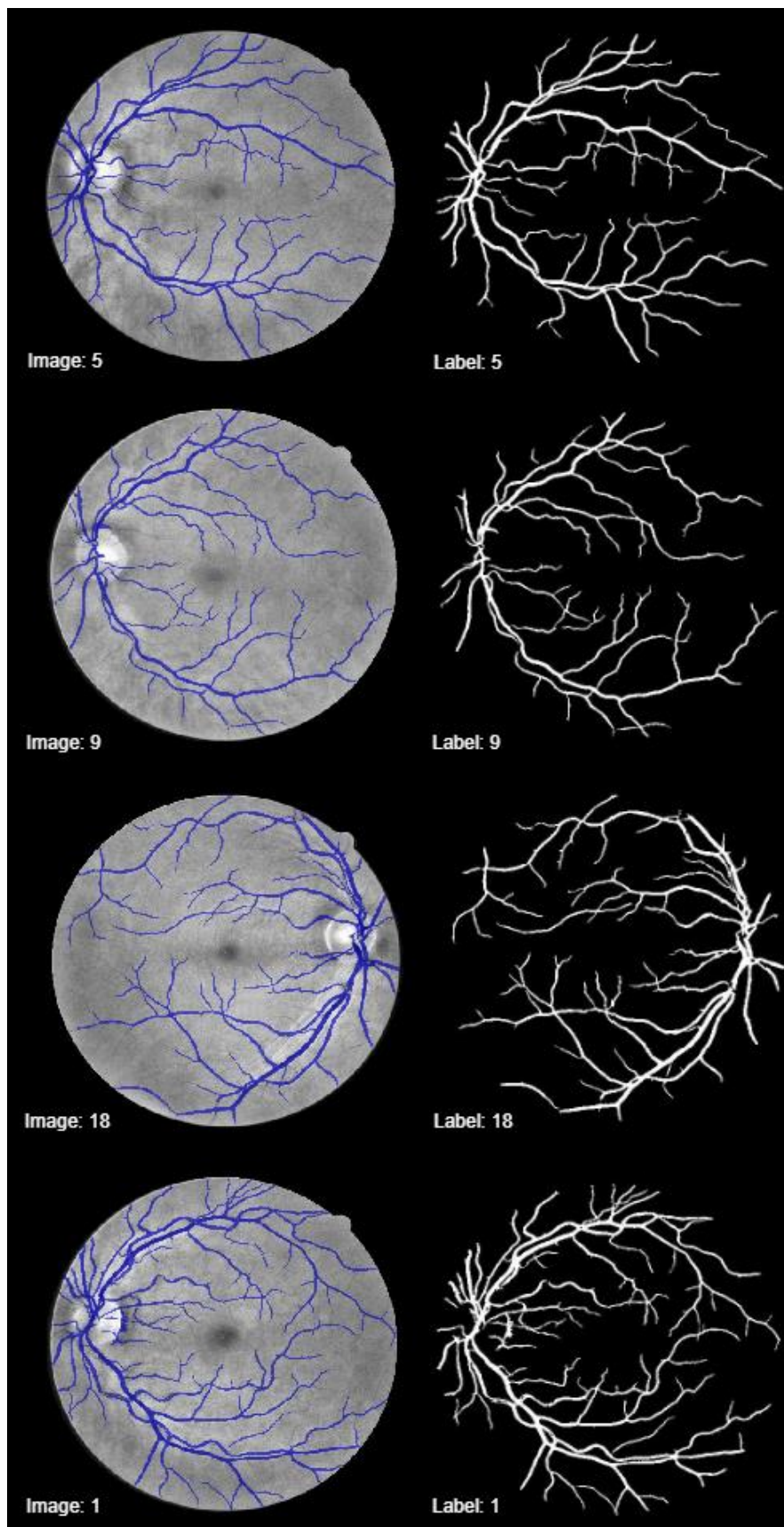
Εικόνα 147: Evaluation – Προβολή μέγιστων τιμών του κάθε δείκτη απόδοσης

Εικόνες με την ελάχιστη τιμή για κάθε δείκτη:



Εικόνα 148: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνες με τη μέγιστη τιμή για κάθε δείκτη:

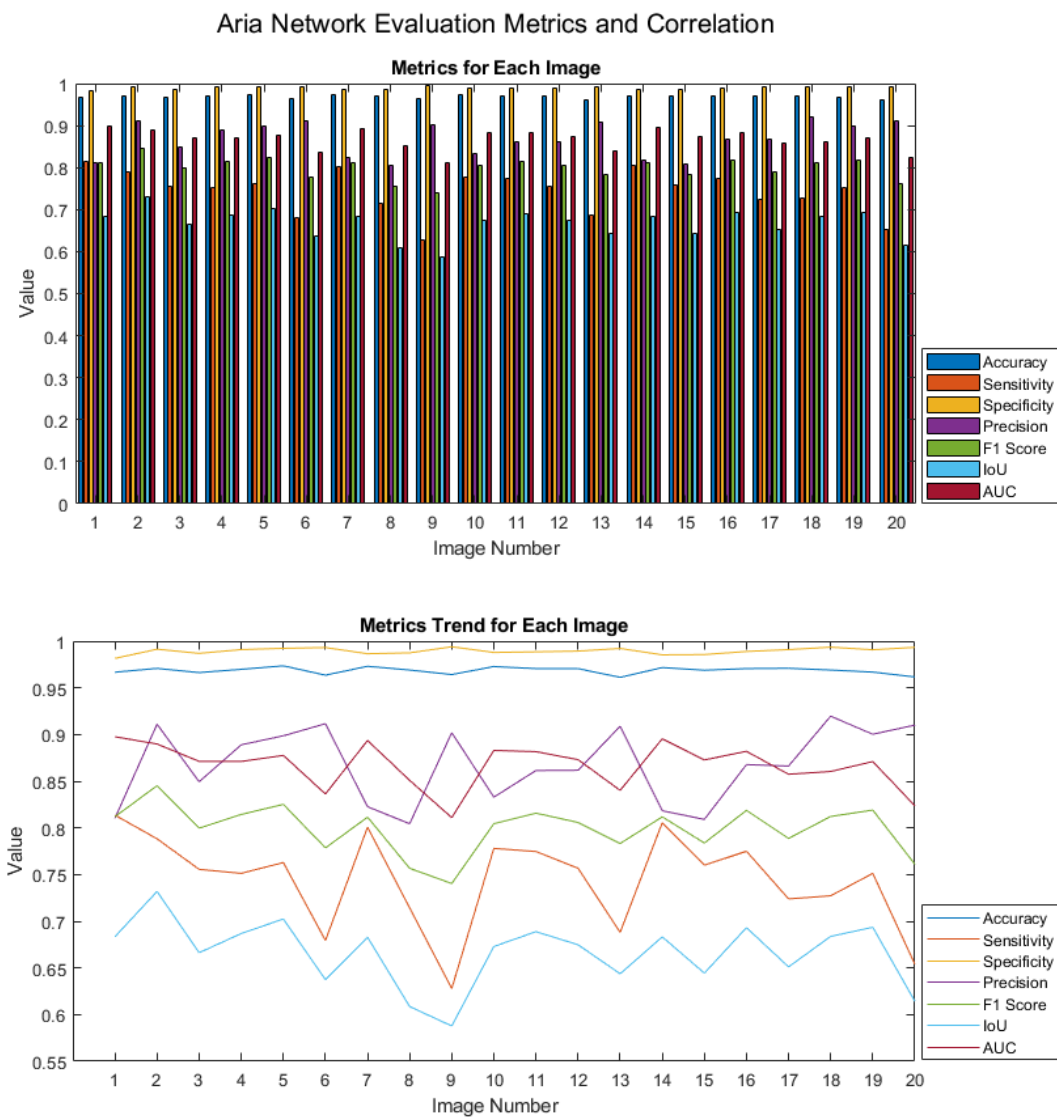


Εικόνα 149: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

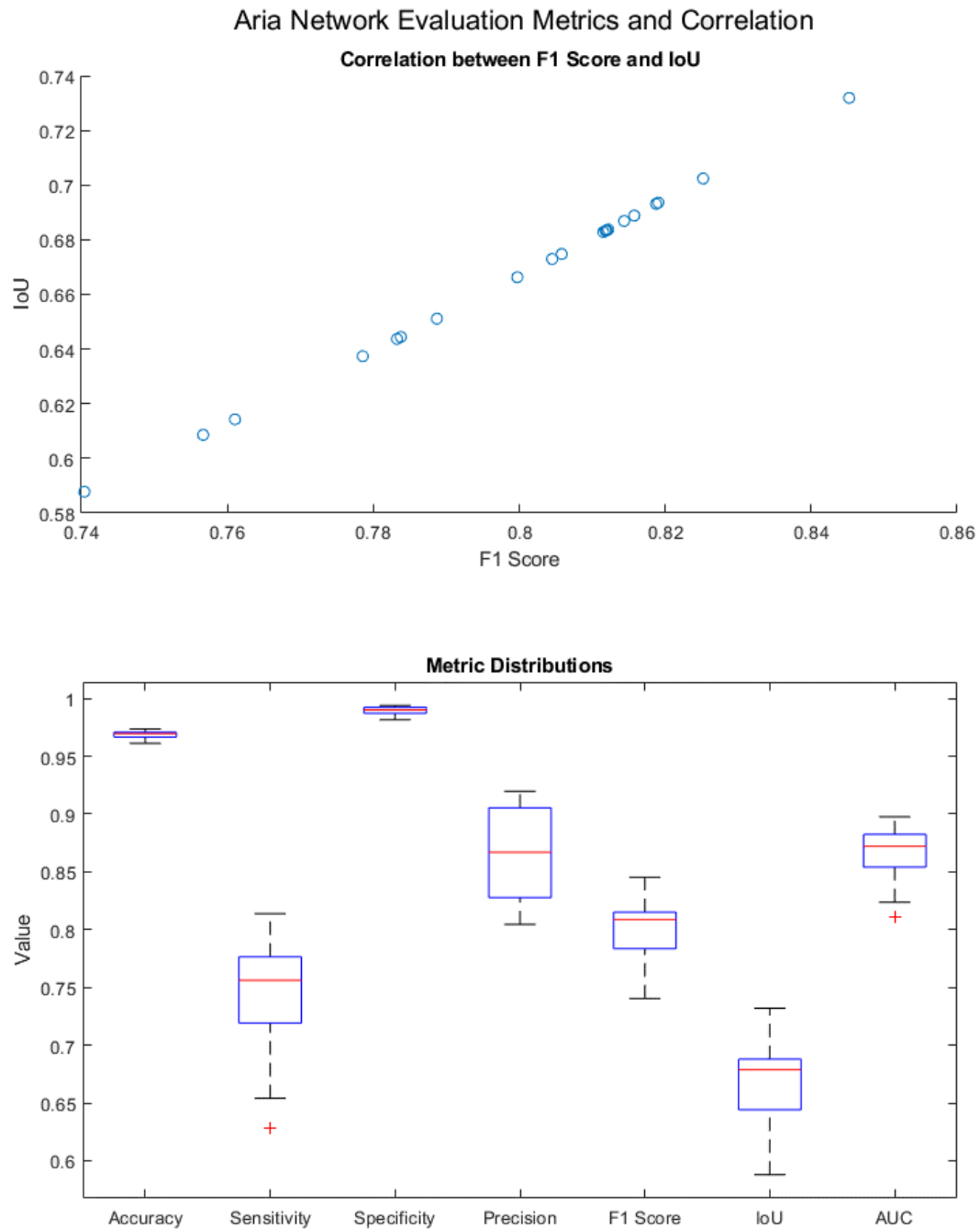
Με το τέλος της αξιολόγησης όλων των εικόνων και της παρουσίασης των αποτελεσμάτων στη καρτέλα Evaluation, σε ξεχωριστό παράθυρο εμφανίζονται και οι παρακάτω γραφικές παραστάσεις:

Γραφικές παραστάσεις και συσχέτιση:

(1/2)



Εικόνα 150: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

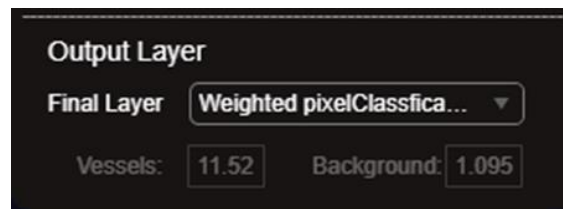


Εικόνα 151: Evaluation – Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

7.5 Αξιολόγηση με Weighted pixelClassificationLayer (2)

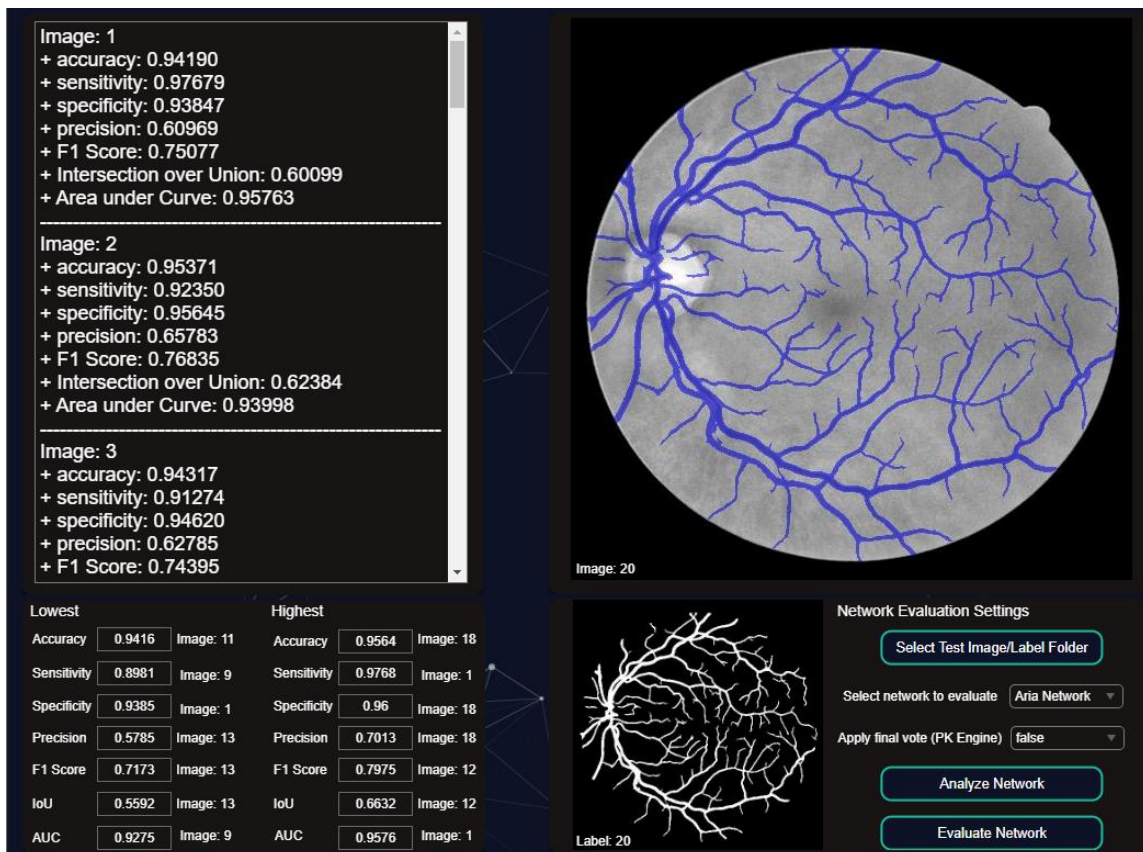
Evaluation – 5

Στη καρτέλα Evaluation, θα αξιολογήσουμε ένα δίκτυο που εκπαιδεύτηκε μέσω της εφαρμογής, χρησιμοποιώντας πάλι τις ίδιες παραμέτρους και το τελικό layer να είναι το **Weighted pixelClassificationLayer** [41]:



Εικόνα 152: Evaluation – Επιλογή Weighted pixelClassificationLayer ως final output layer

με εξαίρεση των αριθμό των epochs, που έχει τεθεί στα 300 και τον αριθμό των εικόνων για εκπαίδευση στα 15, της επαλήθευσης στα 5 και αξιολόγησης στα 20, με συνολικό χρόνο εκπαίδευσης μόλις ~2 ώρες και τα αποτελέσματα είναι συγκρίσιμα με σχετικές έρευνες [42] [43] [44] [45] [46] [47] [48] [49] [50]. Τα αποτελέσματα του δικτύου εμφανίζονται ως ακολούθως:



Εικόνα 153: Evaluation – Προβολή των αποτελεσμάτων αξιολόγησης του δικτύου Aria Network

Μέσος όρος δεικτών απόδοσης – αξιολόγησης:

```
=== Total Images: 20
= Mean accuracy: 0.94959
= Mean sensitivity: 0.93290
= Mean specificity: 0.95125
= Mean precision: 0.64908
= Mean F1 Score: 0.76457
= Mean Intersection over Union: 0.61950
= Mean Area under Curve: 0.94207
```

Εικόνα 154: Evaluation – Προβολή μέσου όρου των δεικτών απόδοσης

Ελάχιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Lowest
Accuracy 0.9416 Image: 11
Sensitivity 0.8981 Image: 9
Specificity 0.9385 Image: 1
Precision 0.5785 Image: 13
F1 Score 0.7173 Image: 13
IoU 0.5592 Image: 13
AUC 0.9275 Image: 9
```

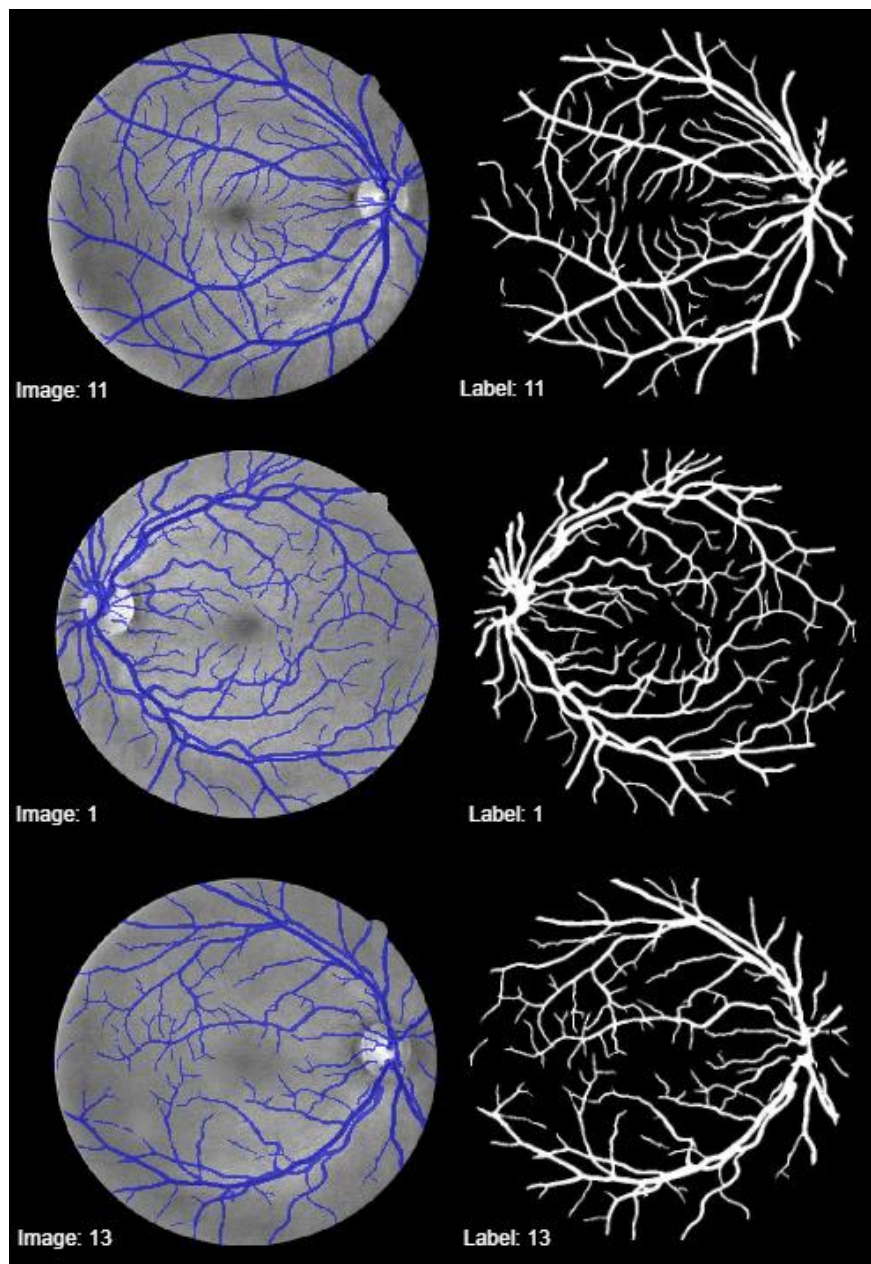
Εικόνα 155: Evaluation – Προβολή ελάχιστων τιμών του κάθε δείκτη απόδοσης

Μέγιστη τιμή των δεικτών απόδοσης – αξιολόγησης:

```
Highest
Accuracy 0.9564 Image: 18
Sensitivity 0.9768 Image: 1
Specificity 0.96 Image: 18
Precision 0.7013 Image: 18
F1 Score 0.7975 Image: 12
IoU 0.6632 Image: 12
AUC 0.9576 Image: 1
```

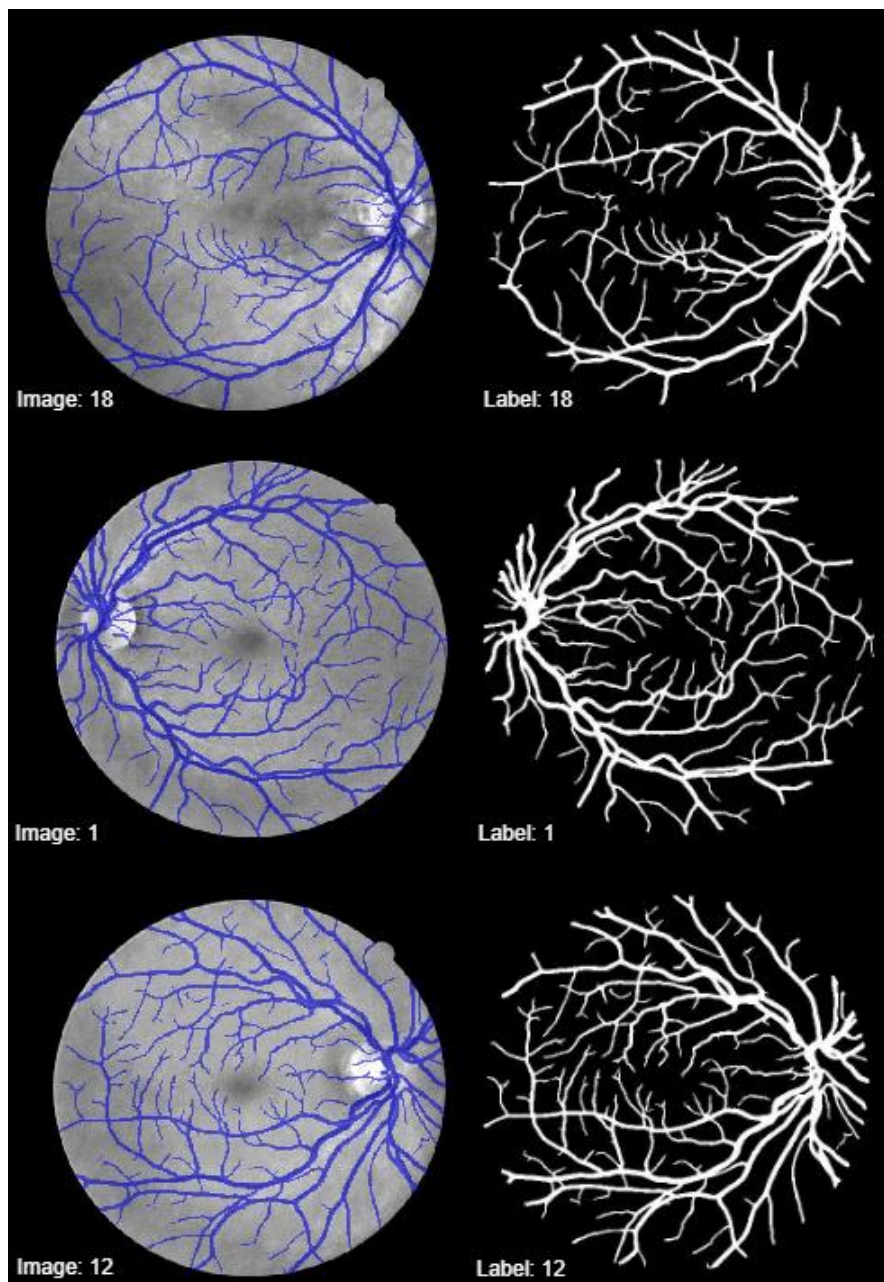
Εικόνα 156: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

Εικόνες με την ελάχιστη τιμή για κάθε δείκτη:



Εικόνα 157: Evaluation – Εικόνες με την ελάχιστη τιμή για κάθε δείκτη

Εικόνες με τη μέγιστη τιμή για κάθε δείκτη:

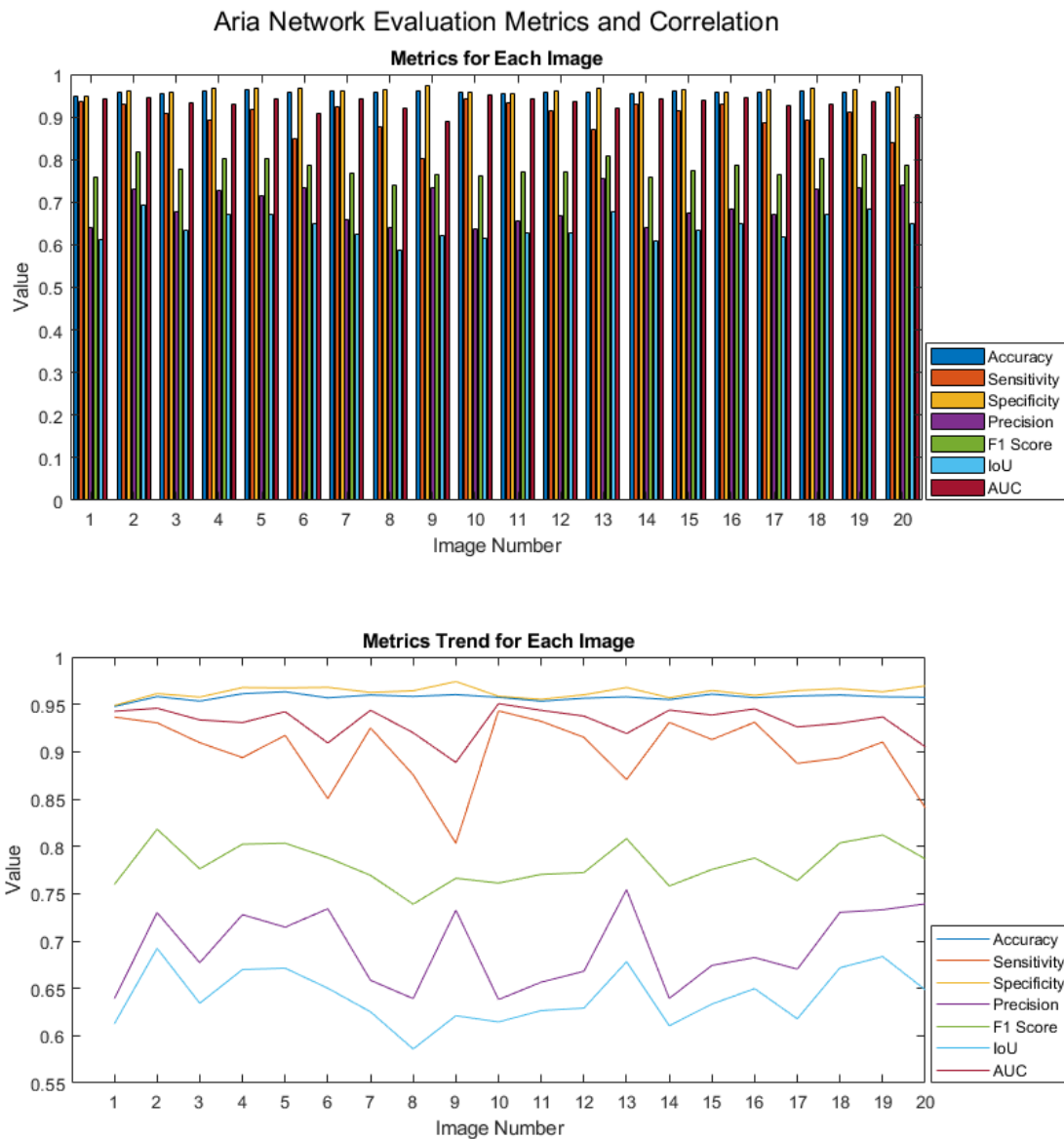


Εικόνα 158: Evaluation – Εικόνες με τη μέγιστη τιμή για κάθε δείκτη

Με το τέλος της αξιολόγησης όλων των εικόνων και της παρουσίασης των αποτελεσμάτων στη καρτέλα Evaluation, σε ξεχωριστό παράθυρο εμφανίζονται και οι παρακάτω γραφικές παραστάσεις:

Γραφικές παραστάσεις και συσχέτιση:

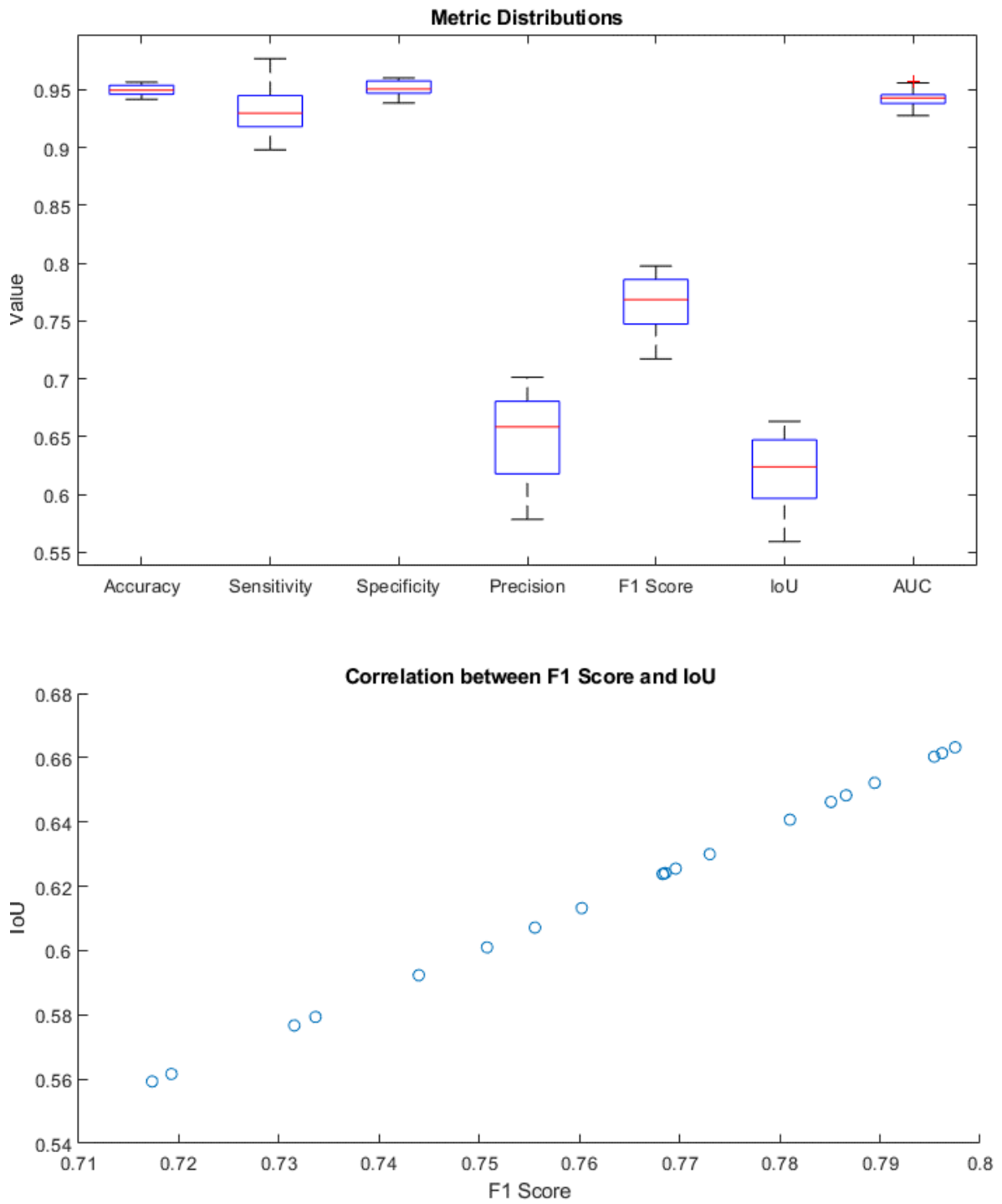
(1/2)



Εικόνα 159: Evaluation – Παρουσίαση γραφημάτων απόδοσης δεικτών και τάσεων

Γραφικές παραστάσεις και συσχέτιση:

(2/2)



Εικόνα 160: Παρουσίαση θηκογράμματος και συσχέτισης μεταξύ F1 και IoU

Κεφάλαιο 8: Τελικά συμπεράσματα

8.1 Συμπεράσματα

Μέσα από αυτήν τη διπλωματική εργασία, προέκυψε μια χρήσιμη εφαρμογή, η οποία αποτελεί ένα εργαλείο για ερευνητές, ιατρούς, ακαδημαϊκούς, και πάσης φύσεως ειδικούς ή μη, της Ιατρικής. Μέσω της εφαρμογής αυτής, το χρονοβόρο και σύνθετο έργο της επεξεργασίας, της δημιουργίας δικτύων U-Net, της εκπαίδευσης και της αξιολόγησης μοντέλων U-Net μετατρέπεται σε μια απλή και γρήγορη διαδικασία η οποία παρουσιάζει υψηλά αποτελέσματα τόσο σε κομμάτι επεξεργασίας αλλά και εκπαίδευσης του τελικού δικτύου.

Η εφαρμογή Aria αποτελεί ένα εργαλείο που προωθεί την ιατρική επιστήμη και την τεχνολογία, ενισχύοντας την δυνατότητα των επαγγελματιών της υγείας να αξιοποιούν προηγμένες τεχνικές ανάλυσης εικόνων. Μέσω της εύκολης προσβασιμότητας και της απλοποιημένης διαδικασίας, ανοίγει τον δρόμο για την περαιτέρω πρόοδο και ενασχόληση στο τομέα της οφθαλμολογίας.

8.2 Απλότητα και επιτάχυνση διαδικασιών

Ένα από τα κύρια χαρακτηριστικά πέραν της απλότητας της εφαρμογής που την κάνει εύκολη στη χρήση από όλους τους επιστημονικούς και ιατρικούς κλάδους, αξίζει να τονιστεί είναι η εξοικονόμηση χρόνου που προσφέρει η εφαρμογή. Η δημιουργία και εκπαίδευση ενός μοντέλου U-Net μπορεί να απαιτεί πολλές ώρες ή ακόμα και μέρες με την παραδοσιακή «προγραμματιστική» προσέγγιση, ειδικά στο στάδιο επιλογής των κατάλληλων φίλτρων – τεχνικών επεξεργασίας, και η εφαρμογή το μετατρέπει σε μία διαδικασία μερικών κλικ.

Μέσω της εφαρμογής, ο χρήστης επιλέγει τα σύνολα δεδομένων, εφαρμόζει προ-επεξεργασία, επιλέγει τον τύπο του δικτύου U-Net, εφαρμόζει data augmentation, εκπαιδεύει το μοντέλο και αξιολογεί την απόδοσή του, όλα αυτά σε ένα ολοκληρωμένο, λιτό περιβάλλον που απομακρύνει τον χρήστη από το φράγμα του προγραμματισμού και της πολυπλοκότητας των διάφορων τεχνικών, επιτρέποντας σε οποιονδήποτε να αξιοποιήσει την αξιοσημείωτη ισχύ των μοντέλων U-Net χωρίς την ανάγκη για τεχνικές γνώσεις.

8.3 Μελλοντικές Επεκτάσεις

Πέρα από την παρούσα διπλωματική εργασία, η εφαρμογή Aria διαθέτει ένα εξαιρετικό δυναμικό για μελλοντικές επεκτάσεις. Αν και ήδη προσφέρει αξιολόγηση απόδοσης μοντέλων, μπορεί να εμπλουτιστεί με περισσότερους δείκτες απόδοσης (μετρικά αποτελέσματα), γραφήματα και αναλυτικές αναφορές για την βαθύτερη κατανόηση της απόδοσης των μοντέλων.

Επιπλέον, η εφαρμογή μπορεί να προσφέρει περισσότερες τεχνικές – μεθόδους προ-επεξεργασίας μέσα από μελλοντικά ερευνητικά άρθρα και να υποστηρίξει περισσότερα δίκτυα – αρχιτεκτονικές βάσει της μελλοντικής ανάπτυξης, έρευνας και πειραμάτων από την επιστημονική κοινότητα. Ωστόσο, θα μπορούσε να διευρυνθεί το εύρος ιατρικών εφαρμογών πέραν της ανίχνευσης αγγείων του αμφιβληστροειδούς επιτρέποντας σε ερευνητές να δημιουργήσουν προηγμένα μοντέλα για διάφορες ιατρικές εργασίες και προβλήματα.

Βιβλιογραφία και Παραπομπές

- [1] Samantha Jacques, Barbara Christe, Chapter 2 - Healthcare technology basics, *Introduction to Clinical Engineering*, Academic Press, 2020, Pages 21-50, ISBN 9780128181034, <https://doi.org/10.1016/B978-0-12-818103-4.00002-8>.
- [2] Stivaros SM, Gledson A, Nenadic G, Zeng XJ, Keane J, Jackson A. Decision support systems for clinical radiological practice -- towards the next generation. *Br J Radiol*. 2010 Nov;83(995):904-14. doi: 10.1259/bjr/33620087. PMID: 20965900; PMCID: PMC3473729.
- [3] Mahabadi N, Al Khalili Y. Neuroanatomy, Retina. 2023 Aug 8. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. PMID: 31424894.
- [4] Retinal Diseases and VISION 2020. *Community Eye Health*. 2003;16(46):19-20. PMID: 17491850; PMCID: PMC1705858.
- [5] Beaudelaire Saha Tchinda, Daniel Tchiotsop, Michel Noubom, Valerie Louis-Dorr, Didier Wolf, Retinal blood vessels segmentation using classical edge detection filters and the neural network, *Informatics in Medicine Unlocked*, Volume 23, 2021, 100521, ISSN 2352-9148, <https://doi.org/10.1016/j.imu.2021.100521>.
- [6] Huang SY, Hsu WL, Hsu RJ, Liu DW. Fully Convolutional Network for the Semantic Segmentation of Medical Images: A Survey. *Diagnostics (Basel)*. 2022 Nov 11;12(11):2765. doi: 10.3390/diagnostics12112765. PMID: 36428824; PMCID: PMC9689961.
- [7] Sakshi Indolia, Anil Kumar Goswami, S.P. Mishra, Pooja Asopa, Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach, *Procedia Computer Science*, Volume 132, 2018, Pages 679-688, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2018.05.069>.
- [8] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9, 611–629 (2018). <https://doi.org/10.1007/s13244-018-0639-9>
- [9] Tang, Xikai & Rangraz, Esmaeel & Coudyzer, Walter & Bertels, Jeroen & Robben, David & Schramm, Georg & Deckers, Wies & Maleux, Geert & Baete, Kristof & Verslype, Chris & Gooding, Mark & Deroose, Christophe & Nuyts, Johan. (2020).

- [10] Sarki, R., Ahmed, K., Wang, H. et al. Image Preprocessing in Classification and Identification of Diabetic Eye Diseases. *Data Sci. Eng.* 6, 455–471 (2021).
<https://doi.org/10.1007/s41019-021-00167-z>
- [11] DRIVE, <https://drive.grand-challenge.org/>
- [12] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8, 53 (2021).
<https://doi.org/10.1186/s40537-021-00444-8>
- [13] Zhao Y, Liu Y, Wu X, Harding SP, Zheng Y. Retinal vessel segmentation: an efficient graph cut approach with retinex and local phase. *PLoS One*. 2015 Apr 1;10(4):e0122332. doi: 10.1371/journal.pone.0122332. Erratum in: *PLoS One*. 2015;10(4):e0127486. PMID: 25830353; PMCID: PMC4382050.
- [14] Purves D, Augustine GJ, Fitzpatrick D, et al., editors. *Neuroscience*. 2nd edition. Sunderland (MA): Sinauer Associates; 2001. *Anatomy of the Eye*. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK11120/>
- [15] Panwar N, Huang P, Lee J, Keane PA, Chuan TS, Richhariya A, Teoh S, Lim TH, Agrawal R. Fundus Photography in the 21st Century--A Review of Recent Technological Advances and Their Implications for Worldwide Healthcare. *Telemed J E Health*. 2016 Mar;22(3):198-208. doi: 10.1089/tmj.2015.0068. Epub 2015 Aug 26. PMID: 26308281; PMCID: PMC4790203.
- [16] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4), 115-133.
- [17] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210-229.
- [18] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
- [19] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.
- [20] Lloyd, S. P. (1957). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129-137.
- [21] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1), 9-44.

- [22] Hinton, G. E., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504-507.
- [23] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- [24] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: Jun. 15, 2021. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [28] Van Houdt, Greg & Mosquera, Carlos & Nápoles, Gonzalo. (2020). A Review on the Long Short-Term Memory Model. *Artificial Intelligence Review*. 53. 10.1007/s10462-020-09838-1.
- [29] Brinker TJ, Hekler A, Enk AH, Klode J, Hauschild A, Berking C, Schilling B, Haferkamp S, Schadendorf D, Holland-Letz T, Utikal JS, von Kalle C; Collaborators. Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task. *Eur J Cancer*. 2019 May;113:47-54. doi: 10.1016/j.ejca.2019.04.001. Epub 2019 Apr 10. PMID: 30981091.
- [30] Philomina Simon, Uma V, Deep Learning based Feature Extraction for Texture Classification, *Procedia Computer Science*, Volume 171, 2020, Pages 1680-1687, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2020.04.180>.
- [31] Sami Azam, Sidratul Montaha, Kayes Uddin Fahim, A.K.M. Rakibul Haque Rafid, Md. Saddam Hossain Mukta, Mirjam Jonkman, Using feature maps to unpack the CNN 'Black box' theory with two medical datasets of different modality, *Intelligent Systems with Applications*, Volume 18, 2023, 200233, ISSN 2667-3053, <https://doi.org/10.1016/j.iswa.2023.200233>.

- [32] Olaf Ronneberger, Philipp Fischer, Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", arXiv:1505.04597, <https://doi.org/10.48550/arXiv.1505.04597>
- [33] Wafa Baccouch, Sameh Oueslati, Basel Solaiman, Salam Labidi, A comparative study of CNN and U-Net performance for automatic segmentation of medical images: application to cardiac MRI, *Procedia Computer Science*, Volume 219, 2023, Pages 1089-1096, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2023.01.388>.
- [34] SegNet: Badrinarayanan, Vijay & Handa, Ankur & Cipolla, Roberto. (2015). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling.
- [35] Kiran Maharana, Surajit Mondal, Bhushankumar Nemade, A review: Data pre-processing and data augmentation techniques, *Global Transitions Proceedings*, Volume 3, Issue 1, 2022, Pages 91-99, ISSN 2666-285X, <https://doi.org/10.1016/j.glt.2022.04.020>.
- [36] Massimo Salvi, U. Rajendra Acharya, Filippo Molinari, Kristen M. Meiburger, The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis, *Computers in Biology and Medicine*, Volume 128, 2021, 104129, ISSN 0010-4825, <https://doi.org/10.1016/j.compbiomed.2020.104129>.
- [37] Hamwood J, Alonso-Caneiro D, Read SA, Vincent SJ, Collins MJ. Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers. *Biomed Opt Express*. 2018 Jun 11;9(7):3049-3066. doi: 10.1364/BOE.9.003049. PMID: 29984082; PMCID: PMC6033561.
- [38] Huang L, Qin J, Zhou Y, Zhu F, Liu L, Shao L. Normalization Techniques in Training DNNs: Methodology, Analysis and Application. *IEEE Trans Pattern Anal Mach Intell*. 2023 Aug;45(8):10173-10196. doi: 10.1109/TPAMI.2023.3250241. Epub 2023 Jun 30. PMID: 37027763.
- [39] Saleh MD, Eswaran C, Mueen A. An automated blood vessel segmentation algorithm using histogram equalization and automatic threshold selection. *J Digit Imaging*. 2011 Aug;24(4):564-72. doi: 10.1007/s10278-010-9302-9. PMID: 20524139; PMCID: PMC3138933.
- [40] Agnieszka Mikołajczyk and Michał Grochowski. "Data augmentation for improving deep learning in image classification problem". In: 2018 International Interdisciplinary PhD Workshop (IIPhDW) (2018), pp. 117–122.

[41] <https://mathworks.com>

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention, (Springer), pp. 234–241.

[43] Huadeng Wang, Guang Xu, Xipeng Pan, Zhenbing Liu, Ningning Tang, Rushi Lan, Xiaonan Luo, Attention-inception-based U-Net for retinal vessel segmentation with advanced residual, *Computers & Electrical Engineering*, Volume 98, 2022, 107670, ISSN 0045-7906, <https://doi.org/10.1016/j.compeleceng.2021.107670>.

[44] Wang, C., Zhao, Z. & Yu, Y. Fine retinal vessel segmentation by combining Nest U-net and patch-learning. *Soft Comput* **25**, 5519–5532 (2021).
<https://doi.org/10.1007/s00500-020-05552-w>

[45] Gu Z, Cheng J, Fu H, Zhou K, Hao H, Zhao Y, Zhang T, Gao S, Liu J. CE-Net: Context Encoder Network for 2D Medical Image Segmentation. *IEEE Trans Med Imaging*. 2019 Oct;38(10):2281-2292. doi: 10.1109/TMI.2019.2903562.

[46] P. Yin, R. Yuan, Y. Cheng, and Q. Wu, "Deep guidance network for biomedical image segmentation," *IEEE Access* **8**, 116106 (2020).

[47] B. Cai and L. Ma, "A Transformer-based Cascade Network with Boundary Enhancement Loss for Retinal Vessel Segmentation," 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 2022, pp. 4292-4298, doi: 10.1109/ICPR56361.2022.9956483.

[48] Zhang, H.; Qiu, Y.; Song, C.; Li, J. Landmark-Assisted Anatomy-Sensitive Retinal Vessel Segmentation Network. *Diagnostics* **2023**, *13*, 2260.
<https://doi.org/10.3390/diagnostics13132260>

[49] P. Yin, H. Cai, and Q. Wu, "Df-net: Deep fusion network for multi-source vessel segmentation," *Inf. Fusion* **78**, 199–208 (2022).

[50] Yong-fei Zhu, Xiang Xu, Xue-dian Zhang, and Min-shan Jiang, "CCS-UNet: a cross-channel spatial attention model for accurate retinal vessel segmentation," *Biomed. Opt. Express* **14**, 4739-4758 (2023)