ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών

Σχολή Θετικών Επιστημών, Τμήμα Φυσικής

Τομέας Πυρηνικής Φυσικής και Στοιχειωδών Σωματιδίων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Τεχνητού Νευρωνικού Δικτύου για την Κλινική Εκτίμηση Τομοσπινθηρογραφικών Απεικονίσεων του Εγκεφάλου

Μαζοκοπάκη Ηλιάνα

1110202000097

Επιβλέπων Καθηγητής: Ευστάθιος Στυλιάρης

Αθήνα, 2025

HELLENIC REPUBLIC
National and Kapodistrian University of Athens

School of Sciences, Physics Department

Section of Nuclear and Elementary Particle Physics

BACHELOR'S THESIS

Development of an Artificial Neural Network for the Clinical Assessment of Scintigraphic Brain Images

Mazokopaki Iliana

1110202000097

Thesis Supervisor: Professor Efstathios Stiliaris

Athens, 2025

Σημείωση χρήσης

Πρότυπο παραπομπής:

Μαζοκοπάκη Ηλιάνα (2025), «Ανάπτυξη Τεχνητού Νευρωνικού Δικτύου για την Κλινική Εκτίμηση Τομοσπινθηρογραφικών Απεικονίσεων του Εγκεφάλου». Πτυχιακή Εργασία. Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Φυσικής.

Usage Note

Suggested Citation Format:

Mazokopaki Iliana (2025), "Development of an Artificial Neural Network for the Clinical Assessment of Scintigraphic Brain Images". Bachelor's thesis. National and Kapodistrian University of Athens, Physics Department.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κύριο Στυλιάρη Ευστάθιο, Καθηγητή του Τμήματος Φυσικής του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών, επιβλέποντα της εργασίας αυτής, τόσο για το πολύ ενδιαφέρον θέμα που μου προσέφερε, όσο και για τον χρόνο και την καθοδήγησή του στην πορεία αυτής της εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω τους ανθρώπους μου που ήταν εκεί για εμένα σε όλη τη διάρκεια και σε κάθε συναίσθημα στηρίζοντάς με, ο καθένας με τον μοναδικό του τρόπο.

Πίνακας περιεχομένων

Περίληψη

Η παρούσα εργασία αποσκοπεί στην προσέγγιση της απεικονιστικής ιατρικής με τη χρήση τεχνητών νευρωνικών δικτύων. Η ιατρική φυσική και ειδικότερα η απεικονιστική ιατρική διαδραματίζουν καίριο ρόλο στη σύγχρονη ιατρική, ενισχύοντας τη διαγνωστική και θεραπευτική διαδικασία μέσω προηγμένων τεχνολογιών και ως εκ τούτου αποτελούν ένα χρήσιμο εργαλείο στην έγκαιρη διάγνωση και την παρακολούθηση της εξέλιξης νευροεκφυλιστικών ασθενειών του εγκεφάλου όπως η νόσος Alzheimer.

Η τομογραφία εκπομπής ποζιτρονίων PET και η τομογραφία μονοφωτονικής εκπομπής SPECT αποτελούν τις κύριες εφαρμογές της πυρηνικής ιατρικής που χρησιμοποιούν ραδιοϊχνηθέτες για την αξιολόγηση βιολογικών διεργασιών σε έμβιους οργανισμούς και αξιοποιούνται τόσο στη διάγνωση όσο και στην εκτίμηση της πρόγνωσης της νόσου Alzheimer. Μία τομοσπινθηρογραφική τεχνική που βασίζεται στην απεικονιστική SPECT αποτελεί το σπινθηρογράφημα εγκεφάλου (DaTSCAN), το οποίο ανιχνεύει τη δραστηριότητα του μεταφορέα της ντοπαμίνης στον εγκέφαλο και χρησιμοποιείται στη διάκριση των ασθενών με άνοια με σώματα Lewy από τους ασθενείς με νόσο Alzheimer και στη διάκριση των ασθενών με νόσο Parkinson από αυτούς με νόσο Alzheimer. Μία φυσιολογική εικόνα DaTSCAN περιλαμβάνει ομοιόμορφη απορρόφηση του ραδιοϊχνηθέτη από τους δύο εγκεφαλικούς λοβούς οι οποίοι εμφανίζονται με ελλειπτικό σχήμα. Στην περίπτωση ασθενών με νόσο Alzheimer, η εικόνα DaTSCAN παρουσιάζει ήπιες αλλοιώσεις καθώς η απορρόφηση του ραδιοφαρμάκου δεν επηρεάζεται σε μεγάλο βαθμό από τη νόσο, σε αντίθεση με τις άλλες νευροεκφυλιστικές νόσους στις οποίες οι αλλοιώσεις είναι πιο έντονες. Σε κάθε περίπτωση, η διάγνωση της αντίστοιχης νόσου γίνεται από τον εγκεκριμένο νευρολόγο.

Αρχικά, σε μία πρώτη προσέγγιση των νευρωνικών δικτύων, παρουσιάζονται απλά νευρωνικά δίκτυα που κατασκευάζουν το απαιτούμενο δείγμα εικόνων διαστάσεων 128×128 και στη συνέχεια τις ταξινομούν ως προς συγκεκριμένα χαρακτηριστικά που αποτελούν το ζητούμενο κάθε προβλήματος. Τα δείγματα εικόνων περιλαμβάνουν απλά γεωμετρικά σχήματα όπως τα τετράγωνα και οι κύκλοι και παρουσιάζουν ορισμένες παραλλαγές ως προς τη θέση, το μέγεθος, την πολλαπλότητα και την αλληλοεπικάλυψη αυτών. Με βάση τα συγκεκριμένα χαρακτηριστικά ζητείται η ταξινόμηση των εικόνων σε κατηγορίες ανάλογα με το ζητούμενο. Με τα παραπάνω απλά νευρωνικά δίκτυα, μελετάται η διαδικασία εκπαίδευσης αυτών και διαπιστώνεται πως θέτοντας τις κατάλληλες συνθήκες στην εκπαίδευση και τις κατάλληλες παραμέτρους στην αρχιτεκτονική του δικτύου, τα νευρωνικά δίκτυα είναι σε θέση να γενικεύουν σε άγνωστα για αυτά δεδομένα και να κατηγοριοποιούν ορθά τις εικόνες, με τα ποσοστά της ακρίβειας να κυμαίνονται από 82.5% έως και 100%.

Στο πλαίσιο της προσέγγισης της απεικονιστικής ιατρικής με τη χρήση τεχνητής νοημοσύνης, αναπτύσσεται ένα συνελικτικό νευρωνικό δίκτυο που δέχεται ως δεδομένα εισόδου ομοιώματα τομοσπινθηρογραφικών εικόνων εγκεφάλου και έχει ως έξοδο την ποσοτικοποιημένη απορρόφηση του ραδιοφαρμάκου από τους δύο εγκεφαλικούς λοβούς. Επίσης, αναπτύσσεται άλλο ένα συνελικτικό νευρωνικό δίκτυο με τα ίδια δεδομένα εισόδου και έχει ως δεδομένα εξόδου τις απορροφούμενες εντάσεις κάθε εγκεφαλικού λοβού ξεχωριστά όπως και του υποβάθρου. Για την κατασκευή των ομοιωμάτων αξιοποιείται το πρόγραμμα Simulix3x το οποίο δημιουργεί το απαιτούμενο δείγμα εικόνων διαστάσεων 128×128 που περιέχουν το ελλειψοειδές της κεφαλής και

τα δύο ελλειψοειδή των λοβών που παράγονται με μία στοχαστική μετατόπιση ως προς τη θέση και την ένταση της απορροφούμενης ακτινοβολίας τόσο από τους λοβούς, όσο και από το υπόβαθρο. Με το συγκεκριμένο λογισμικό παρήχθησαν συνολικά 3000 ομοιώματα, εκ των οποίων το 80% χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου, ενώ το υπόλοιπο 20% αξιοποιήθηκε στον έλεγχο της απόδοσης και της σύγκλισης του νευρωνικού δικτύου.

Τα δύο παραπάνω συνελικτικά νευρωνικά δίκτυα παρουσιάζουν τυπικές αποκλίσεις $\sigma_1 = 0{,}014$ για την ποσοτικοποίηση της ανομοιογενούς απορρόφησης του ραδιοϊχνηθέτη και $\sigma_2 = 0{,}078$ για την πρόβλεψη της απορροφούμενης έντασης κάθε εγκεφαλικού λοβού. Όσον αφορά την πρόβλεψη για την απορρόφηση του υποβάθρου, η τυπική απόκλιση είναι ίση με $\sigma_3 = 0{,}025$. Οι παραπάνω τυπικές αποκλίσεις όπως και η σύγκλιση των συνελικτικών νευρωνικών δικτύων υποδεικνύουν πως αυτά τα τεχνητά νευρωνικά δίκτυα μπορούν να εφαρμοστούν και σε πραγματικά ιατρικά δεδομένα.

Τέλος, γίνεται η αξιολόγηση των νευρωνικών δικτύων σε δύο πραγματικές ιατρικές εικόνες DaTSCAN και υπολογίζεται η ποσοστιαία απόκλιση των προβλεπόμενων τιμών σε σύγκριση με τις τιμές που προκύπτουν από τις τομοσπινθηρογραφικές εικόνες, η οποία είναι της τάξης του 4.6% για τις απορροφούμενες εντάσεις των δύο εγκεφαλικών λοβών. Καταληκτικά, θεωρώντας πως η συγκεκριμένη απόκλιση βρίσκεται εντός τα πλαίσια του αποδεκτού για ιατρικές εφαρμογές, υπάρχει η δυνατότητα χρήσης του συνελικτικού νευρωνικού δικτύου επικουρικά στη διάγνωση και στη λήψη αποφάσεων από τον ειδικό θεράποντα ιατρό.

Summary

This Bachelor's thesis aims to approach medical imaging through the use of artificial neural networks. Medical physics and especially medical imaging plays a crucial role in modern medicine, reinforcing diagnostic and therapeutic process through advanced technologies and are therefore, a useful tool in the early diagnosis and monitoring of the progress of neurodegenerative brain diseases such as Alzheimer's disease.

Positron emission tomography (PET) and single photon emission tomography (SPECT) are the main applications of nuclear medicine that use radiotracers to evaluate biological processes in living organisms and are both used in the diagnosis and assessment of the prognosis of Alzheimer's disease. One tomoscintigraphic technique based on SPECT imaging is brain scintigraphy (DaTSCAN), which detects dopamine transporter activity in the brain and is used to distinguish patients with dementia with Lewy bodies from patients with Alzheimer's disease and to distinguish patients with Parkinson's disease from those with Alzheimer's disease. A normal DaTSCAN image includes equal absorption of the radiotracer by the two brain lobes which are displayed in an elliptical shape. In the case of patients with Alzheimer's disease, the DaTSCAN image shows mild alterations as the uptake of the radiopharmaceutical is not greatly affected by the disease, in contrast to other neurodegenerative diseases in which the alterations are more intense. In any case, the diagnosis of the respective disease is made by the registered neurologist.

As an initial approach to neural networks, simple neural networks are being presented that generate the required sample of images with dimensions of 128×128 and then classify them according to specific features that are the the task of each problem. The samples of images include simple geometric shapes such as squares and circles and display some variations in their position, size, multiplicity and overlap. Based on these characteristics, the image classification into categories is requested according to the predefined task. Through the mentioned simple neural networks, the process of training is being studied and it is found that by setting the appropriate conditions in training and the appropriate parameters in the network architecture, neural networks are able to generalize to unknown data and correctly categorize the images, with accuracy rates ranging from 82.5% up to 100%.

In the scope of approaching medical imaging through the use of artificial intelligence, a convolutional neural network is being developed that accepts as input data phantoms of tomoscintigraphic brain images and has as output the quantified absorption of the radiopharmaceutical by the two brain lobes. Another convolutional neural network is also being developed with the same input data and has as output the absorbed intensities od each brain lobe seperately as well as the absorbed intensity of the background. For the construction of the phantoms, the Simulix3x program is used to generate the required sample images with dimensions of 128×128, containing the head ellipsoid and the two lobe ellipsoids with a stochastic shift in the position and intensity of the absorbed radiation from both the lobes and the background. A total of 3000 phantoms were generated with this software, of which 80% were used for model training, while the remaining 20% were used to monitor the performance and convergence of the neural network.

The two convolutional neural networks mentioned above present standard deviations of $\sigma_1 = 0{,}014$ for the quantified absorption of the radiopharmaceutical by the two brain lobes and $\sigma_2 = 0{,}078$ for the prediction of the absorbed intensity of each brain lobe. Regarding to the prediction for the absorbed intensity of the background, the standard deviation is equal to $\sigma_3 = 0{,}025$. These standard deviations as well as the convergence of the convolutional neural networks indicate that these artificial neural networks can be applied to real medical data.

Finally, the neural networks are evaluated on two real DaTSCAN medical images and the percentage deviation of the predicted values compared to the values obtained from the tomoscintigraphic images is calculated, which is on the order of 4.6% for the absorbed intensities of the two cerebral lobes. In conclusion, considering that this deviation is within the acceptable range for medical applications, it is possible to use the convolutional neural network as an adjunct in diagnosis and in decision making by the treating medical specialist.

1. Νευρωνικά Δίκτυα

1.1 Εισαγωγή

Στην εποχή της τεχνολογίας, η τεχνητή νοημοσύνη λαμβάνει ολοένα και περισσότερο χώρο στην καθημερινότητα του ανθρώπου, από απλές εφαρμογές όπως η πρόβλεψη του καιρού, σε πιο σύνθετες όπως η ανάλυση ιατρικών εικόνων. Ο τομέας της ιατρικής και πιο συγκεκριμένα η απεικονιστική ιατρική διαθέτει πληθώρα ακατέργαστων δεδομένων τα οποία είναι δύσκολο να γίνουν διαχειρίσιμα με συμβατικές μεθόδους όπως η επεξεργασία εικόνων και οι στατιστικές μέθοδοι. Τόσο η ραγδαία αύξηση στο μέγεθος των δεδομένων όσο και η αύξηση της ανάλυσης των εικόνων οδήγησαν στην απαίτηση διαχείρισης αυτών με τη χρήση προηγμένων τεχνολογιών όπως η τεχνητή νοημοσύνη.

Η συνεχής εξέλιξη της τεχνητής νοημοσύνης και η χρήση αλγορίθμων μηχανικής και ειδικότερα βαθιάς μάθησης λειτουργεί ως πολύτιμος αρωγός της απεικονιστικής ιατρικής, καθιστώντας την πιο αποτελεσματική και ακριβή, υποστηρίζοντας την έγκαιρη διάγνωση, ανάλυση και θεραπευτική προσέγγιση ασθενειών, ενισχύοντας τους επαγγελματίες υγείας στη λήψη τεκμηριωμένων αποφάσεων.

1.2 Τι είναι τα Νευρωνικά Δίκτυα (Neural Networks – NN)

Οι ζώντες οργανισμοί, από τους πιο απλούς μέχρι τον άνθρωπο, έχουν ένα νευρικό σύστημα, το οποίο είναι υπεύθυνο για μια πληθώρα διεργασιών, όπως είναι η επαφή με τον εξωτερικό κόσμο, η μάθηση, η μνήμη, κ.α. Το νευρικό σύστημα των οργανισμών αποτελείται από πολλά νευρωνικά δίκτυα τα οποία είναι εξειδικευμένα στην εκτέλεση αυτών των διεργασιών.

Η κεντρική μονάδα του νευρικού συστήματος είναι ο εγκέφαλος, ο οποίος επίσης αποτελείται από νευρωνικά δίκτυα. Κάθε νευρωνικό δίκτυο αποτελείται από ένα μεγάλο αριθμό μονάδων, που λέγονται νευρώνες (neurons). Οι νευρώνες συνεχώς και ασταμάτητα επεξεργάζονται πληροφορίες, παίρνοντας και στέλνοντας ηλεκτρικά σήματα σε άλλους νευρώνες. Ο ανθρώπινος εγκέφαλος αποτελείται από έναν πολύ μεγάλο αριθμό νευρώνων της τάξης του 10^{10}.

Ο ρόλος του νευρώνα σε ένα νευρωνικό δίκτυο είναι να λαμβάνει όλα τα σήματα που έρχονται από άλλους νευρώνες, να τα επεξεργάζεται με κατάλληλο τρόπο και να μεταδίδει το επεξεργασμένο σήμα στους επόμενους νευρώνες. Με αυτόν τον τρόπο, κάθε σήμα διαδίδεται μέσω ενός τεράστιου αριθμού νευρώνων. Τα σήματα που επεξεργάζεται ένας νευρώνας είναι ηλεκτρικής μορφής, της τάξης μερικών $mVolt$.

Οι διεργασίες που επιτελούνται από τα βιολογικά νευρωνικά δίκτυα στους ζώντες οργανισμούς είναι πολύ περίπλοκες, αλλά και απαραίτητες στην καθημερινή ζωή του ανθρώπου. Μερικές από αυτές είναι εργασίες ρουτίνας, τις οποίες ο ανθρώπινος εγκέφαλος εκτελεί με ελάχιστη ή μηδαμινή προσπάθεια, όπως για παράδειγμα η αναγνώριση μιας εικόνας. Το ερώτημα που προκύπτει λοιπόν είναι: Μπορούν οι ηλεκτρονικοί υπολογιστές να κάνουν αυτά που κάνει το ανθρώπινο μυαλό;

1.3 Τι είναι τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks – ANN)

Η απάντηση στην παραπάνω ερώτηση δίνεται μέσω των τεχνητών νευρωνικών δικτύων (Artificial Neural Networks - ANN). Τα δίκτυα αυτά αποτελούν πρότυπα μοντέλα του ανθρώπινου νευρωνικού συστήματος, και περιέχουν όλα τα γνωστά χαρακτηριστικά τα οποία θα μπορούσαν από μόνα τους να επιτελέσουν τις εργασίες αυτές, με τον ίδιο τρόπο που γίνονται στα βιολογικά νευρωνικά δίκτυα.

Η βασική τους διαφορά από τα βιολογικά δίκτυα είναι ότι τα τεχνητά νευρωνικά δίκτυα μαθαίνουν με την εξάσκηση και την εμπειρία, όπως ακριβώς και οι άνθρωποι, αλλά διαφέρουν στο ότι δεν αναπτύσσουν γνώση αυθόρμητα, αλλά ακολουθούν ορισμένους προκαθορισμένους κανόνες, που είναι χαρακτηριστικό των υπολογιστών.

Τα τεχνητά νευρωνικά δίκτυα είναι ικανά να συλλάβουν τις εξαιρετικά μη γραμμικές συσχετίσεις μεταξύ των μεταβλητών πρόβλεψης και του αποτελέσματος για χαρτογράφηση συμπεριφοράς, τον υπολογισμό συναρτήσεων, την ταξινόμηση και την επεξεργασία αναγνώρισης προτύπων *([Abu 23])*. Επίσης, προσφέρουν λύσεις σε προβλήματα που σχετίζονται με τον ανθρώπινο παράγοντα, όπως η αναγνώριση ομιλίας, εικόνας και κειμένου.

Αναλυτικότερα για κάποιους τομείς έχουν αναπτυχθεί τα παρακάτω μοντέλα:

Τομέας	**Μοντέλο**
Βιολογία	Μοντέλα για την όραση
Ιατρική	Ανάγνωση και ανάλυση των ακτίνων Χ
Χρηματοοικονομικά	Ανάλυση επικινδυνότητας δανείων
Βιομηχανία	Έλεγχος στην γραμμή παραγωγής
Περιβάλλον	Πρόβλεψη καιρού

Table 1: Τομείς ανθρώπινης δράσης και ενδεικτικά νευρωνικά δίκτυα που έχουν αναπτυχθεί.

1.4 Αρχές Λειτουργίας ANN

Ένα σύστημα τεχνητής νοημοσύνης πρέπει να είναι ικανό να κάνει τρία πράγματα:

1. Να αποθηκεύει γνώση
2. Να εφαρμόζει την αποθηκευμένη γνώση
3. Να αποκτά νέα γνώση μέσω εμπειρίας

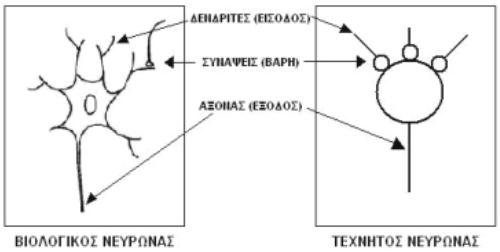
Τα ANN πραγματοποιούν δύο βασικές λειτουργίες: Τη μάθηση – εκπαίδευση (learning) και την ανάκληση (recall).

Πιο συγκεκριμένα, κάθε δίκτυο δέχεται ορισμένες εισόδους (input) και δίνει ορισμένες εξόδους (output). Η εκπαίδευση γίνεται με το να παρουσιαστεί μία ομάδα από πρότυπα δείγματα στο δίκτυο, αντιπροσωπευτικά ή παρόμοια με αυτά που είναι επιθυμητό να μάθει το δίκτυο. Πρακτικά, αυτό σημαίνει ότι δίδονται στο δίκτυο ως είσοδος κάποια πρότυπα δείγματα για τα οποία είναι γνωστή ποια πρέπει να είναι η έξοδος στο δίκτυο, δηλαδή είναι γνωστή η απάντηση που πρέπει να δίνει το δίκτυο στα πρότυπα που του παρουσιάζονται. Ουσιαστικά, είναι σαν να δίνεται στο δίκτυο και η ερώτηση και η απάντηση που αντιστοιχεί.

Το δίκτυο με τα δεδομένα αυτά τροποποιεί την εσωτερική του δομή, ώστε να κάνει την ίδια αντιστοιχία που δόθηκε από τον χειριστή. Ακολούθως, αφού έχει βρει τη σωστή εσωτερική δομή, τότε θα μπορεί να λύνει και άλλα ανάλογα προβλήματα ίδιας φύσης και παρόμοιων χαρακτηριστικών με αυτά της εκπαίδευσης χωρίς να έχει εκπαιδευτεί εκ των προτέρων σε αυτά, λαμβάνοντας το διάνυσμα εισόδου και παράγοντας το αντίστοιχο διάνυσμα εξόδου.

Κατά την εκπαίδευση του δικτύου, το σύνολο των δεδομένων χωρίζεται σε υποσύνολα, για παράδειγμα το 70% των συνολικών δεδομένων ορίζεται να είναι το σετ εκπαίδευσης (training set) το οποίο χρησιμοποιείται για την εκμάθηση των προτύπων και των σχέσεων που τα συνδέει από το νευρωνικό δίκτυο.

Ανάκληση είναι η διαδικασία υπολογισμού ενός διανύσματος εξόδου για συγκεκριμένο διάνυσμα εισόδου με βάση τα βάρη που έχουν καθοριστεί από τη διαδικασία μάθησης. Κατά την ανάκληση, το νευρωνικό δίκτυο χρησιμοποιείται για πρόβλεψη ή ταξινόμηση των δεδομένων χωρίς να απαιτείται η εκ νέου εκπαίδευση του μοντέλου. Έτσι πραγματοποιείται η επαλήθευση και η εφαρμογή του μοντέλου, για παράδειγμα στο υπόλοιπο 30% των συνολικών δεδομένων, το οποίο ονομάζεται σετ επαλήθευσης (validation set) και το νευρωνικό δίκτυο αξιολογείται στην εκτέλεση της ζητούμενης διαδικασίας.



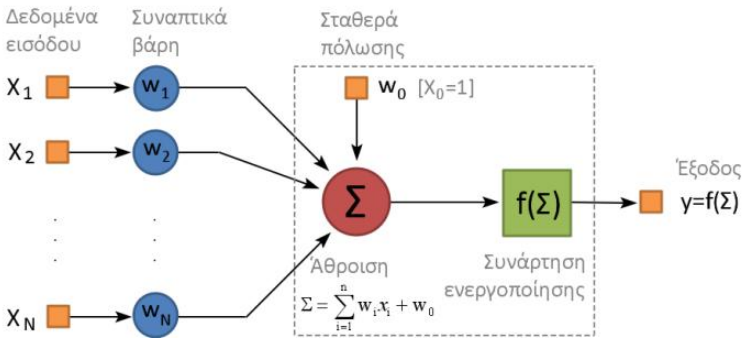
Εικόνα 1: Σύγκριση βιολογικού και τεχνητού νευρώνα. Διακρίνονται οι είσοδοι, τα βάρη και οι έξοδοι

1.5 Εκπαίδευση και διαδικασία μάθησης ΝΝ

Κάθε νευρώνας του νευρωνικού δικτύου έχει έναν αριθμό σημάτων που έρχονται ως είσοδος σε αυτόν, έχει μερικές καταστάσεις στις οποίες μπορεί να βρεθεί και έχει μία μόνο έξοδο η οποία είναι συνάρτηση των σημάτων εισόδου.

Κάθε είσοδος έχει μία δική της τιμή βάρους w_i η οποία υποδηλώνει την στενή σύνδεση των δύο νευρώνων που συνδέονται με το συγκεκριμένο βάρος και τον τρόπο σύνδεσης αυτών. Η τιμή αυτή βρίσκεται στο διάστημα $[-1, 1]$.

Όταν ένας νευρώνας ενεργοποιείται, υπολογίζει μία συνάρτηση από όλα τα δεδομένα που έχει και συγκρίνει την τιμή της συνάρτησης αυτής με μία τιμή κατωφλίου που είναι χαρακτηριστική του συγκεκριμένου νευρώνα. Αν η τιμή της συνάρτησης είναι μεγαλύτερη από την τιμή κατωφλίου, τότε ο νευρώνας υπολογίζει την έξοδο, την οποία προωθεί ως είσοδο στους επόμενους νευρώνες. Κατά τη διάρκεια της εκπαίδευσης αλλάζουν μόνο οι τιμές των βαρών w_i των συνδέσεων των νευρώνων, αλλαγές οι οποίες εξαρτώνται από τη μέθοδο που χρησιμοποιείται στον αλγόριθμο.



Εικόνα 2: Δομή ενός τεχνητού νευρώνα: Τα δεδομένα εισόδου σταθμίζονται με τα συναπτικά βάρη, αθροίζονται με τη σταθερά πόλωσης και προωθούνται στην συνάρτηση ενεργοποίησης για την εξαγωγή της εξόδου

Η διαδικασία μάθησης διακρίνεται σε δύο κατηγορίες:

1. Μάθηση υπό επίβλεψη (Supervised Learning):

Αποτελεί την εκπαίδευση υπολογιστικών προγραμμάτων με σκοπό να μάθουν συσχετισμούς ανάμεσα στα δεδομένα εισόδου και στα δεδομένα εξόδου δια μέσου της ανάλυσης των επιθυμητών εξόδων όπως τις καθορίζει ο επιβλέπων, ο οποίος συνήθως είναι άνθρωπος.

Για κάθε παράδειγμα εκπαίδευσης θα υπάρχει ένα σετ δεδομένων εισόδου σε μορφή πολυδιάστατων διανυσμάτων που θα τροφοδοτηθεί στο επίπεδο εισόδου και μία ή περισσότερες αντίστοιχες καθορισμένες τιμές εξόδου, δηλαδή τα δεδομένα εισόδου είναι προεπισημασμένα. Στόχο αυτής της μορφής εκπαίδευσης είναι να μειώσει το συνολικό σφάλμα ταξινόμησης του μοντέλου προσδιορίζοντας κατάλληλα την τιμ ΄γ της εξόδου μέσω της διαδικασίας της εκπαίδευσης *([O' Sh 15])*.

Πρακτικά, το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν «δάσκαλο» (training data) με στόχο να μάθει έναν γενικό κανόνα προκειμένου να αντιστοιχίζει ορθώς τις εισόδους με τα αποτελέσματα.

2. Μάθηση χωρίς επίβλεψη (Unsupervised Learning):

Διαφέρει από τη μάθηση υπό επίβλεψη υπό το πρίσμα ότι τα δεδομένα εκπαίδευσης δεν είναι επισημασμένα. Δίνονται μόνο δεδομένα εισόδου. Οι αλγόριθμοι μάθησης χωρίς επίβλεψη δεν χρειάζονται την ανθρώπινη μεσολάβηση για τον εντοπισμό χρήσιμων δεδομένων και συσχετίσεων. Στόχος δεν είναι η εκτίμηση των αποτελεσμάτων από τα δεδομένα εισόδου αλλά η επεξεργασία τους. Η επιτυχία καθορίζεται συνήθως από την ικανότητα του δικτύου να μειώσει ή να αυξήσει μία σχετική συνάρτηση κόστους.

Αξίζει να σημειωθεί πως συνήθως, οι εργασίες με σκοπό την αναγνώριση μοτίβων σε εικόνες επιτυγχάνονται μέσω της ταξινόμησής τους, μία διαδικασία που βασίζεται στη μάθηση υπό επίβλεψη *([O' Sh 15])*.

Η εκπαίδευση των νευρωνικών δικτύων διακρίνεται επίσης σε δύο κατηγορίες:

1. Αναστροφή μετάδοσης λάθους (Back propagation):

Η πιο διαδεδομένη μέθοδος εκπαίδευσης ANN πολλών επιπέδων. Έχει ως στόχο τη μείωση της διαφοράς ανάμεσα στις προβλεπόμενες εξόδους του μοντέλου και στις πραγματικές τιμές των δεδομένων προσαρμόζοντας τα βάρη και τις προκαταλήψεις του δικτύου. Ο τρόπος λειτουργίας του είναι επαναληπτικός καθώς σε κάθε εποχή

προσαρμόζει τις παραπάνω παραμέτρους υπολογίζοντας τον τελεστή ανάδελτα (κλίση) του σφάλματος ως προς την προκατάληψη και τα βάρη του μοντέλου *([Goo 16])*.

Πρακτικά, εφαρμόζεται ο κανόνας της αλυσιδωτής παραγώγισης για τον υπολογισμό των μερικών παραγώγων της συνάρτησης σφάλματος ως προς τις παραμέτρους βάρους w_i κάθε επιπέδου του νευρωνικού δικτύου $\frac{\partial E_{cost}}{\partial w_{ij}}$.

Στην προώθηση προς τα εμπρός, τα δεδομένα εισόδου τροφοδοτούνται στο στρώμα εισόδου, τα οποία στη συνέχεια, μαζί με τα αντίστοιχα βάρη τους, προωθούνται στα κρυφά επίπεδα. Τα κρυφά επίπεδα πολλαπλασιάζουν τα δεδομένα εισόδου με τα βάρη και σε κάθε αποτέλεσμα προστίθεται η παράμετρος της προκατάληψης (bias).

Τα χαρακτηριστικά των δεδομένων σε ένα νευρωνικό δίκτυο περιγράφονται με μία μη γραμμική συνάρτηση λόγω της πολυπλοκότητάς τους. Αυτό επιτυγχάνεται μέσω ενός μαθηματικού μετασχηματισμού που ελέγχεται από παραμέτρους που έχει μάθει το μοντέλο ακολουθούμενο από μία μη γραμμική συνάρτηση που ονομάζεται συνάρτηση ενεργοποίησης.

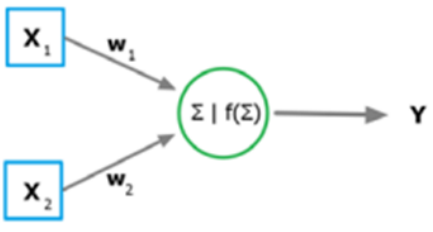
Οι πιο διαδεδομένες συναρτήσεις ενεργοποίησης είναι η Σιγμοειδής (Sigmoid) η οποία χρησιμοποιείται σε προβλήματα δυαδικής ταξινόμησης και η ReLU (Rectified Linear Unit) η οποία χρησιμοποιείται στη διαχείριση αρνητικών τιμών δεδομένων στα νευρωνικά δίκτυα βαθιάς μάθησης *([Rel 25])*.

Η συνάρτηση ενεργοποίησης αποφασίζει αν ένας νευρώνας θα μεταδώσει την έξοδό του στους επόμενους νευρώνες. Αυτή η διαδικασία προσθέτει μη γραμμικότητα, επιτρέποντας στο μοντέλο να μάθει τις περίπλοκες σχέσεις των δεδομένων. Τελικά, οι έξοδοι από το τελευταίο κρυφό επίπεδο προωθούνται στο στρώμα εξόδου, όπου σε ένα πρόβλημα ταξινόμησης η συνάρτηση ενεργοποίησης μετατρέπει τις σταθμισμένες εξόδους σε πιθανότητες προς ταξινόμηση, ενώ σε ένα πρόβλημα παλινδρόμησης μέσω της συνάρτησης ενεργοποίησης παράγονται οι συνεχείς τιμές.

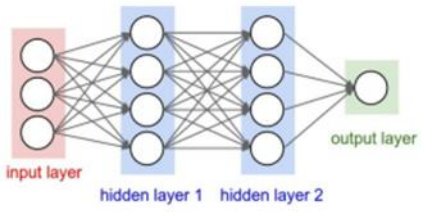
Στην οπισθοδιάδοση, το σφάλμα διαδίδεται προς τα πίσω μέσω του δικτύου με σκοπό την προσαρμογή τόσο των βαρών όσο και των προκαταλήψεων. Η οπισθοδιάδοση συνεχίζεται από επίπεδο σε επίπεδο, διασφαλίζοντας πως το δίκτυο μαθαίνει και βελτιώνει την απόδοσή του.

2. <u>Κανόνας Δέλτα (Delta Rule):</u>

Αφορά ΑΝΝ τα οποία δεν έχουν κρυφά (ενδιάμεσα) επίπεδα. Αποτελεί έναν κανόνα διαβάθμισης κλίσης για την αναπροσαρμογή των βαρών των δεδομένων εισόδου στους τεχνητούς νευρώνες σε ένα μονοεπίπεδο νευρωνικό δίκτυο. Μπορεί να προκύψει ως τον αλγόριθμο αναστροφής μετάδοσης λάθους νευρωνικού δικτύου με ένα επίπεδο, με συνάρτηση απώλειας τη συνάρτηση μέσου τετραγωνικού σφάλματος.



Εικόνα 3: Κανόνας δέλτα



Εικόνα 4: Αναστροφή μετάδοσης λάθους

Διάκριση Δεδομένων:

- Γραμμικά ανεξάρτητες συναρτήσεις:

Ένα απλό νευρωνικό δίκτυο ορίζει μία ευθεία που χωρίζει το επίπεδο σε τμήματα δημιουργώντας ένα όριο απόφασης μεταξύ των δύο εναλλακτικών αποφάσεων ανάλογα με τη θέση των δεδομένων εισόδου (x_1, x_2), δηλαδή των σημείων του επιπέδου, πάνω ή κάτω από την ευθεία που χωρίζει τις δύο κλάσεις καλύτερα.

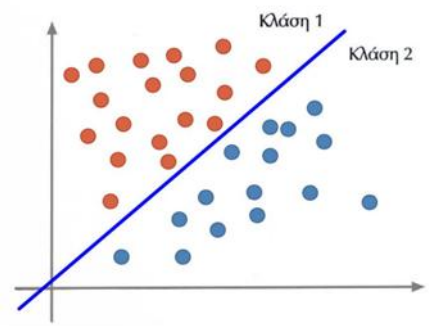
Η εξίσωση της ευθείας είναι της μορφής $\boldsymbol{w_1 x_1 + w_2 x_2 + b = 0}$, όπου

$\boldsymbol{w_1, w_2}$: Τα βάρη που καθορίζουν την κλίση της ευθείας και

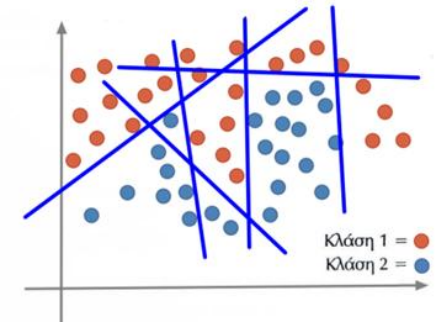
$\boldsymbol{b}$: Η σταθερά προκατάληψης που μετατοπίζει την ευθεία.

- Μη γραμμικά ανεξάρτητες συναρτήσεις:

Στην περίπτωση περίπλοκων δεδομένων, χρησιμοποιούνται πολλαπλά όρια αποφάσεων και κατ' επέκταση πολυεπίπεδα νευρωνικά δίκτυα (Perceptron).



Εικόνα 5: Διαχωρισμός δεδομένων σε δύο κλάσεις χρησιμοποιώντας γραμμικά ανεξάρτητες συναρτήσεις.



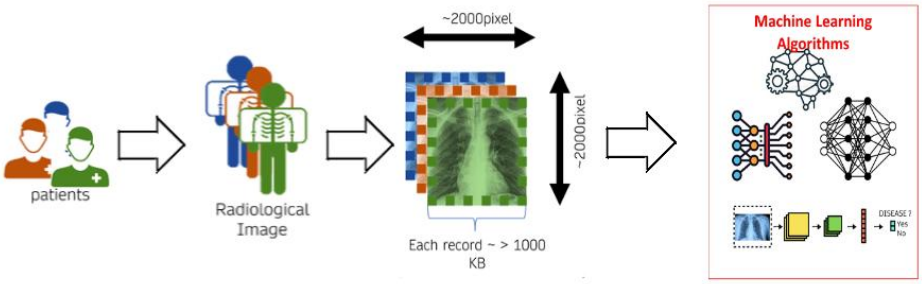
Εικόνα 6: Διαχωρισμός περίπλοκων δεδομένων σε δύο κλάσεις χρησιμοποιώντας μη γραμμικά ανεξάρτητες συναρτήσεις.

1.6 Μηχανική Μάθηση

Έναν υποτομέα της τεχνητής νοημοσύνης αποτελεί η μηχανική μάθηση (Machine Learning) στον οποίο τα υπολογιστικά συστήματα μαθαίνουν συσχετίσεις προβλεπτικής ικανότητας από παραδείγματα στα δεδομένα. Η μηχανική μάθηση αποτελεί την εφαρμογή στατιστικών μοντέλων σε δεδομένα αξιοποιώντας τους υπολογιστές και χρησιμοποιεί ένα ευρύτερο σύνολο στατιστικών τεχνικών σε σχέση με τις συμβατικές μεθόδους στατιστικής ανάλυσης.

Η βαθιά μάθηση αποτελεί μία από τις πιο ευρέως χρησιμοποιούμενες μεθόδους μηχανικής μάθησης. Τα παραδοσιακά νευρωνικά δίκτυα διαθέτουν 2-3 κρυφά επίπεδα μεταξύ του επιπέδου της εισόδου και της εξόδου, ενώ τα δίκτυα βαθιάς μάθησης (Deep Learning) μπορεί να έχουν δεκάδες ή εκατοντάδες κρυφά επίπεδα καθιστώντας τα ικανά να χειρίζονται πιο σύνθετα δεδομένα . Τα συγκεκριμένα δίκτυα χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών, την ταξινόμηση και τη μείωση της διαστατικότητας. Τα δίκτυα βαθιάς μάθησης επιτρέπουν σε μία μηχανή να τροφοδοτείται με μεγάλες ποσότητες ακατέργαστων δεδομένων και να εξάγουν χαρακτηριστικά που είναι απαραίτητα για τον εντοπισμό χαρακτηριστικών και την ταξινόμηση δεδομένων *([Abu 23])*. Μέσω των πολλαπλών κρυφών επιπέδων είναι δυνατή η εμβάθυνση στα χαρακτηριστικά των

δεδομένων, οδηγώντας στη δημιουργία μοντέλων βαθιάς μάθησης με εξαιρετικά υψηλή ακρίβεια η οποία ενδεχομένως να ξεπερνάει και την ακρίβεια αναγνώρισης χαρακτηριστικών του ανθρώπινου εγκέφαλου, καθιστώντας τα ένα πολύτιμο εργαλείο και στην απεικονιστική ιατρική.



Εικόνα 7: Από τους ασθενείς στις ιατρικές εικόνες μεγάλων διαστάσεων και στην ανάλυσή τους από μοντέλα μηχανικής μάθησης

2. Συνελικτικά Νευρωνικά Δίκτυα (CNN)

Τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks – CNN) είναι ανάλογα των Τεχνητών Νευρωνικών Δικτύων καθώς αποτελούνται από νευρώνες που αυτοβελτιστοποιούνται κατά τη διαδικασία μάθησης. Κάθε νευρώνας λαμβάνει μία είσοδο και εκτελεί μία διαδικασία. Ο όρος συνέλιξη χρησιμοποιείται συχνά στην επεξεργασία εικόνων τόσο για την εξαγωγή χαρακτηριστικών, όσο και για τον εντοπισμό αντικειμένων σε εικόνες.

Η κύρια διαφορά μεταξύ των ANN και των CNN έγκειται στο ότι τα CNN χρησιμοποιούνται κυρίως στην αναγνώριση μοτίβων σε εικόνες επιτρέποντας στον δίκτυο να εκτελέσει εργασίες οι οποίες είναι περισσότερο εστιασμένες στην εικόνα μειώνοντας παράλληλα τις παραμέτρους που απαιτούνται για τη δημιουργία του μοντέλου.

2.1 Αρχιτεκτονική των CNN

Τα CNN αποτελούνται από πολλά επίπεδα, όπου το κάθε επίπεδο αποτελείται από έναν αριθμό νευρώνων οι οποίοι έχουν σαν παραμέτρους εκμάθησης τα βάρη τους w_i και την τιμή προκατάληψης b. Κάθε νευρώνας δέχεται ένα σήμα εισόδου, εφαρμόζει μία πράξη εσωτερικού γινομένου σε αυτό και στη συνέχεια εφαρμόζει στο αποτέλεσμα μία μη γραμμική συνάρτηση. Το τελευταίο επίπεδο των δικτύων συνέλιξης είναι πλήρως συνδεδεμένο και διαθέτει μία συνάρτηση κόστους – σφάλματος *([O' Sh 15])*. Η μαθηματική έκφραση για την έξοδο ενός νευρώνα σε ένα συνελικτικό νευρωνικό δίκτυο είναι:

$$y = f \cdot \left(\sum_i x_i w_i + b \right),$$

Όπου y: Η έξοδος του νευρώνα

f: Η μη γραμμική συνάρτηση ενεργοποίησης

x_i: Οι τιμές εισόδου του προηγούμενου επιπέδου

w_i: Τα αντίστοιχα βάρη

b: Η τιμή προκατάληψης

Τα CNN αποτελούνται από τρεις τύπους επιπέδων. Αυτά είναι τα επίπεδα συνελικτικής ανάλυσης, τα επίπεδα υποδειγματοληψίας και τα πλήρως συνδεδεμένα επίπεδα. Όταν αυτά τα επίπεδα συνδυάζονται, δημιουργείται η αρχιτεκτονική του συνελικτικού νευρωνικού δικτύου. Τα επίπεδα εκτελούν διεργασίες οι οποίες ενημερώνουν με προσαρμοστικό τρόπο τα δεδομένα με σκοπό να μάθουν συγκεκριμένα χαρακτηριστικά των δεδομένων, επαναλαμβάνοντας αυτές τις διαδικασίες σε όλα τα κρυφά επίπεδα, με κάθε επίπεδο να μαθαίνει να ανιχνεύει διαφορετικά χαρακτηριστικά. Αναλυτικότερα:

Το επίπεδο εισόδου αποθηκεύει τις τιμές των pixel της εικόνας.

Το επίπεδο συνέλιξης λαμβάνει την είσοδο που δίδεται, εφαρμόζει φίλτρα για την εξαγωγή χαρακτηριστικών και παράγει χάρτες χαρακτηριστικών που περιγράφουν τις πληροφορίες της εισόδου με πιο συμπυκνωμένο τρόπο.

Τα φίλτρα της αρχιτεκτονικής του μοντέλου εφαρμόζονται σε κάθε εικόνα με την οποία εκπαιδεύεται το δίκτυο σε διαφορετικές αναλύσεις προσδιορίζοντας με ξεχωριστό τρόπο την εικόνα *([Abu 23])*. Αναλυτικότερα, ως φίλτρο ορίζεται ένας πίνακας μικρών διαστάσεων (συνήθως 3x3 ή 5x5) με αριθμητικές τιμές που αποτελούν τα βάρη. Κάθε φίλτρο ελέγχει την είσοδο και υπολογίζει τη συσχέτιση μεταξύ του φίλτρου και των τοπικών περιοχών της εισόδου, δηλαδή μικρών διαδοχικών κομματιών της εικόνας που αναλύεται κάθε στιγμή από το φίλτρο. Στη συνέχεια, για κάθε θέση του φίλτρου, υπολογίζεται το εσωτερικό γινόμενο μεταξύ των τιμών της περιοχής της εισόδου και του φίλτρου. Το αποτέλεσμα αποθηκεύεται σε έναν πίνακα που ονομάζεται χάρτης χαρακτηριστικών (feature map).

Μετά το επίπεδο συνέλιξης, εφαρμόζεται μία μη γραμμική συνάρτηση, η ReLU (Rectified Linear Unit) που μηδενίζει τις αρνητικές τιμές για να εισάγει μη γραμμικότητα στο δίκτυο. Αποτελεί στοιχειομετρική συνάρτηση ενεργοποίησης στην έξοδο που παράγεται από το προηγούμενο επίπεδο. Με τη χρήση της αλλάζει δραστικά η συμπεριφορά της εξόδου ανάλογα με το αν η είσοδος είναι θετική ή μη.

Η ReLU χρησιμοποιείται στα ενδιάμεσα επίπεδα των νευρωνικών δικτύων και ορίζεται ως:

$$\boldsymbol{f(x) = max(0, x) \Rightarrow f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}}$$

Αντίστοιχα, η συνάρτηση ενεργοποίησης Softmax χρησιμοποιείται στο επίπεδο εξόδου για προβλήματα ταξινόμησης. Ορίζεται ως:

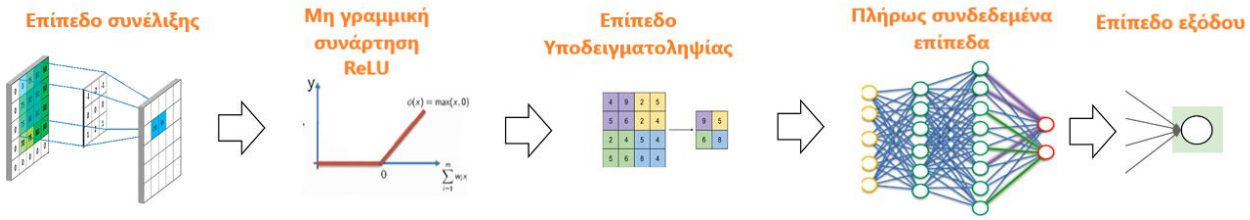
$$\boldsymbol{f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}}$$

Όπου έχει ως διάνυσμα εισόδου: $\boldsymbol{x = [x_1, x_2, \dots, x_N]}$ και ως $\boldsymbol{x_i}$ την τιμή της i εισόδου *([ReL 25])*.

Στη συνέχεια, το επίπεδο υποδειγματοληψίας (Pooling layer) μειώνει το δείγμα κατά μήκος της χωρικής διάστασης της δεδομένης εισόδου μειώνοντας περαιτέρω τον αριθμό των παραμέτρων εντός της εν λόγω συνάρτησης ενεργοποίησης. Ουσιαστικά, το επίπεδο υποδειγματοληψίας είναι μικρότερη αναπαράσταση της εισαγόμενης εικόνας. Με αυτόν τον τρόπο, όταν μία περιοχή στο επίπεδο εισόδου αφαιρείται, οι περισσότερες αναπαραστάσεις στο επίπεδο συγκέντρωσης διατηρούνται. Αυτό το επίπεδο μειώνει την εξάρτηση των αποτελεσμάτων που παράγει το μοντέλο σε ολόκληρη την είσοδο *([O' Sh 15])*.

Τέλος, τα πλήρως συνδεδεμένα επίπεδα συνδέονται με όλους τους νευρώνες στα προηγούμενα επίπεδα και παράγουν τα αποτελέσματα των τάξεων από τις ενεργοποιήσεις. Ειδικότερα, σε ένα πρόβλημα ταξινόμησης υπολογίζουν την πιθανότητα εμφάνισης κάθε ετικέτας, ενώ σε ένα πρόβλημα παλινδρόμησης υπολογίζει τις εκτιμώμενες τιμές εξόδου στα τελευταία δεδομένα που μειώθηκαν στη μία διάσταση.

Συμπερασματικά, η αρχιτεκτονική των CNN είναι ικανή να εξάγει (ιεραρχικά) χαρακτηριστικά από τις εικόνες χρησιμοποιώντας διαδικασίες συνέλιξης με τα φίλτρα. Συνεπώς, τα μοντέλα CNN στοχεύουν στην αυτόματη αναγνώριση γωνιών, σχημάτων, και την αυτόματη εξαγωγή άλλων σημαντικών χαρακτηριστικών μέσα σε μία εικόνα και για το λόγο αυτό απαιτούν λιγότερες παραμέτρους από το τυπικό ΤΝΔ για την επίλυση του ίδιου προβλήματος.



Εικόνα 8: Σχηματική αναπαράσταση αρχιτεκτονικής νευρωνικού δικτύου

2.2 Εκπαίδευση CNN

Προκειμένου να ληφθούν οι παραπάνω προβλέψεις, το CNN εκπαιδεύεται μέσω μίας διαδικασίας που περιλαμβάνει τη διαδοχική προώθηση πληροφορίας (forward pass) και την προσαρμογή των βαρών μέσω οπισθοπροβολής (back propagation). Στην προώθηση, τα δεδομένα εισόδου περνούν μέσα από τα επίπεδα του δικτύου και παράγεται η πρόβλεψη ως έξοδος. Το σφάλμα της πρόβλεψης υπολογίζεται μέσω της συνάρτησης απώλειας η οποία ποσοτικοποιεί τη διαφορά μεταξύ της προβλεπόμενης και της πραγματικής ετικέτας.

Η συνάρτηση της απώλειας εκφράζει τη διαφορά μεταξύ των προβλέψεων του εκπαιδευόμενου μοντέλου και των πραγματικών καταστάσεων – ετικετών του προβλήματος. Η διαφορά των δύο τομέων απορρέει από τον στόχο της γενίκευσης: Ενώ οι αλγόριθμοι βελτιστοποίησης μπορούν να ελαχιστοποιήσουν την απώλεια ενός συνόλου εκπαίδευσης, η μηχανική μάθηση εστιάζει στην ελαχιστοποίηση της απώλειας σε άγνωστες καταστάσεις που στα παρακάτω προβλήματα αποτελούν τα σετ δοκιμής

(test set). Στη συνέχεια, το σφάλμα επιστρέφεται προς τα πίσω με τη διαδικασία της οπισθοδιάδοσης.

Συνάρτηση βελτιστοποίησης (optimizer) αποτελεί ένας αλγόριθμος ή μέθοδος που χρησιμοποιείται για την ελαχιστοποίηση της συνάρτησης σφάλματος (συνάρτηση απώλειας) ή για τη μεγιστοποίηση της απόδοσης του αλγορίθμου. Είναι μαθηματική συνάρτηση που εξαρτάται από τις παραμέτρους του μοντέλου. Η συνάρτηση βελτιστοποίησης συνεισφέρει στην εύρεση του ρυθμού μάθησης του νευρωνικού δικτύου με σκοπό την μείωση των απωλειών *([Goo 16]).* Ο αλγόριθμος βελτιστοποίησης που χρησιμοποιείται στους παρακάτω αλγόριθμους ονομάζεται Adam (Adaptive Model Estimation – Προσαρμοζόμενη Εκτίμηση Μοντέλου). Η αρχή λειτουργίας της συνάρτησης βελτιστοποίησης περιγράφεται στα παρακάτω βήματα:

Αρχικά, ορίζει την πρώτη και τη δεύτερη ροπή. Η πρώτη ροπή είναι ο μέσος όρος του πόσο απότομη είναι η κλίση και αποτυπώνει τη γενική κατεύθυνση της βελτιστοποίησης, ενώ η δεύτερη ροπή είναι η διακύμανση της κλίσης της συνάρτησης απώλειας με στόχο την προσαρμογή του ρυθμού εκμάθησης για κάθε παράμετρο.

Και οι δύο ροπές έχουν ως αρχική τιμή το 0. Έχουν ως αποκλειστικό στόχο τη διαχείριση της κλίσης, δηλαδή της παραγώγου της συνάρτησης απώλειας. Η διαχείριση της κλίσης γίνεται μέσω της επίτευξης της σταθερότητας και την αύξηση της ταχύτητας βελτιστοποίησης, προσαρμόζοντας τον τρόπο με τον οποίο ανανεώνονται οι παράμετροι του μοντέλου για να εξυπηρετούν τον σκοπό της εκπαίδευσης.

Κατά τη διάρκεια της εκπαίδευσης, ο Adam ελέγχει την εξέλιξη της κλίσης και την ταχύτητα με την οποία αλλάζει. Καθώς οι δύο ροπές έχουν αρχική τιμή το 0, ο αλγόριθμος βελτιστοποίησης κάνει κάποιες προσαρμογές για να τις κάνει πιο ακριβείς.

Τέλος, χρησιμοποιεί αυτές τις μέσες τιμές για να βελτιώσει τις ρυθμίσεις του δικτύου με απώτερο σκοπό την βελτίωση της απόδοσής του. Αναλυτικότερα:

Υπολογίζεται η κλίση της συνάρτησης απώλειας L ως: $\boldsymbol{g_t = \nabla L(\theta_t)}$, όπου θ_i η κάθε παράμετρος.

Υπολογίζεται η πρώτη ροπή: $\boldsymbol{m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t}$, όπου $\boldsymbol{\beta_1 \approx 0,9}$ ο ρυθμός εκθετικής εξομάλυνσης για τη μέση τιμή.

Υπολογίζεται η δεύτερη ροπή: $\boldsymbol{v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2}$, όπου

$\boldsymbol{\beta_2 \approx 0,999}$ ο ρυθμός εκθετικής εξομάλυνσης για τη διακύμανση.

Αν η κλίση έχει μεγάλη διακύμανση, η δεύτερη ροπή θα είναι υψηλή και ο Adam θα μειώσει το μέγεθος του βήματος για μεγαλύτερη σταθερότητα.

Στη συνέχεια διορθώνεται η προκατάληψη bias που εισάγεται στις δύο πρώτες ροπές ως:

$$\begin{cases} \widehat{m_t} = \dfrac{m_t}{\beta_{1t}} \\ \widehat{v_t} = \dfrac{v_t}{\beta_{2t}} \end{cases}$$

Τέλος οι παράμετροι ανανεώνονται ως:

$$\boldsymbol{\theta_{t+1} = \theta_t - \eta \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon}}$$

Όπου $\boldsymbol{\eta}$: ο ρυθμός εκμάθησης (learning rate)

Και $\boldsymbol{\epsilon = const.} \approx \boldsymbol{10^{-8}}$: για την αποφυγή μηδενισμού του παρονομαστή *([Kin 15])*.

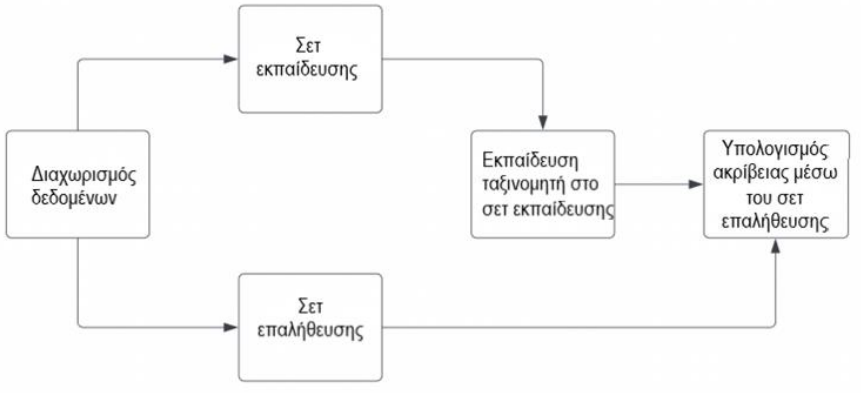
Το συνολικό πλήθος δεδομένων που τροφοδοτείται στο πρόγραμμα χωρίζεται σε δύο κατηγορίες, στο σετ εκπαίδευσης και στο σετ επαλήθευσης. Μέσω του σετ εκπαίδευσης, ο ταξινομητής του μοντέλου εκπαιδεύεται να εκτελεί την ταξινόμηση των δεδομένων σύμφωνα με το ζητούμενο αποτέλεσμα, ενώ το σετ επαλήθευσης χρησιμοποιείται για τον υπολογισμό της συνάρτησης ακρίβειας.

Η εκπαίδευση του δικτύου πραγματοποιείται σε ένα πλήθος επαναλήψεων που ονομάζονται εποχές (epochs). Κατά τη διάρκεια αυτών, το μοντέλο βλέπει ολόκληρο το σύνολο εκπαίδευσης μία φορά ανά epoch, κατά τη διάρκεια των οποίων το δίκτυο βελτιώνεται σταδιακά καθώς τα βάρη του προσαρμόζονται διαρκώς. Τόσο η συνάρτηση της απώλειας όσο και η συνάρτηση της ακρίβειας υπολογίζονται σε κάθε εποχή ξεχωριστά για το σετ εκπαίδευσης και για το σετ επαλήθευσης.

Ως epoch στη μηχανική μάθηση ορίζεται ολόκληρο το πέρασμα των δεδομένων εκπαίδευσης από τον αλγόριθμο. Πρόκειται για υπερπαράμετρο η οποία καθορίζει την εκπαίδευση του μοντέλου. Τα δεδομένα εκπαίδευσης χωρίζονται σε μικρότερες ομάδες (batches) τα οποία γίνονται ευκολότερα δεκτά από τον αλγόριθμο καθώς μειώνονται οι περιορισμοί λόγω ελλιπούς χώρου αποθήκευσης του ηλεκτρονικού υπολογιστή διασφαλίζοντας την ανεμπόδιστη εκπαίδευση του μοντέλου.

Όταν όλα τα πακέτα δεδομένων τροφοδοτηθούν στον αλγόριθμο, ολοκληρώνεται μία εποχή. Για την ολοκληρωμένη εκπαίδευση του δικτύου πρέπει να πραγματοποιηθούν αρκετές epochs. Με το βήμα αυτό, ουσιαστικά επαναλαμβάνεται ο κύκλος εκπαίδευσης τόσες φορές όσες και ο αριθμός των epochs που προσαρμόζεται για τις ανάγκες κάθε δικτύου.

Τα test data (δεδομένα δοκιμής) τροφοδοτούνται στο νευρωνικό δίκτυο με το πέρας κάθε εποχής και τα δεδομένα εξόδου, δηλαδή τα αποτελέσματα του κώδικα, συγκρίνονται με τις ζητούμενες αποκρίσεις. Γενικά, αν η απόδοση ενός υπολογιστικού προγράμματος βελτιώνεται με τον χρόνο, τότε το δίκτυο μαθαίνει αποτελεσματικά.



Εικόνα 9: Διαδικασία διαχωρισμού δεδομένων, εκπαίδευσης και εξαγωγής της ακρίβειας ενός νευρωνικού δικτύου

2.3 Υπερπροσαρμογή Δεδομένων (Overfitting)

Προκειμένου η εκμάθηση του μοντέλου να είναι αποτελεσματική, πρέπει το δίκτυο να είναι σε θέση να γενικεύει σε νέα δεδομένα στα οποία δεν έχει εκπαιδευτεί. Για το λόγο αυτό, πρέπει να γίνει αποφυγή υπερπροσαρμογής του μοντέλου στα δεδομένα εκπαίδευσης (Overfitting). Η υπερπροσαρμογή είναι ένα φαινόμενο στο οποίο το μοντέλο προσαρμόζεται πολύ στενά στα δεδομένα εκπαίδευσης και καταγράφει ακόμα και το θόρυβο και τις τυχαίες διακυμάνσεις στα δεδομένα αυτά πέρα από τα κύρια χαρακτηριστικά, με αποτέλεσμα να μην μπορεί να μάθει αποτελεσματικά, δηλαδή να αδυνατεί να παρέχει ορθή πρόβλεψη σε δεδομένα εισόδου που διαφέρουν ακόμα και ελάχιστα στο σετ εκπαίδευσης.

Η υπερπροσαρμογή μπορεί να αποφευχθεί αυξάνοντας το πλήθος των δεδομένων εκπαίδευσης και κανονικοποιώντας τις παραμέτρους του μοντέλου καθώς όσες λιγότερες παράμετροι απαιτούνται για την εκπαίδευση του νευρωνικού δικτύου, τόσο λιγότερο πιθανό είναι το νευρωνικό δίκτυο να υπερπροσαρμοστεί *([O'Sh 15])*. Με αυτόν τον τρόπο, το μοντέλο θα μπορεί να γενικεύει ικανοποιητικά σε νέα δεδομένα που δεν έχουν τροφοδοτηθεί σε εκείνο κατά την εκπαίδευση, αυξάνοντας την απόδοσή του και μειώνοντας την τιμή της συνάρτησης απώλειας.

3. Εισαγωγικά Προβλήματα Ταξινόμησης Εικόνων

Τα παρακάτω εισαγωγικά προγράμματα έχουν ως στόχο την ανάλυση της αρχιτεκτονικής απλών νευρωνικών δικτύων, εστιάζοντας στην διαδικασία εκπαίδευσης του μοντέλου και ιδιαίτερα στη διαδικασία εκμάθησης χαρακτηριστικών των εικόνων, αξιολογώντας τη βελτίωση της απόδοσής τους μέσω της προσαρμογής των βαρών και της χρήσης τεχνικών βελτιστοποίησης. Επίσης, θα αναλυθεί η αρχιτεκτονική κάθε νευρωνικού δικτύου, όπως και η δομή και η λειτουργία κάθε επιπέδου. Εν γένει, θα εξεταστεί η ικανότητα των νευρωνικών δικτύων να ταξινομούν ορθά τις εικόνες με βάση συγκεκριμένα χαρακτηριστικά που διαθέτουν.

3.1 Κατασκευή Απαιτούμενου Δείγματος Εικόνων

Το αρχικό πλαίσιο στο οποίο τοποθετούνται τα γεωμετρικά σχήματα είναι ένα πλαίσιο διαστάσεων 128x128, δηλαδή ένας πίνακας με όλα του τα στοιχεία ίσα με το 0. Η επιλογή αυτή ισοδυναμεί στην παρουσίαση του πλαισίου ως ένα τετράγωνο σε μαύρο χρώμα.

Εφαρμόζοντας τις κατάλληλες επιλογές για τον σχηματισμό των σχημάτων, η περιοχή σχηματισμού των τετραγώνων και των κύκλων θα θέσει τα αντίστοιχα στοιχεία του πίνακα ίσα με 1, δηλαδή θα παρουσιάζονται με άσπρο χρώμα.

Περιορισμοί:

Το μέγεθος του τετραγώνου τίθεται ως:

$$\boldsymbol{square\ size} = (\mathbf{14} + \boldsymbol{int}(\boldsymbol{np.random.uniform}(\mathbf{0}, \mathbf{1}) \cdot \mathbf{14})$$

Προκειμένου να αποφευχθεί ο σχηματισμός του τετραγώνου εκτός του αρχικού πλαισίου 128x128 αλλά και να δημιουργηθούν τετράγωνα με διαφορετικά μεγέθη,

προστίθεται ένας τυχαίος ακέραιος αριθμός στο διάστημα $(0, 1)$ πολλαπλασιασμένος με το 14.

Στη συνέχεια, ορίζονται ακέραιες τιμές για το κάτω και το αριστερό άκρο του τετραγώνου χρησιμοποιώντας γεννήτορα τυχαίων αριθμών στο διάστημα $(0, 1)$ πολλαπλασιασμένο με το μέγεθος του τετραγώνου μειούμενου κατά 28 προκειμένου να μην είναι τοποθετημένο στα άκρα της εικόνας.

$$\begin{cases} \boldsymbol{bottom\ edge = int\big(np.random.uniform(0, 1) \cdot (size - 28)\big)} \\ \boldsymbol{left\ edge = int\big(np.random.uniform(0, 1) \cdot (size - 28)\big)} \end{cases}$$

Ορίζεται επίσης το άνω άκρο του τετραγώνου και η δεξιά πλευρά του ως:

$$\boldsymbol{top\_edge = bottom\_edge + square\_size}$$

Για $\boldsymbol{x} \in [\boldsymbol{left\ edge, right\ edge}]$ και $\boldsymbol{y} \in [\boldsymbol{bottom\ edge, top\ edge}]$: $\boldsymbol{x = y = 1}$,

δηλαδή τα στοιχεία του πίνακα που οριοθετούνται σε αυτές τις περιοχές τίθενται ίσα με 1, έχοντας ως οπτικό αποτέλεσμα την εμφάνιση των τετραγώνων με λευκό χρώμα.

Για την κατασκευή ενός κύκλου ακολουθούνται τα παρακάτω βήματα:

Το μέγεθος της ακτίνας του κύκλου ορίζεται ως: $\begin{cases} \boldsymbol{\min\_radius = 5} \\ \boldsymbol{\max\_radius = 25} \end{cases} \Rightarrow$

$\boldsymbol{radius = random} \in [\boldsymbol{\min\_radius}, \boldsymbol{\max\_radius}]$ με $\boldsymbol{radius} \in \mathbb{Z}$

Το κέντρο του κύκλου: $\begin{cases} \boldsymbol{center\_x = random} \in (\boldsymbol{radius, size - radius}) \\ \boldsymbol{center\_y = random} \in (\boldsymbol{radius, size - radius}) \end{cases}$

Άρα για τον κύκλο: $\boldsymbol{circle} = (\boldsymbol{x - center\_x})^2 + (\boldsymbol{y - center\_y})^2 \leq \boldsymbol{radius^2}$

Για $\boldsymbol{x, y\ \epsilon\ circle}$: $\boldsymbol{x = y = 1}$, δηλαδή τα στοιχεία του πίνακα που οριοθετούνται σε αυτές τις περιοχές τίθενται ίσα με 1, έχοντας ως οπτικό αποτέλεσμα την εμφάνιση των κύκλων με λευκό χρώμα.

3.2 **1ο Πρόβλημα Ταξινόμησης**

Ταξινόμηση εικόνων διαφορετικών απλών γεωμετρικών σχημάτων μεταβλητών διαστάσεων ως προς το είδος αυτών

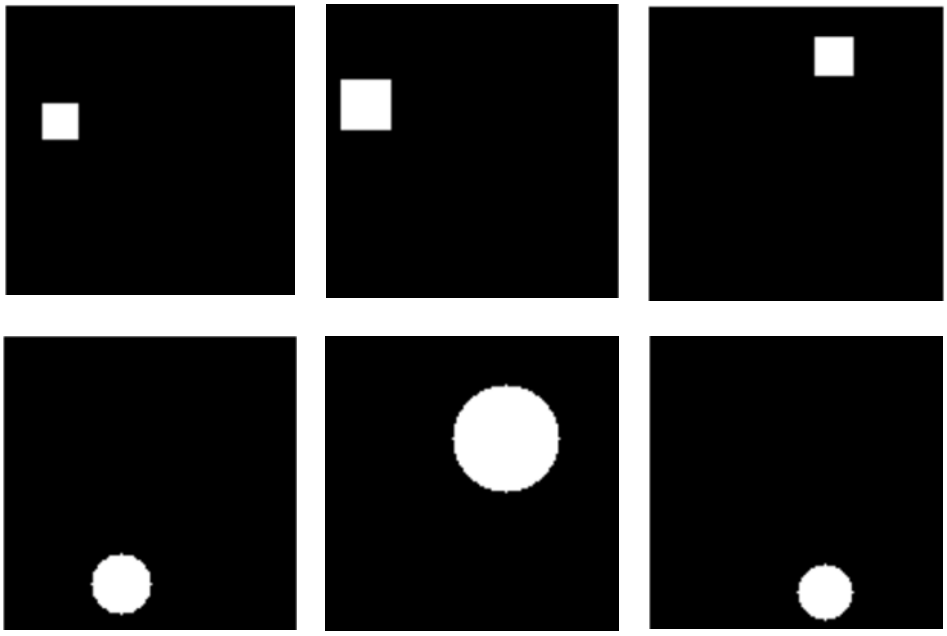
3.2.1 Σκοπός του προβλήματος ταξινόμησης

Το παρόν πρόβλημα αποτελείται από ένα δείγμα 3000 εικόνων απλών γεωμετρικών σχημάτων. Οι 1500 εικόνες του δείγματος περιλαμβάνουν ένα τετράγωνο μεταβλητών διαστάσεων και οι άλλες 1500 έναν κύκλο επίσης μεταβλητών διαστάσεων. Σκοπός του προγράμματος είναι η δυαδική ταξινόμηση των εικόνων σε δύο κατηγορίες: Στις εικόνες που περιλαμβάνουν το τετραγωνικό σχήμα και στις εικόνες που περιλαμβάνουν το κυκλικό σχήμα.

3.2.2 Μεθοδολογία κατασκευής νευρωνικού δικτύου ταξινόμησης

Αρχικά κατασκευάζεται το απαιτούμενο δείγμα: 1500 εικόνες που η καθεμία θα περιλαμβάνει ένα τετράγωνο μεταβλητών διαστάσεων και 1500 εικόνες που η καθεμία θα περιλαμβάνει έναν κύκλο μεταβλητών διαστάσεων.

Ενδεικτικές εικόνες που παράγονται:



Οι εικόνες αποθηκεύονται σε αντίστοιχα ευρετήρια – directories μαζί με τις αντίστοιχες ετικέτες, δηλαδή το είδος του σχήματος που περιλαμβάνει η κάθε εικόνα.

Το σετ δεδομένων, συνολικού όγκου 3000 εικόνων, χωρίζεται σε training data (70% του αρχικού δείγματος) και σε test data (30% του αρχικού δείγματος).

Η εκπαίδευση του μοντέλου ολοκληρώνεται σε 20 κύκλους εκπαίδευσης (epochs) κατά την ολοκλήρωση των οποίων παρακολουθείται η εξέλιξη των μετρικών.

Οι μετρικές του συγκεκριμένου προγράμματος ορίστηκαν να είναι η ακρίβεια (accuracy) και η απώλεια (loss).

Ως συνάρτηση της απώλειας ορίζεται η Binary Cross – Entropy, ή αλλιώς η λογαριθμική απώλεια:

$\boldsymbol{Loss} = -\frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} [\boldsymbol{y_i} \cdot \mathbf{log}(\boldsymbol{p}(\boldsymbol{y_i})) + (\mathbf{1} - \boldsymbol{y_i}) \cdot \mathbf{log}(\mathbf{1} - \boldsymbol{p}(\boldsymbol{y_i}))]$, όπου:

$\boldsymbol{y}$: είναι η ετικέτα, 0 για τις εικόνες που περιέχουν το κυκλικό σχήμα και 1 για εκείνες που περιέχουν το τετραγωνικό σχήμα.

$\boldsymbol{p}(\boldsymbol{y})$**:** η προβλεπόμενη πιθανότητα για κάθε εικόνα να περιλαμβάνει το τετραγωνικό σχήμα

$\boldsymbol{N}$: το πλήθος των εικόνων

Καθώς η πιθανότητα πρόβλεψης της σωστής τάξης είναι 1.0, ζητούμενο είναι η απώλεια σε αυτή την περίπτωση να είναι 0. Αντίστοιχα, αν η πιθανότητα πρόβλεψης σωστής τάξης είναι χαμηλή, για παράδειγμα 0.01, η απώλεια σε αυτή την περίπτωση θα είναι αρκετά μεγάλη. Για τον λόγο, ο υπολογισμός της απώλειας εμπεριέχει τη λογαρίθμηση των πιθανοτήτων, ενώ το αρνητικό πρόσημο διατηρείται για να εξαχθεί θετική τιμή για την απώλεια.

Συμπερασματικά, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής, πρέπει η μετρική της απώλειας να ακολουθεί αρνητική λογαριθμική πορεία και τείνει προς το 0.

Για τον υπολογισμό της απόδοσης ακολουθείται το εξής σενάριο:

- Κατηγορία 1 (Positive, label =1): Η εικόνα περιέχει το τετραγωνικό σχήμα
- Κατηγορία 2 (Negative, label =0): Η εικόνα περιέχει το κυκλικό σχήμα

Κατά την εκπαίδευση του μοντέλου δημιουργούνται 4 νέες καταστάσεις:

- True Positive (TP): Η εικόνα περιλαμβάνει τετράγωνο και το μοντέλο την ταξινόμησε σωστά ως τετράγωνο.
- True Negative (TN): Η εικόνα περιλαμβάνει κύκλο και το μοντέλο την ταξινόμησε σωστά ως κύκλο.
- False Positive (FP): Η εικόνα περιλαμβάνει κύκλο και το μοντέλο την ταξινόμησε λανθασμένα ως τετράγωνο.
- False Negative (FN): Η εικόνα περιλαμβάνει τετράγωνο και το μοντέλο την ταξινόμησε λανθασμένα ως κύκλο.

Πραγματική κατάσταση	Πρόβλεψη μοντέλου	Κατηγορία
Τετράγωνο (label =1)	Τετράγωνο	True Positive (TP)
Τετράγωνο (label =1)	Κύκλος	False Negative (FN)
Κύκλος (label =0)	Κύκλος	True Negative (TN)
Κύκλος (label =0)	Τετράγωνο	False Positive (FP)

Table 2: Συγκεντρωτικός πίνακας με τις πραγματικές καταστάσεις του 1ου προβλήματος ταξινόμησης σε αντιδιαστολή με τις προβλεπόμενες και οι κατηγορίες που δημιουργούνται.

Έτσι, ορίζεται η ακρίβεια (accuracy) ως: $\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}$

Κατά τον υπολογισμό της από τον αλγόριθμο, προκύπτει ένας δεκαδικός αριθμός ο οποίος πολλαπλασιασμένος με το 100 δίνει την τιμή της ακρίβειας επί τοις εκατό.

Συμπερασματικά, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής η ακρίβεια πρέπει να αυξάνει προς την τιμή 1.0 με το πέρας των εποχών.

3.2.3 Ανάλυση αρχιτεκτονικής νευρωνικού δικτύου

Το dataset χωρίστηκε με τον εξής τρόπο:

Σετ εκπαίδευσης (Training set): 70% των παραγόμενων εικόνων, δηλαδή 2100 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου.

Σετ επαλήθευσης (Validation set): 30% των παραγόμενων εικόνων, δηλαδή 900 εικόνες χρησιμοποιήθηκαν για την επικύρωση του μοντέλου.

Τα δύο σετ δεν αναμειγνύονται προκειμένου η αξιολόγηση να γίνει πάνω σε άγνωστες για το μοντέλο εικόνες.

Ως προς την αρχιτεκτονική του μοντέλου, χρησιμοποιήθηκαν 2 επίπεδα συνέλιξης, το πρώτο με 32 φίλτρα (3x3) και το δεύτερο με 64 φίλτρα (3x3) για την εξαγωγή χαρακτηριστικών από τις εικόνες.

Στο επίπεδο της υποδειγματοληψίας χρησιμοποιήθηκε ο αλγόριθμος MaxPooling2D για τη μείωση των διαστάσεων κάθε εικόνας με σκοπό την αποδοτικότερη εξαγωγή χαρακτηριστικών. Επίσης, στο πλήρως συνδεδεμένο επίπεδο , η συνάρτηση Flatten μετατρέπει τους δισδιάστατους πίνακες με τις εξαγωγές των χαρακτηριστικών σε μονοδιάστατο διάνυσμα, ενώ το επίπεδο Dense διαθέτει 128 νευρώνες οι οποίοι είναι πλήρως συνδεδεμένοι με όλους τους νευρώνες των προηγούμενων επιπέδων και ενισχύει την εκμάθηση μη γραμμικών σχέσεων, συνθέτοντας τα χαρακτηριστικά των προηγούμενων επιπέδων συνέλιξης.

Χρησιμοποιήθηκε η τεχνική Dropout για να αποφευχθεί η υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης, όπου το 25% των νευρώνων απενεργοποιούνται σε κάθε στάδιο της εκπαίδευσης.

Ως συνάρτηση ενεργοποίησης ορίστηκε η ReLU για την εξαγωγή αφαιρετικών χαρακτηριστικών. Τέλος, ως έξοδος ορίστηκε ένας μόνο νευρώνας με ενεργοποίηση sigmoid η οποία επιτρέπει τη δυαδική ταξινόμηση.

3.2.4 Αξιολόγηση μετρικών του νευρωνικού δικτύου

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε σε 20 εποχές στις οποίες αξιολογούνταν τόσο η ακρίβεια, όσο και η συνάρτηση απώλειας του δικτύου κάνοντας χρήση της συνάρτησης βελτιστοποίησης Adam.

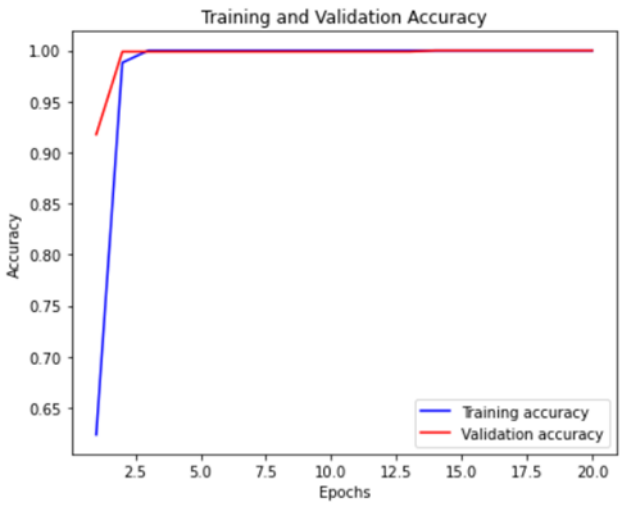


Figure 10: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται σταθεροποίηση σε μέγιστη τιμή

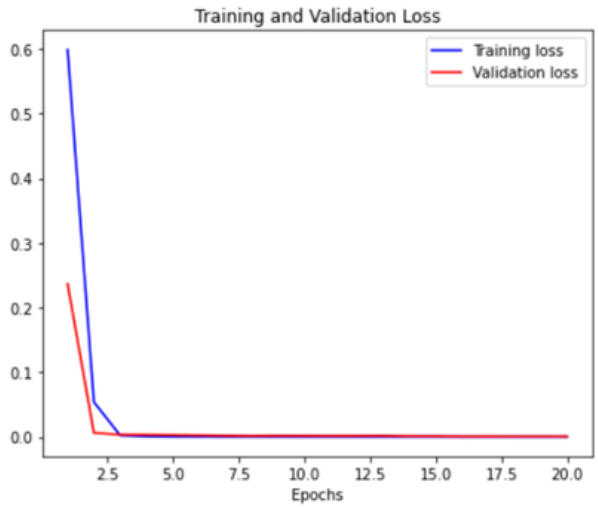
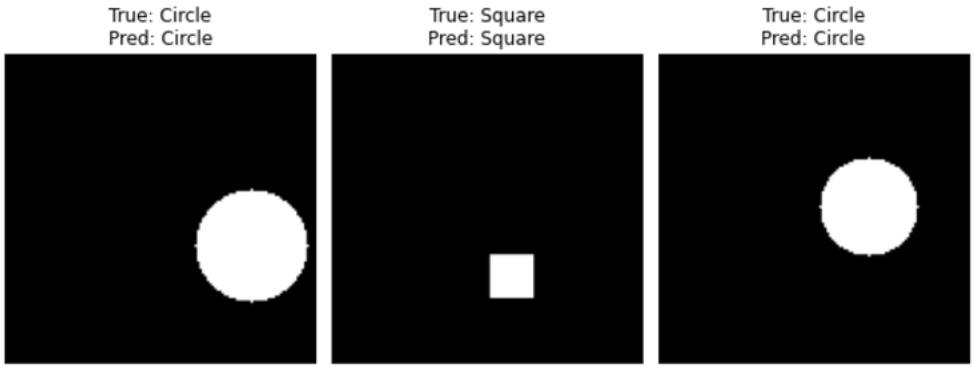


Figure 11: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή

Το δίκτυο κατάφερε να επιτύχει ακρίβεια σχεδόν της τάξης του 100% τόσο στο σετ εκπαίδευσης (Epoch 3/20) , όσο και στο σετ επαλήθευσης (Epoch 14/20). Όμοια, η συνάρτηση της απώλειας ελαχιστοποιήθηκε, παρουσιάζοντας πλατό, γεγονός που δεν υποδεικνύει υπερπροσαρμογή στα δεδομένα εκπαίδευσης. Ειδικότερα, στην τελευταία εποχή η συνάρτηση απώλειας που αφορά τα δεδομένα εκπαίδευσης είναι $2{,}32 \cdot 10^{-5}$, ενώ στην αξιολόγηση του μοντέλου μέσω του τεστ σετ, η συνάρτηση της απώλειας φτάνει την τιμή $4{,}66 \cdot 10^{-4}$. Η διαφορά στην τάξη μεγέθους οφείλεται στη μεγαλύτερη έκταση του σετ εκπαίδευσης, γεγονός που επιτρέπει στο μοντέλο να επεξεργάζεται τα δεδομένα περισσότερες φορές και ως εκ τούτου να οδηγείται σε μεγαλύτερη προσαρμογή και σε ελαχιστοποίηση της απώλειας, ενώ τα δεδομένα εκπαίδευσης χρησιμοποιούνται απλώς για την αξιολόγηση του μοντέλου στο τέλος κάθε εποχής. Γενικά, η απόδοση του μοντέλου στην δυαδική ταξινόμηση των εικόνων ως προς το είδος του απλού γεωμετρικού σχήματος που περιλαμβάνουν μπορεί να θεωρηθεί επιτυχής καθώς η συνολική ακρίβεια μετά το πέρας των εποχών εκπαίδευσης είναι 100% ενώ η συνάρτηση της απώλειας είναι της τάξης του $\sim 4{,}7 \cdot 10^{-4}$.



Εικόνα 12: Τυχαίες εικόνες από το σετ επαλήθευσης με τις προβλεπόμενες ετικέτες τους

3.3 **2ο Πρόβλημα Ταξινόμησης**

Ταξινόμηση εικόνων σε 4 κατηγορίες ως προς την πολλαπλότητα του ιδίου γεωμετρικού σχήματος σταθερών διαστάσεων

3.3.1 Σκοπός του προβλήματος ταξινόμησης

Στο παρόν πρόβλημα κατασκευάζεται ένα δείγμα 3000 εικόνων, η καθεμία διαστάσεων 128x128 που περιλαμβάνουν πολλαπλότητα 1 έως 4 του ιδίου γεωμετρικού σχήματος (τετραγώνου) με μεταβλητές διαστάσεις. Ζητούμενο είναι η ταξινόμηση των εικόνων σε τέσσερις κατηγορίες, ανάλογα με το πλήθος των σχημάτων που περιέχουν. Για τον λόγο αυτό, θα παρουσιαστούν δύο αλγόριθμοι, ένας που θα επιτρέπει την επικάλυψη μεταξύ των σχημάτων και ένας που θα την απαγορεύει.

Στη συνέχεια, μετά τον υπολογισμό των αντίστοιχων ακριβειών, θα υπολογιστούν οι τυπικές αποκλίσεις και θα κατασκευαστεί ένα κοινό διάγραμμα που θα απεικονίζει τα εύρη των δύο τυπικών αποκλίσεων.

Ζητούμενο αποτέλεσμα είναι αφενός η επίτευξη υψηλότερης ακρίβειας πρόβλεψης σωστών προβλέψεων στο πρόγραμμα που δεν επιτρέπει την επικάλυψη μεταξύ των γεωμετρικών σχημάτων αλλά και η μη αλληλοεπικάλυψη των δύο ζωνών που ορίζουν οι τυπικές αποκλίσεις.

3.3.2 Μεθοδολογία κατασκευής νευρωνικών δικτύων ταξινόμησης

Αρχικά κατασκευάζεται το απαιτούμενο δείγμα κάνοντας χρήση των αρχικών περιορισμών, ώστε να διασφαλιστεί ότι κανένα σχήμα δεν θα υπερβαίνει τα καθορισμένα όρια. Επίσης, μέσω ενός γεννήτορα ακέραιων αριθμών εντός του διαστήματος $(1, 5)$ καθορίζεται το πλήθος των τετραγώνων σε κάθε πλαίσιο.

Το μέγεθος του κάθε τετραγώνου ορίζεται να είναι $\boldsymbol{square\ size = 30 = \sigma\tau\alpha\theta}$.

Στη συνέχεια, ακολουθούνται οι αρχικοί περιορισμοί ως προς των σχηματισμό των τετραγώνων εντός των προκαθορισμένων ορίων.

Στο πρόγραμμα που δεν επιτρέπει την επικάλυψη ορίζεται η συνάρτηση ελέγχου επικάλυψης check_overlap , η οποία ελέγχει τα τετράγωνα ανά δύο και αναζητάει νέα θέση στην οποία θα τοποθετηθεί το νέο τετράγωνο αν εντοπιστεί επικάλυψη με το υπάρχον.

Ειδικότερα, η συνάρτηση ελέγχου επικάλυψης των σχημάτων ακολουθεί την παρακάτω λογική:

$$\boldsymbol{d = \sqrt{(x_i{}' - x_i)^2 + (y_i{}' - y_i)^2}}$$

$$\boldsymbol{tan\varphi} = \frac{\boldsymbol{y_i - y'_i}}{\boldsymbol{x_i - x'_i}} \Rightarrow \boldsymbol{\varphi} = \mathbf{tan^{-1}}\left(\frac{\boldsymbol{y_i - y'_i}}{\boldsymbol{x_i - x'_i}}\right)$$

$$\boldsymbol{cos\varphi} = \frac{\boldsymbol{square\ size}}{\boldsymbol{2b}} \Rightarrow \boldsymbol{b} = \frac{\boldsymbol{square\ size}}{\boldsymbol{cos\varphi}}$$

Η επικάλυψη μεταξύ των γεωμετρικών σχημάτων θα προκύπτει εάν:

$$\boldsymbol{d} < \frac{\boldsymbol{square\ size}}{\boldsymbol{cos\varphi}}$$

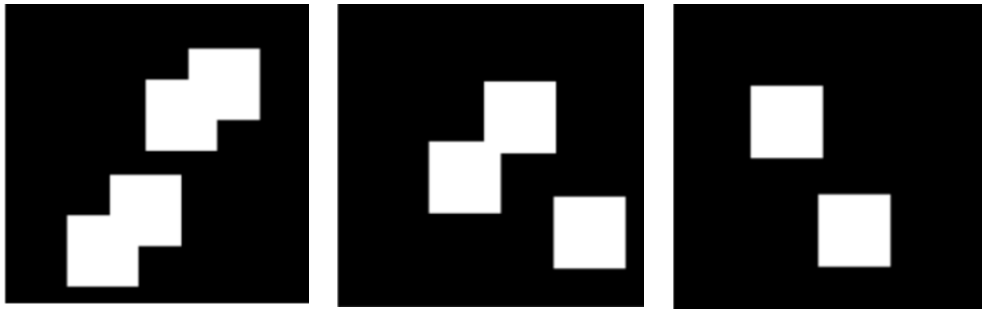
Όπου: $(\boldsymbol{x_i}, \boldsymbol{y_i})$: οι συντεταγμένες του τετραγώνου που προϋπήρχε, και

$(\boldsymbol{x'_i}, \boldsymbol{y'_i})$**:** οι συντεταγμένες του νέου τετραγώνου προς τοποθέτηση.

Στον αλγόριθμο στον οποίο η επικάλυψη μεταξύ των σχημάτων είναι αποδεκτή, η συνάρτηση ελέγχου επικάλυψης check_overlap παραλείπεται.

Ενδεικτικές εικόνες που παράγονται στον αλγόριθμο δίχως επικάλυψη (Non Overlap):

Ενδεικτικές εικόνες που παράγονται στον αλγόριθμο με επικάλυψη (Overlap):



Οι εικόνες, μαζί με τις ετικέτες τους, αποθηκεύονται σε αντίστοιχα ευρετήρια – directories που αξιοποιούνται από τον αλγόριθμο για την εύρεσή τους και έπειτα για την εκπαίδευσή του.

Ο αλγόριθμος καλείται να φορτώσει τις εικόνες δημιουργώντας κενές λίστες για τη αποθήκευση των εικόνων μαζί με τις αντίστοιχες ετικέτες, οι οποίες αντιπροσωπεύουν το πλήθος των σχημάτων σε κάθε εικόνα. Το σύνολο δεδομένων, συνολικού όγκου 3000 εικόνων, διαχωρίζεται σε training data και σε test data με αναλογία 75% και 25% αντίστοιχα.

Στο συγκεκριμένο πρόγραμμα χρησιμοποιείται η μέθοδος “One Hot Encoding”, η οποία μετατρέπει τις κατηγορίες των ετικετών σε δυαδικό πίνακα – μήτρα.

Αναλυτικότερα, μετατρέπει τα κατηγορηματικά δεδομένα, δηλαδή τις ετικέτες “1”, “2”, “3”, “4” σε δυαδικά διανύσματα.

Δημιουργούνται τέσσερις κατηγορίες – κλάσεις, μία για κάθε περίπτωση, με “1” να αντιστοιχεί σε ένα τετράγωνο, “2” για δύο τετράγωνα κ.ο.κ.

Οι κλάσεις στο σύστημα του υπολογιστή αναπαρίστανται ως δυαδικό διάνυσμα υπό τη μορφή πίνακα:

$$|\mathbf{1}\rangle = \begin{bmatrix}\mathbf{1}\\\mathbf{0}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}, \quad |\mathbf{2}\rangle = \begin{bmatrix}\mathbf{0}\\\mathbf{1}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}, \quad |\mathbf{3}\rangle = \begin{bmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{1}\\\mathbf{0}\end{bmatrix}, \quad |\mathbf{4}\rangle = \begin{bmatrix}\mathbf{0}\\\mathbf{0}\\\mathbf{0}\\\mathbf{1}\end{bmatrix}$$

Το σετ δεδομένων, συνολικού όγκου 3000 εικόνων, χωρίζεται σε training data (80% του αρχικού δείγματος) και σε test data (20% του αρχικού δείγματος).

Η εκπαίδευση του μοντέλου ολοκληρώνεται σε 20 κύκλους εκπαίδευσης (epochs) κατά την ολοκλήρωση των οποίων παρακολουθείται η εξέλιξη των μετρικών.

Οι μετρικές του συγκεκριμένου προγράμματος ορίστηκαν να είναι η ακρίβεια (accuracy) και η απώλεια (loss).

Ως συνάρτηση βελτιστοποίησης έχει οριστεί η συνάρτηση Adam.

Ως συνάρτηση απώλειας ορίζεται η Categorical Cross – Entropy, η οποία έχει λογαριθμική μορφή:

$$\boldsymbol{Loss} = -\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}}\sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{C}} \boldsymbol{y_{i,j}} \cdot \mathbf{log}\left(\widehat{\boldsymbol{y_{i,j}}}\right)$$

Όπου: $\boldsymbol{N}$ το συνολικό πλήθος δειγμάτων

$\boldsymbol{C}$ το συνολικό πλήθος κατηγοριών

$\boldsymbol{y_{i,j}}$ Η πραγματική ετικέτα του i δείγματος – εικόνας στην κατηγορία j, δηλαδή

$\boldsymbol{y_{i,j}} = \begin{cases}\mathbf{1}, & \boldsymbol{για\ σωστή\ ετικέτα}\\ \mathbf{0}, & \boldsymbol{για\ λάθος\ ετικέτα}\end{cases}$, με αυτόν τον τρόπο καθορίζεται ποια πιθανότητα χρησιμοποιείται για τον υπολογισμό της συνάρτησης απώλειας.

$\widehat{\boldsymbol{y_{i,j}}}$ η προβλεπόμενη πιθανότητα το μοντέλο να προβλέψει την κατηγορία j για το δείγμα – εικόνα i.

Καθώς η πιθανότητα πρόβλεψης της σωστής τάξης είναι 1.0, ζητούμενο είναι η απώλεια σε αυτή την περίπτωση να είναι 0. Αντίστοιχα, αν η πιθανότητα πρόβλεψης σωστής τάξης είναι χαμηλή, για παράδειγμα 0.01, η απώλεια σε αυτή την περίπτωση θα είναι αρκετά μεγάλη. Για τον λόγο, ο υπολογισμός της απώλειας εμπεριέχει τη λογαρίθμηση των πιθανοτήτων, ενώ το αρνητικό πρόσημο διατηρείται για να εξαχθεί θετική τιμή για την απώλεια.

Συμπερασματικά, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής, πρέπει η μετρική της απώλειας να μειώνεται εκθετικά και με πιο αργό ρυθμό όσο πλησιάζει σε ένα σημείο σταθεροποίησης. Το σημείο σταθεροποίησης θα πρέπει να τείνει προς το 0.

Παράλληλα, ορίζεται η μετρική της ακρίβειας:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Σωστές\ προβλέψεις}}{\boldsymbol{Συνολικό\ πλήθος\ δείγματος}}$$

Κατά τον υπολογισμό της από τον αλγόριθμο, προκύπτει ένας δεκαδικός αριθμός ο οποίος πολλαπλασιασμένος με το 100 δίνει την ποσοστιαία τιμή της ακρίβειας επί τοις εκατό.

Συμπερασματικά, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής η ακρίβεια πρέπει να αυξάνει προς την τιμή 1.0 με το πέρας των εποχών εκπαίδευσης.

3.3.3 Ανάλυση αρχιτεκτονικής νευρωνικών δικτύων

Και στους δυο αλγόριθμους το dataset χωρίστηκε με τον εξής τρόπο:

Σετ εκπαίδευσης (Training set): 75% των παραγόμενων εικόνων, δηλαδή 2250 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου.

Σετ δοκιμής (Test set): 25% των παραγόμενων εικόνων, δηλαδή 750 εικόνες χρησιμοποιήθηκαν για την αξιολόγηση του εκάστοτε μοντέλου.

Λόγω της πολυπλοκότητας του μοντέλου, τα δύο σετ δεν αναμειγνύονται, διασφαλίζοντας ότι η αξιολόγηση πραγματοποιείται σε άγνωστες για το μοντέλο εικόνες.

Ως προς την αρχιτεκτονική του μοντέλου, έγινε χρήση 4 επιπέδων συνέλιξης, 16 φίλτρα (3x3) στο πρώτο επίπεδο, 32 φίλτρα (3x3) στο δεύτερο επίπεδο και 64 φίλτρα (3x3) στα άλλα δύο με σκοπό την εξαγωγή χαρακτηριστικών από τις εικόνες.

Στο επίπεδο της υποδειγματοληψίας χρησιμοποιήθηκε ο αλγόριθμος MaxPooling2D για τη μείωση των διαστάσεων κάθε εικόνας με σκοπό την αποδοτικότερη εξαγωγή χαρακτηριστικών μέσω της μείωσης των διαστάσεων αυτών.

Το επίπεδο της υποδειγματοληψίας ακολουθείται από πλήρως συνδεδεμένα επίπεδα για την ταξινόμηση των εικόνων σε κατηγορίες. Στο πλήρως συνδεδεμένο επίπεδο, η συνάρτηση Flatten μετατρέπει τους δισδιάστατους πίνακες με τις εξαγωγές των χαρακτηριστικών σε μονοδιάστατο διάνυσμα, ενώ το επίπεδο Dense διαθέτει 128 νευρώνες οι οποίοι είναι πλήρως συνδεδεμένοι με όλους τους νευρώνες των προηγούμενων επιπέδων.

Χρησιμοποιήθηκε η τεχνική Dropout για να αποφευχθεί η υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης, όπου το 20% των νευρώνων απενεργοποιούνται σε κάθε στάδιο της εκπαίδευσης.

Ως συνάρτηση ενεργοποίησης ορίστηκε η ReLU για την εξαγωγή αφαιρετικών χαρακτηριστικών. Τέλος, ως έξοδος χρησιμοποιείται πάλι ένα επίπεδο Dense με 4 μόνο νευρώνες, που αντιστοιχούν στις 4 κατηγορίες που το μοντέλο έχει ταξινομήσει τις εικόνες. Το επίπεδο Dense χρησιμοποιείται ως έξοδος για την τελική ταξινόμηση κάθε

εικόνας καθώς είναι υπεύθυνο για τη σύνθεση όλων των χαρακτηριστικών που εξάγονται από τα προηγούμενα συνελικτικά επίπεδα.

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε σε 20 εποχές στις οποίες αξιολογούνταν τόσο η ακρίβεια, όσο και η συνάρτηση απώλειας του δικτύου κάνοντας χρήση της συνάρτησης βελτιστοποίησης Adam. Παράλληλα, ο ρυθμός μάθησης προσαρμόζεται κατάλληλα κατά τη διάρκεια της εκπαίδευσης για να επιτευχθούν βέλτιστα αποτελέσματα.

3.3.4 Αξιολόγηση μετρικών των νευρωνικών δικτύων

a. Αλγόριθμος χωρίς επικάλυψη μεταξύ των γεωμετρικών σχημάτων

Για το σετ της εκπαίδευσης, στην εποχή 1 η απώλεια έχει τιμή $1{,}282$, ενώ στην $20^{η}$ εποχή η τιμή είναι της τάξης $0{,}136$ καθώς από το διάγραμμα φαίνεται να σταθεροποιείται στην $16^{η}$ εποχή. Αντίστοιχα, για το σετ της επαλήθευσης, στην $1^{η}$ εποχή η απώλεια έχει τιμή $1{,}321$ ενώ στην τελευταία φτάνει την τιμή $0{,}058$, ενώ φαίνεται να σταθεροποιείται στην $16^{η}$ εποχή. Για την πορεία της μετρικής της ακρίβειας, στην $1^{η}$ εποχή είναι της τάξης του $0{,}3604$ για τα δεδομένα εκπαίδευσης ενώ για τα δεδομένα επαλήθευσης είναι της τάξης του $0{,}452$, ενώ στην $20^{η}$ εποχή οι τιμές της ακρίβειας είναι $0{,}96$ και $1{,}00$ αντίστοιχα.

Ο ρυθμός μάθησης μειώθηκε στην $18^{η}$ εποχή από 10^{-5} σε $\sim 3 \cdot 10^{-6}$ προκειμένου να ελαχιστοποιηθεί περαιτέρω η τιμή της συνάρτησης απώλειας. Συνολικά, στις εικόνες αξιολόγησης του μοντέλου, η συνάρτηση της απώλειας είναι της τάξης του $0{,}058$ ενώ η ακρίβεια φτάνει την τιμή $1{,}00$.

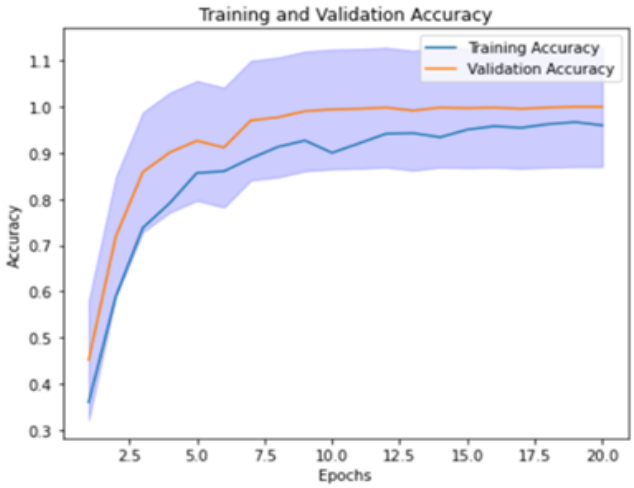


Figure 12: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται ανοδική πορεία και σταθεροποίηση στην $16^{η}$ εποχή

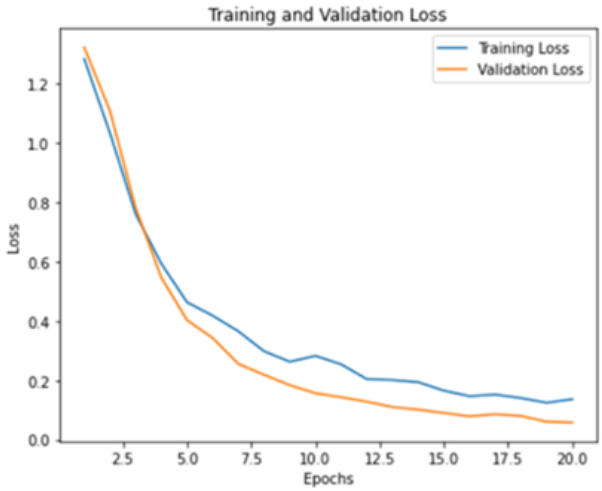


Figure 13: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται συνεχής μείωση και έπειτα σταθεροποίηση στην $16^{η}$ εποχή

Στα αντίστοιχα διαγράμματα είναι εμφανής τόσο η εκθετική μείωση της μετρικής της απώλειας και για τα δύο σετ στις πρώτες εποχές και η ελαχιστοποίηση και έπειτα σταθεροποίηση της κατά το τέλος των κύκλων επανάληψης της εκπαίδευσης. Επίσης, και για τα δύο σετ δεδομένων οι τιμές της ακρίβειας ξεκινούν από χαμηλές τιμές και αυξάνονται με γρήγορο ρυθμό φτάνοντας σε σταθερές και μέγιστες τιμές. Οι διαφορές των μετρικών των παραπάνω σετ δεδομένων οφείλονται στο μικρότερο πλήθος δεδομένων που περιλαμβάνει το σετ της επαλήθευσης και στο γεγονός ότι στα δεδομένα επαλήθευσης το μοντέλο απλώς αξιολογείται χωρίς να προσαρμόζει τις παραμέτρους του.

Γενικά, η απόδοση του μοντέλου στην ταξινόμηση των εικόνων σε 4 κατηγορίες ανάλογα με το πλήθος των ίδιων γεωμετρικών σχημάτων που περιλαμβάνουν μπορεί να θεωρηθεί επιτυχής.

b. Αλγόριθμος με επιτρεπτή την επικάλυψη μεταξύ των γεωμετρικών σχημάτων

Το μοντέλο στην συγκεκριμένη περίπτωση ξεκινάει με χαμηλή ακρίβεια και υψηλή απώλεια και για τα δύο σετ δεδομένων. Ειδικότερα, στην $1^{η}$ εποχή, η απώλεια του σετ της εκπαίδευσης είναι $1{,}339$ και η ακρίβεια είναι της τάξης του $0{,}296$, ενώ για το σετ επαλήθευσης οι τιμές είναι $1{,}367$ και $0{,}313$ αντίστοιχα. Στην $20^{η}$ εποχή, οι τιμές των απωλειών έχουν μειωθεί σε $0{,}307$ για τα δεδομένα εκπαίδευσης και $0{,}408$ για τα δεδομένα επαλήθευσης. Αντίστοιχα, η ακρίβεια στα δεδομένα εκπαίδευσης φτάνει το $0{,}894$ στην τελευταία εποχή και η ακρίβεια στα δεδομένα επαλήθευσης φτάνει το $0{,}825$. Συνολικά, στις εικόνες αξιολόγησης του μοντέλου, η συνάρτηση της απώλειας είναι της τάξης του $0{,}408$ ενώ η ακρίβεια φτάνει την τιμή $0{,}825$.

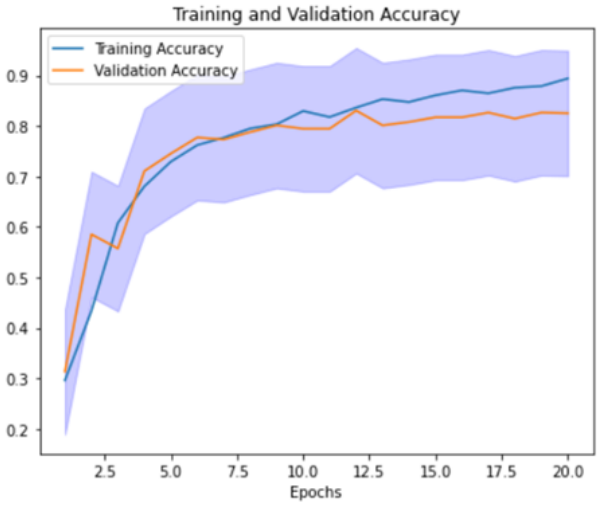


Figure 14: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται ανοδική πορεία και σταθεροποίηση στις τελευταίες εποχές

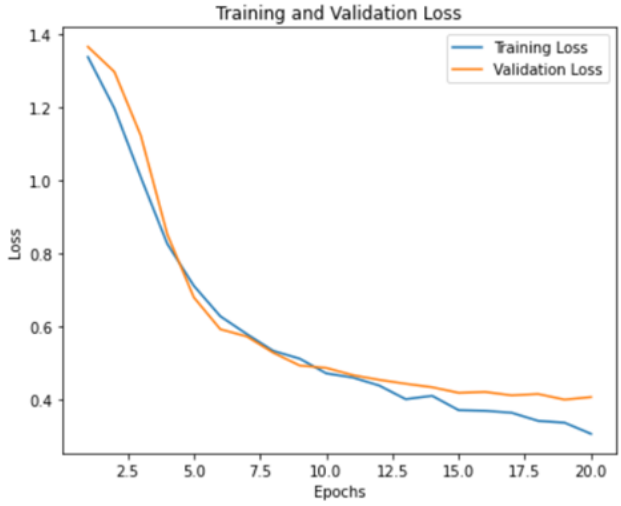


Figure 15: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται συνεχής μείωση και έπειτα σταθεροποίηση στην $15^{η}$ εποχή

Στα αντίστοιχα διαγράμματα παρατηρείται σταθερή βελτίωση στην ακρίβεια η οποία στις τελευταίες εποχές φαίνεται να σταθεροποιείται. Η συνάρτηση της απώλειας και στα δυο σετ δεδομένων μειώνεται με εκθετικό ρυθμό και φαίνεται να σταθεροποιείται μετά την 15η εποχή, γεγονός που δεν υποδεικνύει υπερπροσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης. Οι διαφορές των μετρικών των παραπάνω σετ δεδομένων οφείλονται στο μικρότερο πλήθος δεδομένων που περιλαμβάνει το σετ της επαλήθευσης και στο γεγονός ότι στα δεδομένα επαλήθευσης το μοντέλο απλώς αξιολογείται χωρίς να προσαρμόζει τις παραμέτρους του.

Γενικά φαίνεται πως το μοντέλο γενικεύει σωστά, με τον ρυθμό μάθησης να παραμένει σταθερός καθ' όλη τη διάρκεια της εκπαίδευσης, υποδεικνύοντας ότι το μοντέλο μαθαίνει με σταθερό και αργό ρυθμό αποφεύγοντας μεγάλες διακυμάνσεις στις τιμές της απώλειας. Η πορεία αυτή καθιστά την εκπαίδευση του μοντέλου επιτυχή.

Συνολικά, η απόδοση του μοντέλου στην ταξινόμηση των εικόνων σε 4 κατηγορίες ανάλογα με το πλήθος των ίδιων γεωμετρικών σχημάτων που περιλαμβάνουν μπορεί να θεωρηθεί επιτυχής παρά την αυξημένη πολυπλοκότητα του προβλήματος λόγω της επιτρεπτής επικάλυψης μεταξύ των γεωμετρικών σχημάτων.

3.3.5 Σύγκριση συνολικής απόδοσης των δύο αλγόριθμων

Παρακάτω παρατίθεται το κοινό διάγραμμα της εκάστοτε ακρίβειας των δύο μοντέλων με τις αντίστοιχες τυπικές αποκλίσεις:

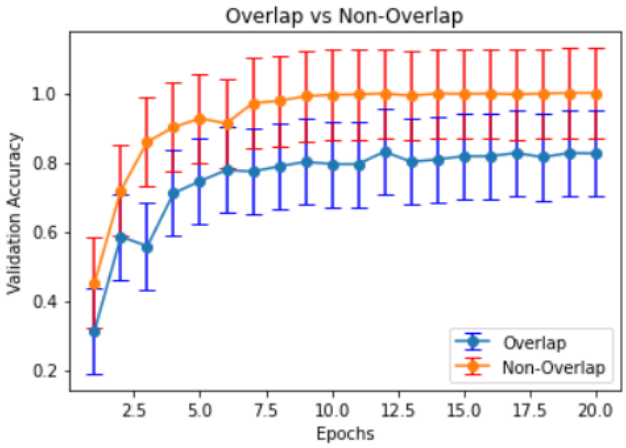


Figure 16: Διάγραμμα των ακριβειών επαλήθευσης με τις αντίστοιχες τυπικές αποκλίσεις για τα δύο νευρωνικά δίκτυα.

Από το διάγραμμα αντιδιαστολής των τυπικών αποκλίσεων της ακρίβειας για τα δεδομένα επαλήθευσης των δύο αλγορίθμων είναι εμφανής η υψηλότερη ακρίβεια του αλγόριθμου ταξινόμησης που δεν επιτρέπει την επικάλυψη μεταξύ των γεωμετρικών σχημάτων σε σχέση με τον αλγόριθμο ταξινόμησης που επιτρέπει την επικάλυψη.

Παρατηρείται μεγαλύτερη διακύμανση της τυπικής απόκλισης για την καμπύλη Overlap στις πρώτες εποχές, υποδεικνύοντας αστάθεια στην εκπαίδευση. Η αστάθεια αυτή οφείλεται στην αυξημένη πολυπλοκότητα των εικόνων που καλείται το μοντέλο να ταξινομήσει. Και οι δύο καμπύλες σταθεροποιούνται μετά από 10 – 12 εποχές.

Στις πρώτες 5 εποχές οι τυπικές αποκλίσεις είναι μεγαλύτερες και για τα δύο μοντέλα, ενώ με την πάροδο των εποχών μειώνονται. Παρατηρείται εντονότερη μείωση της τυπικής απόκλισης για το μοντέλο Non Overlap, γεγονός που υποδηλώνει ότι το μοντέλο έχει μάθει να προβλέπει με υψηλή ακρίβεια την κατηγορία στην οποία ανήκει η κάθε εικόνα σε αντίθεση με το μοντέλο Overlap που δεν υπάρχει τέτοια μείωση στην τυπική απόκλιση με το πέρας των εποχών.

Οι ζώνες των τυπικών αποκλίσεων είναι ξεκάθαρα διαχωρισμένες στις περισσότερες εποχές. Αυτό σημαίνει πως το μοντέλο το οποίο τροφοδοτείται με λιγότερο πολύπλοκα δεδομένα αποδίδει πιο αποτελεσματικά και μπορεί να αποδώσει πιο σταθερά από όταν τα δεδομένα είναι περισσότερο πολύπλοκα, όταν δηλαδή περιλαμβάνουν επικάλυψη μεταξύ των γεωμετρικών σχημάτων. Συνολικά, η σαφής διαφορά στις τυπικές αποκλίσεις και η μη επικάλυψη των δύο ζωνών υπογραμμίζει την ανώτερη απόδοση του μοντέλου που τροφοδοτείται με εικόνες χωρίς επικάλυψη των γεωμετρικών σχημάτων .

Συμπερασματικά, αυξανομένης της πολυπλοκότητας των δεδομένων, αυξάνεται και η δυσκολία που αντιμετωπίζει το μοντέλο στην εκμάθηση και έπειτα στην σωστή πρόβλεψη των κατηγοριών, όπως παρουσιάστηκε παραπάνω στην ανάλυση του εκάστοτε αλγόριθμου.

3.4 **3ο Πρόβλημα Ταξινόμησης**

Ταξινόμηση εικόνων ως προς την αλληλοεπικάλυψη δύο όμοιων απλών γεωμετρικών σχημάτων μεταβλητών διαστάσεων

3.4.1 Σκοπός του προβλήματος ταξινόμησης

Στο συγκεκριμένο πρόβλημα κατασκευάζεται ένα δείγμα 3000 εικόνων, η καθεμία διαστάσεων 128x128 που περιλαμβάνει δύο κύκλους μεταβλητών ακτίνων με επιτρεπτή την αλληλοεπικάλυψη. Ζητούμενη είναι η δυαδική ταξινόμηση των εικόνων ως προς την επικάλυψη των γεωμετρικών σχημάτων.

Στόχο αποτελεί η επίτευξη βέλτιστης απόδοσης, δηλαδή σωστή πρόβλεψη των ετικετών και η ελαχιστοποίηση της συνάρτησης απώλειας του αλγόριθμου.

3.4.2 Μεθοδολογία κατασκευής νευρωνικού δικτύου ταξινόμησης

Αρχικά, κατασκευάζεται το απαιτούμενο δείγμα κάνοντας χρήση των περιορισμών που ορίστηκαν στην αρχή, ώστε να διασφαλιστεί ότι κανένα σχήμα δεν θα υπερβαίνει τα καθορισμένα όρια. Όλα τα στοιχεία του πίνακα τίθενται ίσα με το 0.

Ορίζονται οι τιμές για τη μέγιστη και την ελάχιστη ακτίνα που μπορεί να έχει ένας κύκλος:

$$\begin{cases} \mathbf{min\_radius} = \mathbf{5} \\ \mathbf{max\_radius} = \mathbf{25} \end{cases}$$

Η ακτίνα κάθε κύκλου ορίζεται ως ένας ακέραιος μέσω γεννήτορα τυχαίων ακεραίων αριθμών στο διάστημα $[\mathbf{min\_radius}, \mathbf{max\_radius}]$.

$\boldsymbol{center\_x1}, \boldsymbol{center\_y1} \in [\boldsymbol{radius1}, (\boldsymbol{size} - \boldsymbol{radius1})]$

$\boldsymbol{center\_x2}, \boldsymbol{center\_2} \in [\boldsymbol{radius2}, (\boldsymbol{size} - \boldsymbol{radius2})]$

Τα σημεία που ανήκουν στις εξισώσεις κύκλου, δηλαδή που ικανοποιούν τη συνθήκη:

$(\boldsymbol{x_i} - \boldsymbol{center\ x_i})^2 + (\boldsymbol{y_i} - \boldsymbol{center\ y_i})^2 \leq \boldsymbol{radius_i^2}$, τίθενται ίσα με 1.

Η επιλογή αυτή, στην οπτικοποίηση των εικόνων, ισοδυναμεί στον χρωματισμό των κύκλων με λευκό χρώμα ενώ στον υπόλοιπο χώρο με μαύρο χρώμα.

Ενδεικτικές εικόνες που παράγονται:

Για τον έλεγχο επικάλυψης μεταξύ των δύο σχημάτων ακολουθείται η παρακάτω λογική:

Ορίζεται η απόσταση μεταξύ των κέντρων των δύο κύκλων:

$$\boldsymbol{d = \sqrt{(center\ x2 - center\ x1)^2 + (center\ y2 - center\ y1)^2}}$$

Η επικάλυψη θα προκύπτει εάν:

$$\boldsymbol{d \leq (radius1 + radius2)}$$

Ενώ διαφορετικά οι δύο κύκλοι δεν επικαλύπτονται.

Οι εικόνες αποθηκεύονται σε αντίστοιχα ευρετήρια – directories μαζί με τις αντίστοιχες δυαδικές ετικέτες, την ετικέτα 0 εάν δεν επικαλύπτονται τα δύο γεωμετρικά σχήματα και την ετικέτα 1 εάν επικαλύπτονται.

Το σετ δεδομένων, συνολικού όγκου 3000 εικόνων, χωρίζεται σε training data (80% του αρχικού δείγματος) και σε test data (20% του αρχικού δείγματος). Η εκπαίδευση του μοντέλου ολοκληρώνεται σε 30 κύκλους εκπαίδευσης (epochs), κατά την εξέλιξη των οποίων παρατηρείται η εξέλιξη των μετρικών.

Οι μετρικές του συγκεκριμένου προγράμματος ορίστηκαν να είναι η ακρίβεια (accuracy) και η απώλεια (loss). Καθώς πρόκειται για πρόβλημα δυαδικής ταξινόμησης, ως συνάρτηση της απώλειας ορίζεται να είναι η λογαριθμική συνάρτηση της απώλειας, ή αλλιώς Binary Cross - Entropy Loss:

$\boldsymbol{Loss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log\left(p(y_i)\right) + (1 - y_i) \cdot \log\left(1 - p(y_i)\right)\right]}$, όπου:

$\boldsymbol{y}$: είναι η ετικέτα, 0 για τις εικόνες που περιέχουν μη επικαλυπτόμενους κύκλους και 1 για εκείνες που περιέχουν αλληλοεπικάλυψη μεταξύ των δύο όμοιων γεωμετρικών σχημάτων.

$\boldsymbol{p(y)}$**:** η προβλεπόμενη πιθανότητα για κάθε εικόνα να περιλαμβάνει δύο κύκλους μεταβλητών ακτίνων οι οποίοι επικαλύπτονται μεταξύ τους

$\boldsymbol{N}$: το πλήθος των εικόνων

Εν γένει, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής, πρέπει η μετρική της απώλειας να ακολουθεί αρνητική λογαριθμική πορεία και τείνει προς το 0.

Για τον υπολογισμό της ακρίβειας ακολουθείται το εξής σενάριο:

- Κατηγορία 1 (Positive, label =1): Η εικόνα περιέχει αλληλοεπικαλυπτόμενα σχήματα.
- Κατηγορία 2 (Negative, label =0): Η εικόνα περιέχει μη αλληλοεπικαλυπτόμενα σχήματα.

Κατά την εκπαίδευση του μοντέλου δημιουργούνται 4 νέες καταστάσεις:

True Positive (TP): Η εικόνα περιλαμβάνει επικάλυψη μεταξύ των σχημάτων και το μοντέλο την ταξινόμησε σωστά ως προς την επικάλυψη αυτών.

- True Negative (TN): Η εικόνα δεν περιλαμβάνει επικάλυψη μεταξύ των σχημάτων και το μοντέλο την ταξινόμησε σωστά ως προς αυτό.

- False Positive (FP): Η εικόνα δεν περιλαμβάνει επικάλυψη των σχημάτων και το μοντέλο την ταξινόμησε λανθασμένα ότι υπάρχει αλληλοεπικάλυψη.
- False Negative (FN): Η εικόνα περιλαμβάνει επικάλυψη μεταξύ των σχημάτων και το μοντέλο την ταξινόμησε λανθασμένα ότι δεν υπάρχει επικάλυψη.

Πραγματική κατάσταση	Πρόβλεψη μοντέλου	Κατηγορία
Αλληλοεπικάλυψη (label =1)	Αλληλοεπικάλυψη	True Positive (TP)
Αλληλοεπικάλυψη (label =1)	Μη Αλληλοεπικάλυψη	False Negative (FN)
Μη Αλληλοεπικάλυψη (label =0)	Μη Αλληλοεπικάλυψη	True Negative (TN)
Μη Αλληλοεπικάλυψη (label =0)	Αλληλοεπικάλυψη	False Positive (FP)

Table 3: Συγκεντρωτικός πίνακας με τις πραγματικές καταστάσεις του 3ου προβλήματος ταξινόμησης σε αντιδιαστολή με τις προβλεπόμενες και οι κατηγορίες που δημιουργούνται.

Έτσι, ορίζεται η ακρίβεια (accuracy) ως: $\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}$

Κατά τον υπολογισμό της από τον αλγόριθμο, προκύπτει ένας δεκαδικός αριθμός ο οποίος πολλαπλασιασμένος με το 100 δίνει την τιμή της απόδοσης επί τοις εκατό.

Συμπερασματικά, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής η απόδοση πρέπει να αυξάνει προς την τιμή 1.0 με το πέρας των εποχών.

3.4.3 Ανάλυση αρχιτεκτονικής νευρωνικού δικτύου

Το dataset χωρίστηκε με τον εξής τρόπο:

Σετ εκπαίδευσης (Training set): 80% των παραγόμενων εικόνων, δηλαδή 2400 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου.

Σετ επαλήθευσης (Validation set): 20% των παραγόμενων εικόνων, δηλαδή 600 εικόνες χρησιμοποιήθηκαν για την επικύρωση του μοντέλου.

Τα δύο σετ δεν αναμειγνύονται προκειμένου η αξιολόγηση να γίνει πάνω σε άγνωστες για το μοντέλο εικόνες.

Ως προς την αρχιτεκτονική του μοντέλου, χρησιμοποιήθηκαν 5 επίπεδα συνέλιξης, 32 φίλτρα (3x3) στο πρώτο επίπεδο, 64 φίλτρα στο δεύτερο και στο τρίτο επίπεδο επιτρέποντας την εξαγωγή περαιτέρω χαρακτηριστικών και 128 φίλτρα στο τέταρτο και στο πέμπτο επίπεδο. Η χρήση περισσότερων φίλτρων έχει ως αποτέλεσμα ένα πιο ισχυρό μοντέλο με κίνδυνο υπερπροσαρμογής λόγω του αυξημένου αριθμού παραμέτρων, η οποία αποφεύγεται με χρήση της τεχνικής Dropout, όπου 20% των νευρώνων απενεργοποιούνται σε κάθε στάδιο της εκπαίδευσης.

Στο επίπεδο της υποδειγματοληψίας χρησιμοποιήθηκε ο αλγόριθμος MaxPooling2D για τη μείωση των διαστάσεων κάθε εικόνας με σκοπό την αποδοτικότερη εξαγωγή χαρακτηριστικών.

Επιπρόσθετα, στο πλήρως συνδεδεμένο επίπεδο , η συνάρτηση Flatten μετατρέπει τους δισδιάστατους πίνακες με τις εξαγωγές των χαρακτηριστικών σε μονοδιάστατο διάνυσμα, ενώ το επίπεδο Dense διαθέτει 128 νευρώνες οι οποίοι είναι πλήρως συνδεδεμένοι με όλους τους νευρώνες των προηγούμενων επιπέδων και ενισχύει την

εκμάθηση μη γραμμικών σχέσεων, συνθέτει δηλαδή όλα τα χαρακτηριστικά που εξάγονται από τα προηγούμενα επίπεδα συνέλιξης.

Ως συνάρτηση ενεργοποίησης ορίστηκε η ReLU για την εξαγωγή αφαιρετικών χαρακτηριστικών. Τέλος, ως έξοδος ορίστηκε ένα επίπεδο Dense με έναν μόνο νευρώνα με ενεργοποίηση sigmoid η οποία επιτρέπει τη δυαδική ταξινόμηση.

3.4.4 Αξιολόγηση μετρικών του νευρωνικού δικτύου

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε σε 30 εποχές στις οποίες αξιολογούνταν τόσο η ακρίβεια, όσο και η συνάρτηση απώλειας του δικτύου κάνοντας χρήση της συνάρτησης βελτιστοποίησης Adam, ενώ ο ρυθμός μάθησης προσαρμόζεται κατάλληλα καθ' όλη τη διάρκεια εκπαίδευσης. Πιο συγκεκριμένα, ο ρυθμός μάθησης μειώνεται κατά 30% εάν η πορεία της απώλειας δεν έχει βελτιωθεί σε δύο εποχές.

Αναφορικά με την εξέλιξη των μετρικών, για το σετ δεδομένων εκπαίδευσης, στην $1^η$ εποχή η απώλεια έχει τιμή $0{,}499$, ενώ στην $30^η$ εποχή έχει τιμή $0{,}042$ και από το διάγραμμα φαίνεται να σταθεροποιείται μετά την $22^η$ εποχή. Αντίστοιχα, για το σετ δεδομένων επαλήθευσης, στην $1^η$ εποχή η τιμή της λογαριθμικής απώλειας έχει τιμή $0{,}6361$, στην $30^η$ εποχή φτάνει την τιμή $0{,}118$, ενώ φαίνεται να σταθεροποιείται στην $24^η$ εποχή. Όσον αφορά τη μετρική της ακρίβειας, στην $1^η$ εποχή είναι της τάξης $0{,}770$ για τα δεδομένα εκπαίδευσης, ενώ για τα δεδομένα επαλήθευσης είναι της τάξης του $0{,}782$, ενώ στην $30^η$ εποχή οι τιμές της είναι $0{,}988$ και $0{,}957$ αντίστοιχα. Η σταθεροποίηση της μετρικής της ακρίβειας και για τα δύο σετ δεδομένων παρατηρείται μετά την $28^η$ εποχή.

Ο ρυθμός μάθησης μειώθηκε στην $14^η$ εποχή από $4 \cdot 10^{-5}$ σε $\sim 1{,}2 \cdot 10^{-5}$ και μειώθηκε περαιτέρω στην $26^η$ εποχή στην τιμή $\sim 3{,}6 \cdot 10^{-6}$ προκειμένου να ελαχιστοποιηθεί περισσότερο η τιμή της συνάρτησης απώλειας.

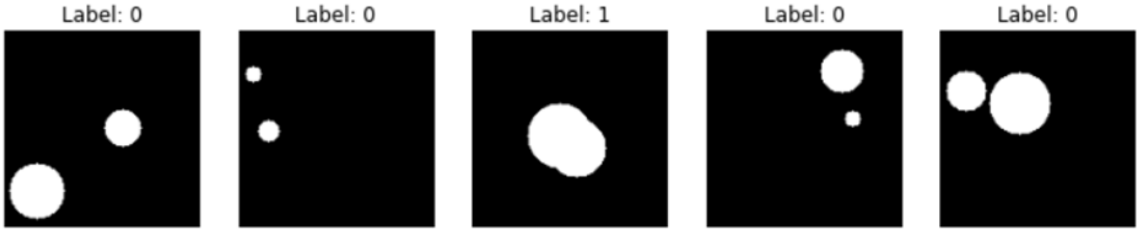
Συνολικά, το μοντέλο δυαδικής ταξινόμησης επιτυγχάνει ακρίβεια ορθών προβλέψεων 96% και η λογαριθμική συνάρτηση της απώλειας φτάνει την τιμή $0{,}118$.

Figure 17: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται ανοδική πορεία και σταθεροποίηση μετά την $28^η$ εποχή και για τα δύο σετ δεδομένων.

Figure 19: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται συνεχής μείωση και σταθεροποίηση στην $22^η$ και στην $24^η$ εποχή για το σετ εκπαίδευσης και το σετ επαλήθευσης αντίστοιχα.

Στα αντίστοιχα διαγράμματα είναι εμφανής τόσο η εκθετική μείωση της μετρικής της απώλειας και για τα δύο σετ δεδομένων κατά τις πρώτες εποχές και η μετέπειτα σταθεροποίηση τους κατά το τέλος των κύκλων επανάληψης της εκπαίδευσης επιδεικνύοντας πως το μοντέλο δεν έχει παρουσιάσει υπερπροσαρμογή στα δεδομένα εκπαίδευσης. Επίσης, και για τα δύο σετ δεδομένων, οι τιμές της ακρίβειας ξεκινούν από χαμηλές τιμές και αυξάνονται ταχέως φτάνοντας μέγιστες, σταθερές τιμές. Οι διαφορές των μετρικών των παραπάνω σετ δεδομένων οφείλονται στο μικρότερο πλήθος δεδομένων που περιλαμβάνει το σετ της επαλήθευσης και στο γεγονός ότι στα δεδομένα επαλήθευσης το μοντέλο απλώς αξιολογείται χωρίς να προσαρμόζει τις παραμέτρους του.

Γενικά, η απόδοση του μοντέλου στην δυαδική ταξινόμηση των εικόνων ως προς την αλληλοεπικάλυψη των όμοιων γεωμετρικών σχημάτων διαφορετικών μεγεθών μπορεί να θεωρηθεί επιτυχής.



Εικόνα 20: Τυχαίες εικόνες από το σετ επαλήθευσης με τις προβλεπόμενες ετικέτες τους

3.5 **4ο Πρόβλημα Ταξινόμησης**

Ταξινόμηση εικόνων ως προς την αλληλοεπικάλυψη δύο απλών γεωμετρικών σχημάτων μεταβλητών διαστάσεων

3.5.1 Σκοπός του προβλήματος ταξινόμησης

Στο παρόν πρόβλημα κατασκευάζεται ένα δείγμα 3000 εικόνων, η καθεμία διαστάσεων 128x128 που περιλαμβάνουν ένα τετράγωνο και έναν κύκλο μεταβλητών διαστάσεων τα οποία μπορούν να επικαλύπτονται μεταξύ τους.

Στόχο αποτελεί η κατασκευή ενός μοντέλου μηχανικής μάθησης που θα εκτελεί επιτυχώς τη δυαδική ταξινόμηση των παραγόμενων εικόνων ως προς την αλληλοεπικάλυψη των απλών γεωμετρικών σχημάτων. Ζητούμενη είναι η επίτευξη βέλτιστης ακρίβειας ως προς τη σωστή ταξινόμηση των εικόνων και η ελαχιστοποίηση της συνάρτησης κόστους του αλγόριθμου.

3.5.2 Μεθοδολογία κατασκευής νευρωνικού δικτύου ταξινόμησης

Αρχικά, ορίζονται οι συναρτήσεις κατασκευής κύκλων και τετραγώνων τηρώντας τους αρχικούς περιορισμούς, διασφαλίζοντας ότι όλα τα παραγόμενα γεωμετρικά σχήματα θα παραμείνουν εντός ορίων. Επίσης, ορίζονται όπως παρακάτω οι συντεταγμένες κέντρου κάθε γεωμετρικού σχήματος ως:

- Κέντρο κύκλου:
  $\begin{cases} \boldsymbol{center\ x_1} \in (\mathbf{min\_}\ \boldsymbol{radius}, \mathbf{max\_}\boldsymbol{radius}) \\ \boldsymbol{center\ y_1} \in (\mathbf{min\_}\ \boldsymbol{radius}, \mathbf{max\_}\boldsymbol{radius}) \end{cases}$, όπου $\boldsymbol{center\ x_1}, \boldsymbol{center\ y_1} \in \mathbb{Z}$
  Η επιλογή των συντεταγμένων του κέντρου του κύκλου γίνεται με γεννήτορα τυχαίων αριθμών στο συγκεκριμένο διάστημα.

- Κέντρο τετραγώνου:
  $\begin{cases} \boldsymbol{center\ x_2} = \frac{\boldsymbol{left\ edge + right\ edge}}{\mathbf{2}} \\ \boldsymbol{center\ y_2} = \frac{\boldsymbol{bottom\ edge + top\ edge'}}{\mathbf{2}} \end{cases}$ όπου $\boldsymbol{center\ x_2}, \boldsymbol{center\ y_2} \in \mathbb{Z}$
  λόγω των μεταβλητών που ορίστηκαν αρχικά.

Στη συνέχεια, κατασκευάζεται το απαιτούμενο δείγμα 3000 εικόνων, η καθεμία εκ των οποίων περιλαμβάνει έναν κύκλο και ένα τετράγωνο μεταβλητών διαστάσεων. Ο έλεγχος των εικόνων ως προς την ύπαρξη επικάλυψης μεταξύ των σχημάτων ακολουθεί την παρακάτω λογική:

$$\boldsymbol{\alpha} = \begin{cases} \frac{\boldsymbol{\pi}}{\mathbf{2}}, & \boldsymbol{center\ x_2} = \boldsymbol{center\ x_1} \\ \frac{\boldsymbol{center\ y_2} - \boldsymbol{center\ y_1}}{\boldsymbol{center\ x_2} - \boldsymbol{center\ x_1}}, & \boldsymbol{center\ x_2} \neq \boldsymbol{center\ x_1} \end{cases}$$

Ορίζεται $\mathbf{tan}(\boldsymbol{\varphi}) = \mathbf{tan}(\boldsymbol{\alpha}) \Rightarrow \boldsymbol{\varphi} = \mathbf{tan}^{-1}(\boldsymbol{\alpha})$

Επίσης, $\boldsymbol{b} = \frac{\mathbf{cos}(\boldsymbol{\varphi})\cdot(\boldsymbol{square\ size})}{\mathbf{2}}$

Ορίζεται επίσης η απόσταση μεταξύ των κέντρων των δύο σχημάτων:

$$\boldsymbol{d} = \sqrt{(\boldsymbol{center\ x_2} - \boldsymbol{center\ x_1})^2 + (\boldsymbol{center\ y_2} - \boldsymbol{center\ y_1})^2}$$

Επικάλυψη μεταξύ των δύο γεωμετρικών σχημάτων υφίσταται εάν $\boldsymbol{d} < (\boldsymbol{radius} + \boldsymbol{b})$ ενώ διαφορετικά δεν υπάρχει επικάλυψη.

Ενδεικτικές εικόνες που παράγονται:

Οι εικόνες αποθηκεύονται σε αντίστοιχα ευρετήρια – directories μαζί με τις αντίστοιχες δυαδικές ετικέτες, την ετικέτα 0 εάν δεν επικαλύπτονται τα δύο γεωμετρικά σχήματα και την ετικέτα 1 εάν επικαλύπτονται.

Το σετ δεδομένων, συνολικού όγκου 3000 εικόνων, χωρίζεται σε training data (80% του αρχικού δείγματος) και σε test data (20% του αρχικού δείγματος).

Η εκπαίδευση του μοντέλου ολοκληρώνεται σε 30 κύκλους εκπαίδευσης (epochs), κατά την εξέλιξη των οποίων παρατηρείται η εξέλιξη των μετρικών. Οι μετρικές του συγκεκριμένου προγράμματος ορίστηκαν να είναι η ακρίβεια (accuracy) και η απώλεια (loss).

Καθώς πρόκειται για πρόβλημα δυαδικής ταξινόμησης, ως συνάρτηση της απώλειας ορίζεται να είναι η λογαριθμική συνάρτηση της απώλειας, ή αλλιώς Binary Cross - Entropy Loss:

$\boldsymbol{Loss} = -\frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} [\boldsymbol{y_i} \cdot \mathbf{log}(\boldsymbol{p}(\boldsymbol{y_i})) + (\mathbf{1} - \boldsymbol{y_i}) \cdot \mathbf{log}(\mathbf{1} - \boldsymbol{p}(\boldsymbol{y_i}))]$, όπου:

$\boldsymbol{y}$: είναι η ετικέτα, 0 για τις εικόνες που περιέχουν μη επικαλυπτόμενα γεωμετρικά σχήματα και 1 για εκείνες που περιέχουν αλληλοεπικάλυψη μεταξύ των δύο γεωμετρικών σχημάτων.

$\boldsymbol{p}(\boldsymbol{y})$**:** η προβλεπόμενη πιθανότητα για κάθε εικόνα να περιλαμβάνει ένα τετράγωνο και έναν κύκλο μεταβλητών μεγεθών οι οποίοι επικαλύπτονται μεταξύ τους

$\boldsymbol{N}$: το πλήθος των εικόνων

Εν γένει, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής, πρέπει η μετρική της απώλειας να ακολουθεί αρνητική λογαριθμική πορεία και τείνει προς το 0.

Για τον υπολογισμό της ακρίβειας ακολουθείται το εξής σενάριο:

- Κατηγορία 1 (Positive, label =1): Η εικόνα περιέχει αλληλοεπικαλυπτόμενα σχήματα.
- Κατηγορία 2 (Negative, label =0): Η εικόνα περιέχει μη αλληλοεπικαλυπτόμενα σχήματα.

Κατά την εκπαίδευση του μοντέλου δημιουργούνται 4 νέες καταστάσεις:

- True Positive (TP): Η εικόνα περιλαμβάνει επικάλυψη μεταξύ των σχημάτων και το μοντέλο την ταξινόμησε σωστά ως προς την επικάλυψη αυτών.
- True Negative (TN): Η εικόνα δεν περιλαμβάνει επικάλυψη μεταξύ των σχημάτων και το μοντέλο την ταξινόμησε σωστά ως προς αυτό.
- False Positive (FP): Η εικόνα δεν περιλαμβάνει επικάλυψη των σχημάτων και το μοντέλο την ταξινόμησε λανθασμένα ότι υπάρχει αλληλοεπικάλυψη.
- False Negative (FN): Η εικόνα περιλαμβάνει επικάλυψη μεταξύ των σχημάτων και το μοντέλο την ταξινόμησε λανθασμένα ότι δεν υπάρχει επικάλυψη.

Πραγματική κατάσταση	Πρόβλεψη μοντέλου	Κατηγορία
Αλληλοεπικάλυψη (label =1)	Αλληλοεπικάλυψη	True Positive (TP)
Αλληλοεπικάλυψη (label =1)	Μη Αλληλοεπικάλυψη	False Negative (FN)
Μη Αλληλοεπικάλυψη (label =0)	Μη Αλληλοεπικάλυψη	True Negative (TN)
Μη Αλληλοεπικάλυψη (label =0)	Αλληλοεπικάλυψη	False Positive (FP)

Table 4: Συγκεντρωτικός πίνακας με τις πραγματικές καταστάσεις του 4ου προβλήματος ταξινόμησης σε αντιδιαστολή με τις προβλεπόμενες και οι κατηγορίες που δημιουργούνται.

Έτσι, ορίζεται η ακρίβεια (accuracy) ως: $\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}$

Κατά τον υπολογισμό της από τον αλγόριθμο, προκύπτει ένας δεκαδικός αριθμός ο οποίος πολλαπλασιασμένος με το 100 δίνει την τιμή της απόδοσης επί τοις εκατό.

Συμπερασματικά, για να θεωρηθεί η εκπαίδευση του μοντέλου επιτυχής η απόδοση πρέπει να αυξάνει προς την τιμή 1.0 με το πέρας των εποχών.

3.5.3 Ανάλυση αρχιτεκτονικής νευρωνικού δικτύου

Το dataset χωρίστηκε με τον εξής τρόπο:

Σετ εκπαίδευσης (Training set): 80% των παραγόμενων εικόνων, δηλαδή 2400 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου.

Σετ επαλήθευσης (Validation set): 20% των παραγόμενων εικόνων, δηλαδή 600 εικόνες χρησιμοποιήθηκαν για την επικύρωση του μοντέλου.

Τα δύο σετ δεν αναμειγνύονται προκειμένου η αξιολόγηση να γίνει πάνω σε άγνωστες για το μοντέλο εικόνες.

Ως προς την αρχιτεκτονική του μοντέλου, χρησιμοποιήθηκαν 5 επίπεδα συνέλιξης, 32 φίλτρα (3x3) στο πρώτο επίπεδο, 64 φίλτρα (3x3) στο δεύτερο και στο τρίτο επίπεδο και

128 φίλτρα στο τέταρτο και στο πέμπτο επίπεδο. Λόγω της πολυπλοκότητας του μοντέλου χρησιμοποιήθηκαν περισσότερα φίλτρα με αυξημένο αριθμό παραμέτρων. Στο επίπεδο της υποδειγματοληψίας χρησιμοποιήθηκε η συνάρτηση MaxPooling2D για τη μείωση των διαστάσεων κάθε εικόνας με σκοπό την αποδοτικότερη εξαγωγή χαρακτηριστικών.

Τέλος, στο πλήρως συνδεδεμένο επίπεδο, η συνάρτηση Flatten μετατρέπει τους δισδιάστατους πίνακες χαρακτηριστικών σε μονοδιάστατο διάνυσμα. Το επίπεδο Dense διαθέτει 128 νευρώνες οι οποίοι είναι πλήρως συνδεδεμένοι με όλους τους προηγούμενους νευρώνες ενισχύοντας την εκμάθηση μη γραμμικών σχέσεων συνθέτοντας έτσι όλα τα χαρακτηριστικά που εξάγονται από τα προηγούμενα επίπεδα συνέλιξης. Για την αποφυγή της υπερεκπαίδευσης του μοντέλου, χρησιμοποιήθηκε η τεχνική Dropout, με απενεργοποίηση του 30% των νευρώνων σε κάθε στάδιο της εκπαίδευσης.

Ως συνάρτηση ενεργοποίησης ορίστηκε η ReLU για την εξαγωγή αφαιρετικών χαρακτηριστικών. Τέλος, ως έξοδος ορίστηκε ένα επίπεδο Dense με έναν μόνο νευρώνα με ενεργοποίηση sigmoid η οποία επιτρέπει τη δυαδική ταξινόμηση.

3.5.4 Αξιολόγηση μετρικών του νευρωνικού δικτύου

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε σε 30 εποχές στις οποίες αξιολογούνταν τόσο η ακρίβεια, όσο και η συνάρτηση απώλειας του δικτύου κάνοντας χρήση της συνάρτησης βελτιστοποίησης Adam, ενώ ο ρυθμός μάθησης προσαρμόζεται κατάλληλα καθ' όλη τη διάρκεια εκπαίδευσης. Ειδικότερα, ο ρυθμός μάθησης μειώνεται κατά 30% αν η συνάρτηση απώλειας του σετ επαλήθευσης δεν μειωθεί μετά το πέρας 2 εποχών.

Αναλυτικότερα για την εξέλιξη των μετρικών κατά την εκπαίδευση, στην $1^{η}$ εποχή η συνάρτηση της απώλειας έχει τιμή $0{,}406$ και $0{,}614$ για το σετ εκπαίδευσης και το σετ επαλήθευσης αντίστοιχα. Στην $30^{η}$ εποχή, η συνάρτηση της απώλειας έχει την τιμή $0{,}078$ για το σετ δεδομένων εκπαίδευσης και $0{,}189$ για τα δεδομένα επαλήθευσης. Επιπροσθέτως, η μετρική της ακρίβειας είναι της τάξης $0{,}861$ για τα δεδομένα εκπαίδευσης και $0{,}865$ για τα δεδομένα επαλήθευσης στην $1^{η}$ εποχή, ενώ στην $30^{η}$ εποχή οι αντίστοιχες τιμές είναι $0{,}972$ για το σετ εκπαίδευσης και $0{,}930$ για τα δεδομένα επαλήθευσης.

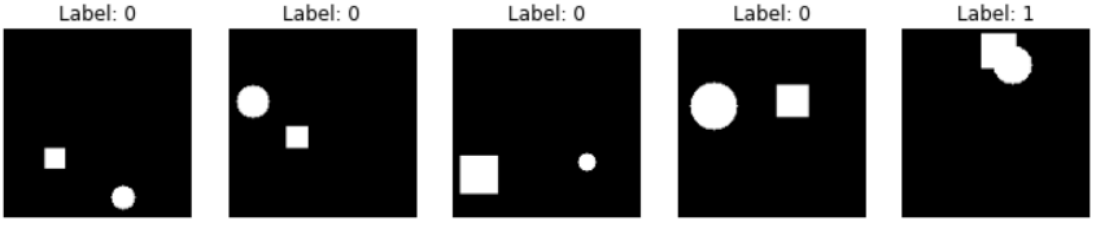
Figure 21: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται ανοδική πορεία και σταθεροποίηση στην 28η εποχή και για τα δύο σετ δεδομένων.

Figure 22: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται συνεχής μείωση και σταθεροποίηση στην 24η εποχή και για τα δύο σετ δεδομένων.

Από το διάγραμμα της απώλειας συναρτήσει των εποχών, η συνάρτηση της απώλειας σταθεροποιείται στην 24η εποχή και για τα δύο σετ δεδομένων. Αντίστοιχα, από το διάγραμμα της ακρίβειας συναρτήσει των εποχών εκπαίδευσης, η μετρική της ακρίβειας σταθεροποιείται και για τα δύο σετ στην 28η εποχή. Η διαφορά στην τάξη μεγέθους οφείλεται στη μεγαλύτερη έκταση του σετ εκπαίδευσης, γεγονός που επιτρέπει στο μοντέλο να επεξεργάζεται τα δεδομένα περισσότερες φορές και ως εκ τούτου να οδηγείται σε μεγαλύτερη προσαρμογή και σε ελαχιστοποίηση της απώλειας, συγκριτικά με τα δεδομένα εκπαίδευσης στα οποία ο αλγόριθμος απλώς αξιολογείται στο τέλος κάθε εποχής.

Ο ρυθμός μάθησης προσαρμόστηκε δύο φορές κατά την εκπαίδευση του μοντέλου ενισχύοντας τη διαδικασία εκμάθησης. Ειδικότερα, από την αρχική τιμή $2 \cdot 10^{-5}$ μειώθηκε στην 20η εποχή σε $\sim 6 \cdot 10^{-6}$ και κατόπιν στην 22η σε $\sim 1{,}8 \cdot 10^{-6}$, επιτυγχάνοντας συνολικά ακρίβεια 92% στην ορθή πρόβλεψη των ετικετών ως προς την αλληλοεπικάλυψη μεταξύ των γεωμετρικών σχημάτων.

Συμπερασματικά, το μοντέλο μαθαίνει ικανοποιητικά και η συνολική απόδοσή του στο ζητούμενο πρόβλημα μπορεί να θεωρηθεί επιτυχής δεδομένης της αυξημένης πολυπλοκότητας των δεδομένων εισόδου λόγω της διαφορετικής φύσης των γεωμετρικών σχημάτων, των διαφορετικών διαστάσεων και της επιτρεπτής επικάλυψης.



Εικόνα 23: Τυχαίες εικόνες από το σετ επαλήθευσης με τις προβλεπόμενες ετικέτες τους

4. Ανάπτυξη CNN για την διάκριση της νόσου Alzheimer μέσω DaTSCAN

4.1 Νόσος Alzheimer

Η νόσος Alzheimer αποτελεί μία προοδευτική νευροεκφυλιστική διαταραχή που χαρακτηρίζεται από σταδιακή εμφάνιση άνοιας η οποία οδηγεί κατά κανόνα σε θάνατο συνήθως 7 με 10 χρόνια μετά τη διάγνωση. Συνδέεται με απώλεια μνήμης, χωρικό αποπροσανατολισμό και σταδιακή επιδείνωση της διανοητικής ικανότητας *([Med 11])*.

Ως άνοια, χαρακτηρίζεται μία κατάσταση στην οποία οι γνωστικές (νοητικές) λειτουργίες του ασθενή εξασθενούν σε σημείο που να επηρεάζεται η ικανότητά του για ανεξάρτητη διαβίωση. Στα συμπτώματα της άνοιας συμπεριλαμβάνονται συχνά συμπεριφορικές και ψυχικές διαταραχές πέραν της νοητικής έκπτωσης. Η νόσος Alzheimer χαρακτηρίζεται από σταδιακή ανάπτυξη παθολογικών διεργασιών σε τμήματα του κεντρικού νευρικού συστήματος, με τελικό αποτέλεσμα την καταστροφή των νευρικών κυττάρων και την ατροφία του εγκεφάλου. Καθώς οι διεργασίες αυτές προσβάλλουν ολοένα και ευρύτερες περιοχές, τα συμπτώματα γίνονται προοδευτικά περισσότερα και εντονότερα. Η διαταραχή της πρόσφατης βιογραφικής μνήμης αποτελεί το συνηθέστερο πρώτο σύμπτωμα.

Η νόσος παρατηρείται σε όλες τις κοινωνίες και σε άτομα οποιασδήποτε φυλετικής καταγωγής. Η ηλικία είναι ο κύριος παράγοντας κινδύνου. Στο ηλικιακό φάσμα 65 – 69 αφορά περίπου το 5% του πληθυσμού. Ακολούθως, το ποσοστό αυτό αυξάνεται εκθετικά με κάθε επόμενη δεκαετία, αγγίζοντας ποσοστά 10 – 15% και 20 – 30% για τις ηλικίες 75 και 85 ετών αντίστοιχα.

Οι νευροεκφυλιστικές διαδικασίες που χαρακτηρίζουν τη νόσο Alzheimer αναπτύσσονται βαθμιαία επί μεγάλο χρονικό διάστημα, ακόμα και πριν από την εμφάνιση των συμπτωμάτων:

- Ασυμπτωματικό - προκλινικό στάδιο:

Διαρκεί πολλά έτη, ίσως και περισσότερα από 10, κατά τα οποία βλάπτονται σταδιακά συγκεκριμένες περιοχές του εγκεφάλου. Παρά το ότι συσσωρεύονται διαφόρων ειδών βλάβες, δεν είναι ικανής βαρύτητας ώστε να προκαλέσουν συμπτώματα, ή ενδεχομένως εξισορροπούνται από τους μηχανισμούς εφεδρείας του εγκεφάλου. Επομένως, αν και τα άτομα δεν νοσούν κλινικά, φέρουν τις βιολογικές αλλοιώσεις που χαρακτηρίζουν τα πρώτα στάδια της νόσου.

- Ήπια γνωστική διαταραχή:

Στο στάδιο αυτό (Mild Cognitive Impairment), οι βλάβες στον εγκέφαλο είναι πιο εκτεταμένες, με αποτέλεσμα την εμφάνιση των πρώτων συμπτωμάτων, συνήθως διαταραχή της πρόσφατης αυτοβιογραφικής μνήμης. Ωστόσο, τα άτομα είναι ακόμα σε θέση να ανταπεξέλθουν στις καθημερινές τους υποχρεώσεις. Επισημαίνεται ότι η ήπια

γνωστική διαταραχή είναι μια περιγραφική διάγνωση με ετερογενή αιτιολογία, δηλαδή δεν οφείλεται πάντα σε υποκείμενη νόσο Alzheimer. Η διάρκεια του σταδίου αυτού ποικίλλει: εκτιμάται ότι 5-15% των ατόμων με ήπια γνωστική διαταραχή μεταπίπτουν σε άνοια κάθε έτος.

- Στάδιο κλινικά έκδηλης νόσου Alzheimer:

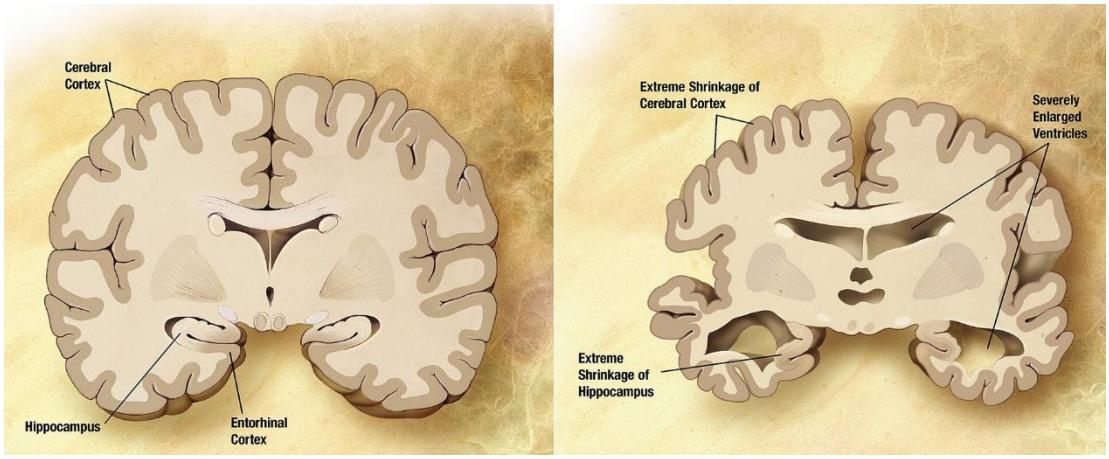
Τα συμπτώματα καθίστανται πλέον αρκετά έντονα, ώστε να επηρεάζεται η ζωή και η καθημερινότητα των ατόμων. Οι νοητικές λειτουργίες σταδιακά επιβαρύνονται, συχνά εμφανίζονται συμπεριφορικά και ψυχολογικά συμπτώματα, ενώ οι ασθενείς χρειάζονται ολοένα και περισσότερη φροντίδα και επιτήρηση. Στα τελικά στάδια της νόσου, οι ασθενείς καθίστανται πλήρως εξαρτημένοι από το περιβάλλον τους, ακόμα και για τις βασικές λειτουργίες.

Τα ανατομικά ευρήματα στον εγκέφαλο ενός ατόμου με νόσο Alzheimer περιλαμβάνουν γενικευμένη ατροφία, ή αλλιώς φθορά του εγκεφάλου *([Kec nd])*. Οι κύριες περιοχές που επηρεάζονται από τη νόσο Alzheimer περιλαμβάνουν τον ιππόκαμπο, ο οποίος επηρεάζεται πρώτα και ελέγχει τη μνήμη και τη μάθηση, τον κροταφικό και το βρεγματικό λοβό που ελέγχουν τη σκέψη, τη γλώσσα και τη λήψη αποφάσεων, τον ενδορρινικό φλοιό και τον υποθάλαμο *([Phy 24])*.

Βιολογικές αλλαγές όπως η παρουσία συγκεκριμένων βιοδεικτών σε δείγματα ασθενών ή η συσσώρευση παθολογικών χαρακτηριστικών μπορούν να βοηθήσουν στη διάγνωση της νόσου στο προκλινικό στάδιο. Ο πιο γνωστός και μελετημένος δείκτης της νόσου Alzheimer είναι η συσσώρευση εξωκυττάριων πλακών που δημιουργούνται από την αμυλοειδή β πρωτεΐνη (Αβ) και ενδοκυττάριων νευροϊνιδιακών σωρών (NFTs) που σχηματίζονται από υπερφωσφορυλιωμένη ταυ πρωτεΐνη στους νευρώνες του εγκεφάλου *([Oro 24])*.

Η νόσος Alzheimer χαρακτηρίζεται ιστολογικά από τη σταδιακή εναπόθεση συσσωματωμάτων πρωτεϊνών με δύο ιδιαίτερες δομές *([CIN 20]):*

- Πλάκες αμυλοειδούς: αποτελούνται από εναποθέσεις ενός πεπτιδίου που ονομάζεται Αβ-αμυλοειδές. Οι πλάκες αυτές βρίσκονται ανάμεσα στα νευρικά κύτταρα και σχετίζονται με τις βιοχημικές διαδικασίες επεξεργασίας της πρόδρομης πρωτεΐνης του αμυλοειδούς (amyloid precursor protein).
  Οι εξωκυτταρικές πλάκες αμυλοειδούς είναι οι χαρακτηριστικές εγκεφαλικές αλλοιώσεις της σποραδικής νόσου του Alzheimer.
- Νευροϊνιδιακοί σωροί (neurofibrillary tangles): αποτελούνται από την πρωτεΐνη tau (ή αλλιώς πρωτεΐνη “τ”) η οποία είναι εξαιρετικά σημαντική για τη λειτουργία των νευρώνων και βρίσκονται εντός των νευρικών κυττάρων. Ειδικότερα, σχηματίζονται στα νευρικά κύτταρα που υφίστανται εκφύλιση κατά τη νόσο, στην οποία η χωρική κατανομή τους συσχετίζεται με το βαθμό της άνοιας. Οι συγκεκριμένες αλλοιώσεις ξεκινούν από συγκεκριμένα σημεία του εγκεφάλου και επεκτείνονται σε παρακείμενες περιοχές καθώς η νόσος εξελίσσεται. Παράλληλα, παρατηρούνται μεταβολές σε αρκετά άλλα κυτταρικά συστήματα, μείωση του αριθμού των νευρώνων και ατροφία του εγκεφάλου.



Εικόνα 24: Στα αριστερά διακρίνεται υγιής εγκέφαλος και στα αριστερά εγκέφαλος που πάσχει από νόσο Alzheimer με εμφανή τη συρρίκνωση του ιππόκαμπου.

Μέχρι στιγμής, δεν υπάρχει θεραπεία η οποία να καθυστερεί την εξέλιξη των διεργασιών που χαρακτηρίζουν τις νευροεκφυλιστικές, μη – αναστρέψιμες μορφές άνοιας. Ως εκ τούτου, οι παρεμβάσεις δρουν επικουρικά βελτιώνοντας τα συμπτώματα.

Για τον λόγο αυτό, κρίνεται απαραίτητη η έγκυρη διάγνωση της νόσου με στόχους της εξέτασης τους παρακάτω:

i. Διερεύνηση του είδους και της βαρύτητας των αποτελεσμάτων
ii. Έλεγχος και αποκλεισμός πιθανών καταστάσεων που μιμούνται τα συμπτώματα της νόσου Alzheimer
iii. Σχηματισμός ενός θεραπευτικού πλάνου μετά τη διάγνωση

4.2 Απεικονιστικές τεχνικές στη Διάγνωση νόσου Alzheimer

Η τρέχουσα διάγνωση της νόσου γίνεται με κλινικές, νευροψυχολογικές και νευροαπεικονιστικές αξιολογήσεις. Νευροαπεικονιστικές τεχνικές όπως η απεικόνιση μαγνητικού συντονισμού (MRI), η τομογραφία εκπομπής ποζιτρονίων (PET) και η τομογραφία μονοφωτονικής εκπομπής (SPECT) είναι ζωτικής σημασίας για την απεικόνιση των δομών του εγκεφάλου για τη διάγνωση και τη θεραπεία της νόσου Alzheimer *([Cas 24])*.

Οι τομογραφίες PET και SPECT αποτελούν εφαρμογές της πυρηνικής ιατρικής που χρησιμοποιούν ραδιοϊχνηθέτες για την αξιολόγηση βιολογικών διεργασιών σε έμβιους οργανισμούς και αξιοποιούνται τόσο στη διάγνωση όσο και στην εκτίμηση της πρόγνωσης της νόσου Alzheimer. Καθώς επιτρέπουν την ποσοτική απεικόνιση των λειτουργικών και μοριακών διεργασιών στον ζωντανό ανθρώπινο εγκέφαλο ακόμα και σε πρώιμο στάδιο της εξέλιξης της νόσου, έχουν καταστεί αναπόσπαστα μέρη της διαγνωστικής εξέτασης ασθενών με νευροεκφυλιστικές νόσους *([Mey 14])*.

Η τομογραφία PET είναι μία ευαίσθητη τεχνική μοριακής απεικόνισης που επιτρέπει την in vivo ποσοτικοποίηση σε απόλυτες τιμές της συγκέντρωσης ενός ραδιοϊχνηθέτη καθιστώντας εφικτή την αξιολόγηση της μοριακής διεργασίας στους τόπους δράσης *([Vil 05])*.

Η τομογραφία SPECT ανιχνεύει πρώιμες λειτουργικές ανωμαλίες του κεντρικού νευρικού συστήματος, συχνά πριν εμφανιστούν μορφολογικές αλλαγές ή πριν εντοπιστούν με άλλες απεικονιστικές μεθόδους. Μπορεί να διακρίνει μεταξύ της άνοιας τύπου Alzheimer Korsakoff, της αγγειακής άνοιας, της άνοιας του μετωπιαίου λοβού, της ψύχωσης και της εγκεφαλοπάθειας από τον ιό της ανθρώπινης ανοσοανεπάρκειας (HIV) *([Blo 96])*. Μέσω της SPECT δίδεται η δυνατότητα ανίχνευσης λειτουργικών ανωμαλιών όπως για παράδειγμα η μειωμένη αιμάτωση σε συγκεκριμένες περιοχές του εγκεφάλου επιτρέποντας την έγκαιρη διάγνωση και την αποφυγή επεισοδίου. Αντιθέτως, η αξονική τομογραφία απεικονίζει πιο μόνιμες βλάβες, συνήθως μορφολογικές, οι οποίες δεν δύναται να αποφευχθούν καθώς είναι ήδη εγκατεστημένες.

Τα ευρήματα της τομογραφίας SPECT μπορούν να οδηγήσουν σε αλλαγή της κλινικής διαχείρισης του ασθενούς.

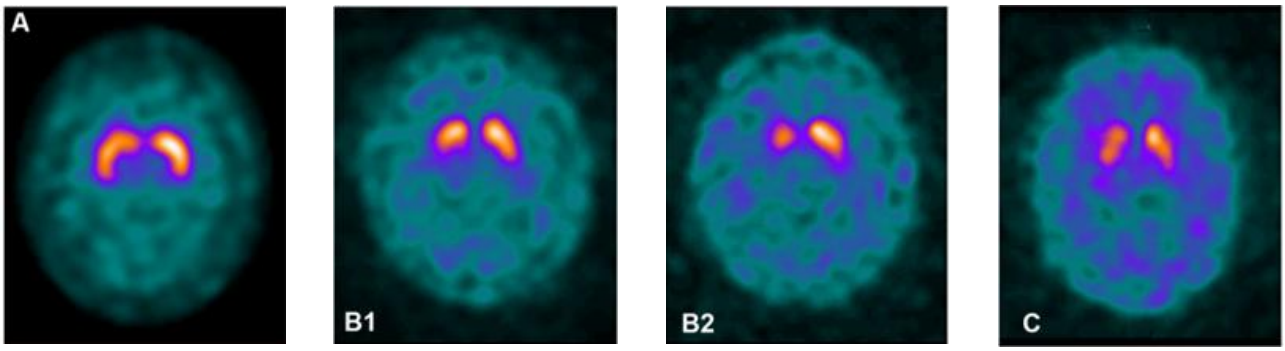
Τα σχετικά πλεονεκτήματα των τομογραφιών PET και SPECT παρατίθενται στον παρακάτω πίνακα *([Vil 05]):*

PET	SPECT
Υψηλότερη ανάλυση	Χαμηλότερο κόστος
Υψηλότερη ευαισθησία	Ευρεία διαθεσιμότητα
Ποσοτική	Ισότοπα με μεγαλύτερους χρόνους ημιζωής
Μεταβολικοί ιχνηθέτες	

4.3 DaTSCAN

Μία ευρέως γνωστή εξέταση που βασίζεται στην απεικονιστική SPECT και είναι χρήσιμη στη διάγνωση της νόσου Alzheimer αποτελεί η εξέταση DaTSCAN (Dopamine Transporter Scan) ή διαφορετικά, το σπινθηρογράφημα εγκεφάλου. Η εξέταση DaTSCAN χρησιμοποιείται για την ανίχνευση της απώλειας των λειτουργικών δοπαμινεργικών νευρικών απολήξεων στο ραβδωτό σώμα ασθενών, η οποία συμβάλλει στην παθοφυσιολογία της νόσου μέσω της ενίσχυσης της γνωστικής και κινητικής δυσλειτουργίας αλλά και στην επιδείνωση των συμπεριφορικών συμπτωμάτων *([EMA])*.

Μία φυσιολογική εικόνα DaTSCAN περιλαμβάνει ομοιόμορφη απορρόφηση του ραδιοϊχνηθέτη στα δύο εγκεφαλικά ημισφαίρια ενώ μια μη φυσιολογική εικόνα DaTSCAN περιλαμβάνει ανομοιογενή απορρόφηση του ραδιοϊχνηθέτη ανάμεσα στα δύο εγκεφαλικά ημισφαίρια.



Εικόνα 25: Η εικόνα Α απεικονίζει φυσιολογική εξέταση DaTSCAN ενώ οι εικόνες Β1, Β2, C απεικονίζουν μη φυσιολογικές εικόνες DaTSCAN καθώς παρατηρείται ανομοιογενής απορρόφηση του ραδιοϊχνηθέτη. ([O'Sh 24])

Η απώλεια των λειτουργικών δοπαμινεργικών νευρικών απολήξεων στο ραβδωτό σώμα ασθενών είναι χαρακτηριστικό της άνοιας με σώματα Lewy (Lewy Body Dementia - LBD) και της νόσου Parkinson αλλά όχι της τυπικής νόσου Alzheimer. Για αυτόν το λόγο χρησιμοποιείται για να διαχωρίσει τους ασθενείς με άνοια LBD από τους ασθενείς με νόσο Alzheimer, στους πρώτους υπάρχει απώλεια δοπαμινεργικών απολήξεων που διακρίνεται μέσω της εξέτασης, ενώ στους δεύτερους η DaTSCAN είναι φυσιολογική *([O' Sh 24])*. Η εξέταση αυτή χρησιμοποιείται επίσης για τη διάκριση ασθενών με νόσο Parkinson και με νόσο Alzheimer καθώς στην πρώτη περίπτωση η απορρόφηση του ραδιοφαρμάκου είναι ανομοιογενής ενώ στην δεύτερη η απορρόφηση είναι ομοιόμορφη.

- Μηχανισμός δράσης διαλύματος DaTSCAN:

Για τους σκοπούς της εξέτασης, χορηγείται ενέσιμο διάλυμα DaTSCAN 74MBq/ml. Περιέχει τη δραστική ουσία ιοφλουπάνιο $^{123}\mathrm{I}$ το οποίο χρησιμοποιείται για να βοηθήσει στον εντοπισμό, δηλαδή τη διάγνωση, παθήσεων του εγκεφάλου. Ανήκει στην κατηγορία των ραδιοφαρμάκων. Το ιοφλουπάνιο έχει μεγάλη συγγένεια με τον προσυναπτικό μεταφορέα ντοπαμίνης και έτσι μπορεί να χρησιμοποιηθεί ως αναπληρωματικός δείκτης για την εξέταση της ακεραιότητας των δοπαμινεργικών μελανοραβδωτών νευρώνων.

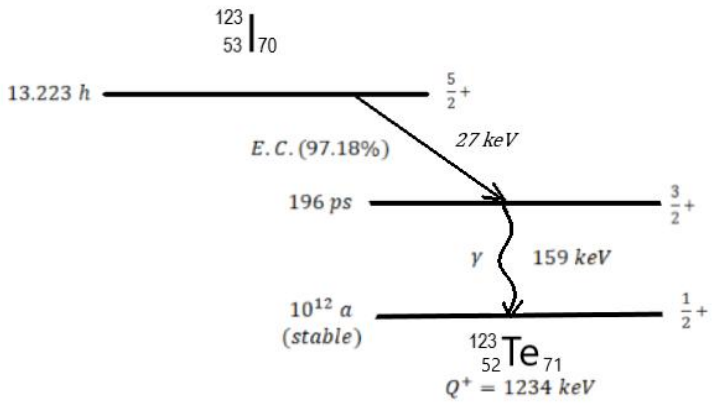
Κάθε ml διαλύματος περιέχει ιοφλουπάνιο $^{123}\mathrm{I}$ $74\ \mathrm{MBq/ml}$ κατά το χρόνο αναφοράς (0,07 έως 0,13 μg/ml ιοφλουπανίου) με λήξη 20 ώρες μετά το χρόνο αναφοράς. Η ραδιενέργεια αναφοράς είναι ίση με $370\ MBq/5\ ml$ στις 23: 00 η ώρα της αναγραφόμενης ημερομηνίας $HH/MM/EEEE$.

- Δοσολογία:

Η κλινική αποτελεσματικότητα έχει αποδειχθεί για την περιοχή 111 έως 185 MBq. Δεν πρέπει να υπερβαίνονται τα 185 MBq αλλά ούτε να χρησιμοποιείται σε ραδιενέργεια χαμηλότερη των 110 MBq *([GE 20])*.

- Δοσιμετρία και πρόσληψη από τα όργανα:

Το $^{123}\mathrm{I}$ έχει χρόνο υποδιπλασιασμού $T_{\frac{1}{2}} = 13{,}223\ h$. Αποδιεγείρεται εκπέμποντας γάμμα ακτινοβολία ενέργειας 159 keV και ακτίνες Χ ενέργειας 27 keV *([NNDC])*.



Εικόνα 26: Διάγραμμα διάσπασης του ^{123}I, ([Chi 04])

Το ιοφλουπάνιο $^{123}\mathrm{I}$ απομακρύνεται ταχέως από το αίμα μετά από την ενδοφλέβια ένεση. Μόνο 5% της χορηγούμενης ραδιενέργειας παραμένει σε όλο το αίμα στα 5 λεπτά μετά την ένεση.

Η πρόσληψη από τον εγκέφαλο είναι ταχεία και φτάνει περίπου το 7% της ενεθείσας ραδιενέργειας στα 10 λεπτά μετά την ενδοφλέβια ένεση. Ελαττώνεται στο 3% μετά από 5 ώρες. Περίπου το 30% του συνόλου της ραδιενέργειας στον εγκέφαλο αποδίδεται σε πρόσληψη από το ραβδωτό σώμα.

Στις 48 ώρες μετά την ένεση, περίπου το 74% την ενεθείσας ραδιενέργειας απεκκρίνεται με φυσικές διεργασίες απόκρισης του οργανισμού *([GE 20])*.

Οι υπολογιζόμενες απορροφούμενες δόσεις ακτινοβολίας σε ένα μέσο ενήλικα ασθενή (70 kg) από ενδοφλέβια ένεση ιοφλουπανίου $^{123}\mathrm{I}$ αναγράφονται παρακάτω:

Όργανο – Στόχος	**Απορροφούμενη δόση ακτινοβολίας μGy / MBq**
Επινεφρίδια	17.0
Οστική επιφάνεια	15.0
Εγκέφαλος	16.0
Μαστοί	7.3
Τοίχωμα χοληδόχου κύστεως	44.0
Γαστρεντερικός σωλήνας	
Τοίχωμα στομάχου	12.0
Τοίχωμα λεπτού έντερου	26.0
Τοίχωμα παχεως έντερου	59.0
(Τοίχωμα ανώτερου παχέος εντέρου	57.0)
(Τοίχωμα κατώτερου παχέος εντέρου	62.0)
Καρδιακό τοίχωμα	32.0
Νεφροί	13.0
Ήπαρ	85.0
Πνεύμονες	42.0
Μυς	8.9
Οισοφάγος	9.4
Ωοθήκες	18.0
Πάγκρεας	17.0
Ερυθροποιητικός μυελός	9.3
Σιελογόνοι αδένες	41.0
Δέρμα	5.2
Σπλήν	26.0
Όρχεις	6.3
Θύμος	9.4
Θυρεοειδής	6.7
Τοίχωμα ουροδόχου κύστεως	35.0
Μήτρα	14.0
Υπόλοιπα όργανα	10.0
Ενεργός δόση (μSv/MBq)	**25.0**

Table 5: Απορροφούμενες δόσεις ακτινοβολίας σε ένα μέσο ενήλικα 70 kg από ενδοφλέβια ένεση ^{123}I.

([Mat 15])

Η ενεργός δόση αντιστοιχεί αριθμητικά στην ολοσωματική ισοδύναμη δόση που θα έπρεπε να δεχτεί το εκτιθέμενο άτομο ώστε να διατρέξει τον ίδιο κίνδυνο από την τοπική ακτινοβόληση ενός και μόνο ιστού με ισοδύναμη δόση H_T. Σχετίζεται με τον ενεχόμενο συνολικό κίνδυνο για την υγεία, ανεξάρτητα από το είδος της προσβάλλουσας ακτινοβολίας, τις συνθήκες ακτινοβόλησης και την ακτινοβολούμενη περιοχή του ανθρώπινου σώματος.

Για την ενεργό δόση ισχύει: $$\boldsymbol{E = \sum_t H \cdot w_t = \sum_t D \cdot w_R \cdot w_t}$$

Όπου $\boldsymbol{w_R}$: σταθερά που αφορά το είδος της ακτινοβολίας με $w_R = 1$ για φωτόνια (χ και γ) όλων των ενεργειών

$\boldsymbol{w_t}$: σταθερά που αφορά το είδος του ιστού που ακτινοβολείται

Η ενεργός δόση (E) που προκύπτει από τη χορήγηση 185 MBq ενέσιμου διαλύματος DaTSCAN είναι 4,63 mSv (ανά άτομο 70 kg). Τα πιο πάω στοιχεία ισχύουν σε άτομα με φυσιολογική φαρμακοκινητική συμπεριφορά καθώς σε περιπτώσεις μειωμένης νεφρικής ή ηπατικής λειτουργίας, η ενεργός δόση και η δόση ακτινοβολίας που αποδίδονται στα όργανα θα μπορούσαν να αυξηθούν.

- Τρόπος χορήγησης:

Το ενέσιμο διάλυμα πρέπει να χρησιμοποιείται χωρίς αραίωση. Η SPECT απεικόνιση πρέπει να λάβει χώρα μεταξύ τριών και έξι ωρών μετά την ένεση. Οι εικόνες πρέπει να λαμβάνονται χρησιμοποιώντας γάμμα κάμερα με κατευθυντήρα υψηλής ανάλυσης και ρυθμισμένη με χρήση της φωτοκορυφής των 159 keV και ενεργειακό παράθυρο ± 10%. Η γωνιακή δειγματοληψία πρέπει κατά προτίμηση να μην είναι λιγότερη από 120 λήψεις σε περιστροφή 360 μοιρών. Για υψηλής ανάλυσης κατευθυντήρες παράλληλου τύπου, η ακτίνα περιστροφής πρέπει να είναι σταθερή και όσο το δυνατόν μικρότερη (τυπικά 11 – 15 cm). Οι συντελεστές μεγέθους μήτρας απεικόνισης και μεγέθυνσης πρέπει να επιλέγονται ώστε να δώσουν μέγεθος στοιχείου εικόνας (pixel size) 3,5 - 4,5 mm για τα συστήματα αυτά που είναι σήμερα σε χρήση. Ένα ελάχιστο 500.000 κρούσεων πρέπει να ανιχνεύεται για άριστες εικόνες *([GE 20])*.

- Απεικόνιση βλάβης:

Οι φυσιολογικές εικόνες χαρακτηρίζονται από δύο συμμετρικές περιοχές σχήματος ημισελήνου ίσης έντασης. Οι μη φυσιολογικές εικόνες είναι είτε ασύμμετρες είτε συμμετρικές με άνιση ένταση και/ή με απώλεια της ημισελήνου.

4.4 Κατασκευή Ομοιωμάτων Εικόνων DaTSCAN

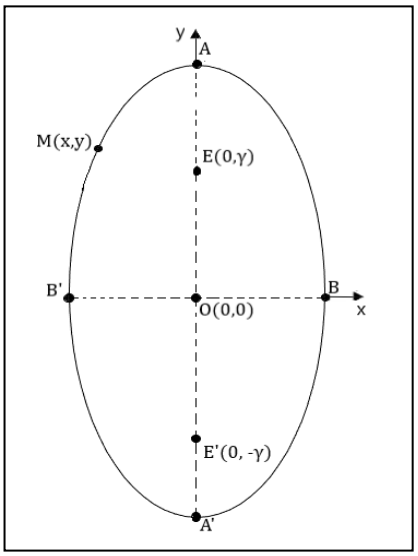
Ο σκοπός του πειράματος οδήγησε στην αναγκαιότητα χρήσης ενός μεγάλου πλήθους τομογραφικών εικόνων εγκεφάλων εξέτασης DaTSCAN, οι οποίες θα χρησιμοποιηθούν στην εκπαίδευση και μετέπειτα στην αξιολόγηση του μοντέλου πρόβλεψης της ποσοτικοποιημένης διαφοράς της απορρόφησης του ^{123}I από τα δύο εγκεφαλικά ημισφαίρια. Λόγω ιατρικού απορρήτου και δυσκολίας ανεύρεσης μεγάλου αριθμού εξετάσεων DaTSCAN, κρίθηκε απαραίτητη η δημιουργία του ζητούμενου δείγματος εικόνων , δηλαδή η δημιουργία ομοιωμάτων τομογραφικών εικόνων εγκεφάλου με ανομοιογένεια της απορροφούμενης δόσης του ραδιοφαρμάκου μεταξύ των εντάσεων των δύο λοβών.

4.5 Κατασκευή CNN

Για τον λόγο αυτό γίνεται χρήση του προγράμματος Simulix3x, ενός προγράμματος κατασκευής ομοιωμάτων τομογραφιών εγκεφάλου, το οποίο λαμβάνοντας ορισμένες

παραμέτρους που θα αναλυθούν στη συνέχεια ως είσοδο, παράγει τις ζητούμενες εικόνες σε μορφή πίνακα, οι οποίες στη συνέχεια μπορούν να επεξεργαστούν από τον αλγόριθμο.

Αναλυτικότερα, στα ομοιώματα των τομογραφικών εικόνων που κατασκευάζονται από το πρόγραμμα, τόσο η κεφαλή όσο και το επίκεντρο απορρόφησης του ραδιοφαρμάκου ^{123}I των δύο εγκεφαλικών λοβών , το λεγόμενο hotspot, θεωρούνται ότι είναι ελλείψεις με εξίσωση:



$\frac{x^2}{\beta^2} + \frac{y^2}{\alpha^2} = 1$,

όπου $\beta = \sqrt{\alpha^2 - \gamma^2}$,

$E(0, \gamma), E'(0, -\gamma)$: Οι εστίες της έλλειψης

$(E'E) = 2\gamma$,

$(ME') + (ME) = 2\alpha$,

$M(x, y)$: Τυχαίο σημείο της έλλειψης.

Οι τιμές που ορίζονται από τον χειριστή του προγράμματος είναι οι παρακάτω:

Αναλογία αξόνων ελλειψοειδούς κεφαλής ($\boldsymbol{head\ ratio}$) : $\frac{x}{y} = 1.30$

Αναλογία αξόνων ελλειψοειδούς hotspot: $\frac{x}{y} = 2.00$

Δραστηριότητα υποβάθρου για το ομοίωμα κεφαλής: 0.10

Αντίθεση του hotspot ($spot\ contrast$): 0.90

Για το μέγεθος της κεφαλής χρησιμοποιείται η συνάρτηση:

$$\boldsymbol{head\ size} = \frac{1}{\boldsymbol{head\ ratio}} \cdot \frac{NN}{2} \cdot \left(0.7 + 0.2 \cdot \boldsymbol{ran1}(\boldsymbol{Iseed})\right)$$

με NN τις διαστάσεις του πλέγματος.

Ως $ran1(Iseed)$ έχει οριστεί ένας γεννήτορας τυχαίων αριθμών στο διάστημα $[0,1]$, με αρχική τιμή της ακολουθίας το όρισμα που έχει δοθεί ως είσοδος. Η ενσωμάτωση του γεννήτορα τυχαίων αριθμών προσθέτει στοχαστικές, τυχαίες παραλλαγές στα παραγόμενα ομοιώματα.

Για τη θέση της κεφαλής, οι συντεταγμένες της (X_H, Y_H) δεν είναι πάντοτε κεντραρισμένες ως προς το κέντρο του πλέγματος εξαιτίας της στοχαστικής μετατόπισης που εισάγεται λόγω της συνάρτησης $ran1(Iseed)$. Όσον αφορά τις διαστάσεις της, $SX(1), SY(1)$ αυτές εξαρτώνται από το μέγεθος της κεφαλής και της αναλογίας των αξόνων του ελλειψοειδούς της.

Για την ένταση του υποβάθρου της κεφαλής χρησιμοποιείται η συνάρτηση:

$$\boldsymbol{I_H} = 0.10 + (\boldsymbol{head\ background}) \cdot \boldsymbol{ran1}(\boldsymbol{seed})$$

Για την δημιουργία των δύο hotspot χρησιμοποιούνται οι συναρτήσεις:

$$\begin{cases} \boldsymbol{Spot\ size = head\ size} \cdot \left(\boldsymbol{0.08 + 0.08 \cdot ran1(Iseed)}\right) \\ \boldsymbol{XDisp = head\ size} \cdot \left(\boldsymbol{0.25 + 0.15 \cdot ran1(Iseed)}\right) \\ \boldsymbol{YDisp = head\ size} \cdot \boldsymbol{0.30 \cdot ran1(Iseed)} \end{cases}$$

Όπου $Spot\ size$ το μέγεθος του επίκεντρου της απορρόφησης - hotspot, $XDisp$ και $YDisp$ οι μετατοπίσεις των hotspot από το κέντρο του κεφαλιού.

Όσον αφορά τις συντεταγμένες των hotspot: $\begin{cases} \boldsymbol{X_L = X_H - XDisp} \\ \boldsymbol{Y_L = Y_H + YDisp} \end{cases}$ και

$\begin{cases} \boldsymbol{X_R = X_H + XDisp} \\ \boldsymbol{Y_R = Y_H + YDisp} \end{cases}$

Οι διαστάσεις $SX(2,3), SY(2,3)$ είναι όμοιες και για τις δύο ελλείψεις και βασίζονται στο $spot\ size$, ενώ ο λόγος ύψους πλάτους προκύπτει από τη μεταβλητή $spot\ ratio$.

Αναφορικά με τις εντάσεις του κάθε hotspot υπολογίζονται μέσω του λογισμικού ως:

$$\boldsymbol{I_i = I_H + (Spot\ contrast) \cdot ran1(Iseed)},$$

Όπου $\boldsymbol{I_i}$**:** Η ένταση του κάθε hotspot - αριστερού $(i = L)$ και δεξιού $(i = R)$

Παρακάτω παρατίθενται τα όρια κάθε μεταβλητής, λαμβάνοντας υπόψιν τη συνεισφορά του $ran1(Iseed)$:

Μεταβλητή	Διάστημα
$head\ size$	$[34.46, 44{,}31]$
$spot\ size$	$[2.76, 7.09]$
$XDisp$	$[8.62, 17.72]$
$YDisp$	$[0, 13.29]$
Y_L & Y_R	$[61, 80.29]$
X_L	$[43.28, 58.38]$
X_R	$[69.62, 84.72]$

Τα παραπάνω όρια διασφαλίζουν τις τυχαίες παραλλαγές στην παραγωγή των ομοιωμάτων των εγκεφαλικών τομογραφικών εικόνων κατά την εξέταση DaTSCAN.

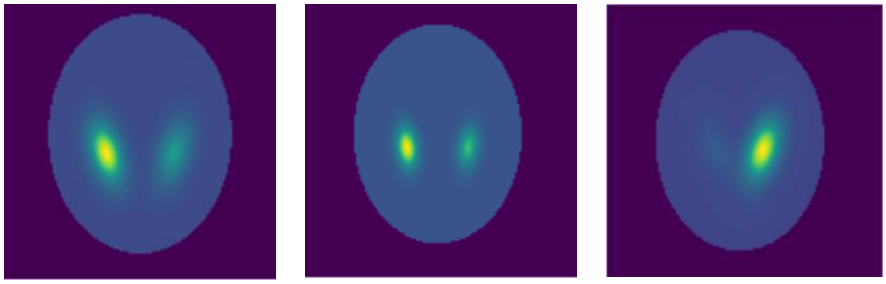
Στη συνέχεια, τοποθετούνται εντός της εικόνας διαστάσεων 128x128, η έλλειψη της κεφαλής όπως και οι ελλείψεις των δύο hotspot.

Οι παράμετροι που υπολογίζονται για τον σχηματισμό των εικόνων είναι οι παρακάτω:

Παράμετροι θέσης	
Συντεταγμένες της έλλειψης της κεφαλής	(X_H, Y_H)
Συντεταγμένες του αριστερού hotspot	(X_L, Y_L)
Συντεταγμένες του δεξιού hotspot	(X_R, Y_R)

Παράμετροι ημιαξόνων	
Ημιάξονες α και β της έλλειψης της κεφαλής	(A_H, B_H)
Ημιάξονες α και β και των δύο hotspot	(A_S, B_S)

Εντάσεις	
Ένταση υποβάθρου κεφαλής ($head\ background$)	$[0.1, 0.2]$
Ένταση αριστερού hotspot (I_L)	$[0.1, 1]$
Ένταση δεξιού hotspot (I_R)	$[0.1, 1]$



Εικόνα 27: Ενδεικτικές εικόνες που παράγονται. Οι εικόνες εμφανίζονται στραμμένες κατά 180° ως προς την κάθετη διεύθυνση.

4.5.1 Ποσοτικοποίηση ανομοιογενούς απορρόφησης του ^{123}I από τα εγκεφαλικά ημισφαίρια

Καθώς η νευροεκφυλιστική διαταραχή Alzheimer παρατηρείται όταν υπάρχει ανομοιόμορφη απορρόφηση του ραδιοφαρμάκου στα δύο ημισφαίρια, είναι κρίσιμο να δημιουργηθεί ένα δίκτυο το οποίο θα μπορεί να ανιχνεύει με ακρίβεια την ανομοιογένεια και να αναγνωρίζει σε ποιο ημισφαίριο η απορρόφηση είναι μεγαλύτερη.

Μέσω του λογισμικού Simulix3x έχουν ληφθεί οι τιμές όλων των παραμέτρων. Από αυτές, μείζονας σημασίας αποτελούν οι τιμές των εντάσεων.

Δεδομένου πως το υπόβαθρο διατηρεί μία μη μηδενική ένταση, στην ένταση κάθε λοβού προστίθεται η ένταση του υποβάθρου. Προκύπτει:

$$\begin{cases} \boldsymbol{R = I_H + I_R} \\ \boldsymbol{L = I_H + I_L} \end{cases}$$

Στη συνέχεια, οι τιμές R , L κανονικοποιούνται στο διάστημα $[-1, +1\ ]$. Δηλαδή, γίνεται έλεγχος μεταξύ των δύο τιμών R, L για την εύρεση του μεγίστου και έπειτα η δεύτερη τιμή διαιρείται με την μέγιστη.

Για παράδειγμα, έστω $\boldsymbol{R > L} \Rightarrow \begin{cases} \boldsymbol{R' = \frac{R}{R} = 1} \\ \boldsymbol{L' = \frac{L}{R} < 1} \end{cases}$

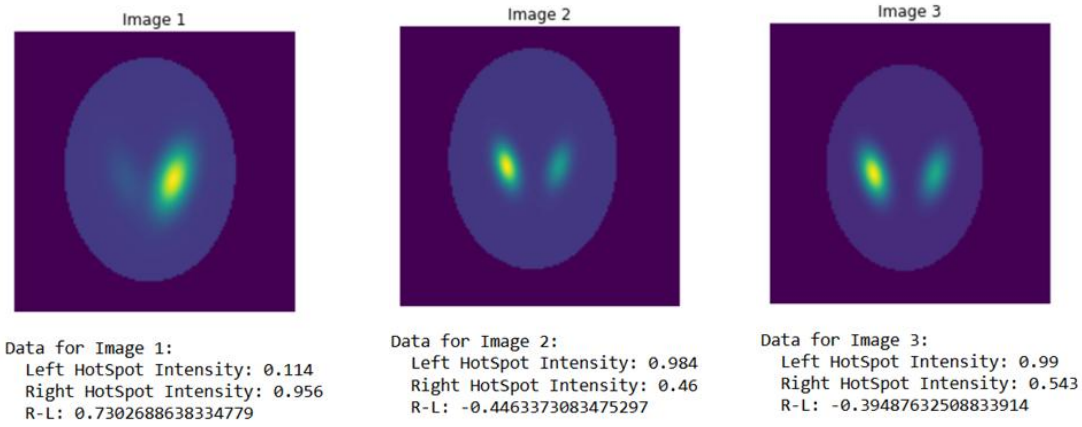
Τέλος, υπολογίζεται η διαφορά των κανονικοποιημένων εντάσεων των δύο λοβών, R', L'.

$$\begin{cases} \boldsymbol{R' - L' > 0} \Rightarrow \textit{Μεγαλύτερη απορροφούμενη ένταση στο δεξί ημισφαίριο} \\ \boldsymbol{R' - L' < 0} \Rightarrow \textit{Μεγαλύτερη απορροφούμενη ένταση στο αριστερό ημισφαίριο} \end{cases}$$

Με τον τρόπο αυτό μπορεί να ποσοτικοποιηθεί η ανομοιογένεια μεταξύ των εντάσεων των δύο ημισελήνων του εγκεφάλου και να εντοπιστεί ο λοβός με τη μεγαλύτερη απορρόφηση του ραδιοφαρμάκου.

4.5.2 Μεθοδολογία Κατασκευής του CNN

Αρχικά, δίνονται ως δεδομένα εισόδου στο νευρωνικό δίκτυο οι εικόνες των ομοιωμάτων σε μορφή πινάκων και οι παράμετροι που αφορούν την κάθε εικόνα. Από τις τιμές των παραμέτρων απομονώνονται εκείνες των εντάσεων, δηλαδή η ένταση του υποβάθρου, η ένταση του αριστερού hotspot και η ένταση του δεξιού hotspot, οι οποίες αποθηκεύονται ως νέος πίνακας δεδομένων και στη συνέχεια γίνονται οι υπολογισμοί που περιεγράφηκαν νωρίτερα. Η νέα ετικέτα $R' - L'$ κάθε εικόνας αποθηκεύεται σε μία κενή λίστα $labels = [\ ]$ σε πλήρη αντιστοιχία για την αντίστοιχη εικόνα.



Εικόνα 2810: Ενδεικτικές εικόνες του σετ εκπαίδευσης με τις αντίστοιχες ετικέτες. Γίνεται επαλήθευση σωστής αποθήκευσης των ομοιωμάτων με τις ετικέτες που τους αντιστοιχούν.

Το σετ δεδομένων, συνολικού όγκου 3000 εικόνων, χωρίζεται σε training data (80% του αρχικού δείγματος) και σε test data (20% του αρχικού δείγματος). Η εκπαίδευση του μοντέλου ολοκληρώνεται σε 30 κύκλους εκπαίδευσης (epochs) κατά την ολοκλήρωση των οποίων παρακολουθείται η εξέλιξη των μετρικών.

Οι μετρικές στο συγκεκριμένο πρόβλημα επιλέχθηκαν να είναι το μέσο τετραγωνικό σφάλμα MSE (Mean Squared Error) και η μέση τετραγωνική απόκλιση RMSE (Root Mean Squared Error) καθώς πρόκειται για πρόβλημα παλινδρόμησης στο οποίο ζητείται από το νευρωνικό δίκτυο η πρόβλεψη μίας συνεχούς τιμής, δηλαδή της διαφοράς $\boldsymbol{R' - L'}$.

Το μέσο τετραγωνικό σφάλμα MSE υπολογίζεται ως:

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{y_i} - \boldsymbol{\hat{y}})^{\mathbf{2}}$$

Αντίστοιχα, η μέση τετραγωνική απόκλιση RMSE υπολογίζεται ως:

$$\boldsymbol{RMSE} = \sqrt{\boldsymbol{MSE}} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{N}} (\boldsymbol{y_i} - \boldsymbol{\hat{y}})^{\mathbf{2}}}$$

Όπου, $\boldsymbol{y_i}$: η προβλεπόμενη διαφορά από το δίκτυο ,

$\boldsymbol{\hat{y}}$: η πραγματική ετικέτα κάθε εικόνας

$\boldsymbol{N}$: το πλήθος των εικόνων

Η διαφορά των δύο μετρικών έγκειται στις μονάδες μέτρησης, η τιμή του MSE είναι σε διαφορετική κλίμακα από τα δεδομένα καθώς τα σφάλματα υψώνονται στη δεύτερη δύναμη. Λόγω της τετραγωνικής ρίζας, το RMSE υπολογίζει το σφάλμα στις ίδιες μονάδες με τα δεδομένα τα οποία είναι ήδη κανονικοποιημένα στο διάστημα $[-1, +1]$, παρέχοντας μια διαισθητική εικόνα του μέσου σφάλματος.

Επειδή το ζητούμενο σε ένα πρόβλημα παλινδρόμησης είναι η δημιουργία ενός μοντέλου που να μπορεί να προβλέψει την τιμή μίας συνεχούς μεταβλητής, η εκπαίδευση του δικτύου θεωρείται επιτυχής κατά την ελαχιστοποίησης της διαφοράς μεταξύ των πραγματικών και των προβλεπόμενων τιμών, δηλαδή κατά την ελαχιστοποίηση των μετρικών.

Συνεπώς, η μείωση και η σταθεροποίηση των μετρικών σε ελάχιστες τιμές, υποδεικνύει πως το νευρωνικό δίκτυο έχει εκπαιδευτεί σωστά στην πρόβλεψη της ποσοτικοποιημένης ανομοιογένειας των εντάσεων των δύο λοβών του εγκεφάλου κατά την εξέταση DaTSCAN και είναι σε θέση να γενικεύει σε νέες τομογραφικές εικόνες.

4.5.3 Ανάλυση αρχιτεκτονικής νευρωνικού δικτύου

Το dataset χωρίστηκε με τον εξής τρόπο:

Σετ εκπαίδευσης (Training set): 80% των παραγόμενων εικόνων, δηλαδή 2400 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου

Σετ επαλήθευσης (Validation set): 20% των παραγόμενων εικόνων, δηλαδή 600 εικόνες χρησιμοποιήθηκαν για την επικύρωση του μοντέλου.

Τα δύο σετ δεν αναμειγνύονται προκειμένου η αξιολόγηση να γίνει πάνω σε άγνωστες για το μοντέλο εικόνες.

Η αρχιτεκτονική του μοντέλου περιλαμβάνει 5 συνελικτικά επίπεδα (Convolutional layers), με το 1^{ο} επίπεδο να χρησιμοποιεί 32 φίλτρα (3x3), τα επόμενα 2 επίπεδα 64 φίλτρα (3x3), και τα τελευταία 2 επίπεδα 128 φίλτρα (3x3). Ο στόχος αυτής της σταδιακής αύξησης των φίλτρων είναι η εξαγωγή πιο περίπλοκων και αφαιρετικών χαρακτηριστικών από τις εικόνες για καλύτερη αναπαράσταση των δεδομένων.

Στο επίπεδο της υποδειγματοληψίας χρησιμοποιήθηκε ο αλγόριθμος MaxPooling2D μετά από κάθε συνελικτικό επίπεδο για τη μείωση των διαστάσεων κάθε εικόνας με σκοπό την αποδοτικότερη εξαγωγή χαρακτηριστικών και στη μείωση του υπολογιστικού κόστους του μοντέλου. Στο πλήρως συνδεδεμένο επίπεδο, το επίπεδο Dense διαθέτει 128 νευρώνες οι οποίοι είναι πλήρως συνδεδεμένοι με όλους τους νευρώνες των προηγούμενων επιπέδων.

Χρησιμοποιήθηκε η τεχνική Dropout για την αποφυγή της υπερπροσαρμογής του μοντέλου στα δεδομένα εκπαίδευσης, όπου το 20% των νευρώνων απενεργοποιούνται σε κάθε στάδιο της εκπαίδευσης.

Ως συνάρτηση ενεργοποίησης ορίστηκε η ReLU για την εξαγωγή αφαιρετικών χαρακτηριστικών και την επιτάχυνση της εκπαίδευσης του μοντέλου. Τέλος, ως έξοδος ορίστηκε ένα επίπεδο Dense με έναν μόνο νευρώνα, με γραμμική ενεργοποίηση linear , η οποία είναι κατάλληλη για προβλήματα παλινδρόμησης.

4.5.4 Αξιολόγηση μετρικών του νευρωνικού δικτύου

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε σε 30 εποχές στις οποίες αξιολογούνταν η συνάρτηση απώλειας του δικτύου κάνοντας χρήση της συνάρτησης βελτιστοποίησης Adam. Ο ρυθμός μάθησης (learning rate) αρχικά ορίζεται ίσος με $4 \cdot 10^{-5}$ και καθορίζει πόσο μεγάλο βήμα πρέπει να κάνει ο αλγόριθμος για την ενημέρωση των βαρών του μοντέλου. Μειώνεται κατά 30% εάν η μετρική της απώλειας δεν μειωθεί με το πέρας 2 εποχών. Έχει ως στόχο την αργή αλλά σταθερή σύγκλιση του μοντέλου ώστε το μοντέλο να μπορέσει να αποδώσει αποδοτικότερα.

Το δίκτυο ξεκίνησε την εκπαίδευση του μοντέλου με έναν ρυθμό $4 \cdot 10^{-5}$ ο οποίος μειώνεται σταδιακά, φτάνοντας την τιμή 10^{-6} στην τελευταία εποχή. Προσαρμόζοντας τον ρυθμό μάθησης κατ' αυτόν τον τρόπο, το μοντέλο συγκλίνει προς τη βέλτιστη τιμή των μετρικών, δηλαδή την σταθεροποίηση τους σε ελάχιστες τιμές.

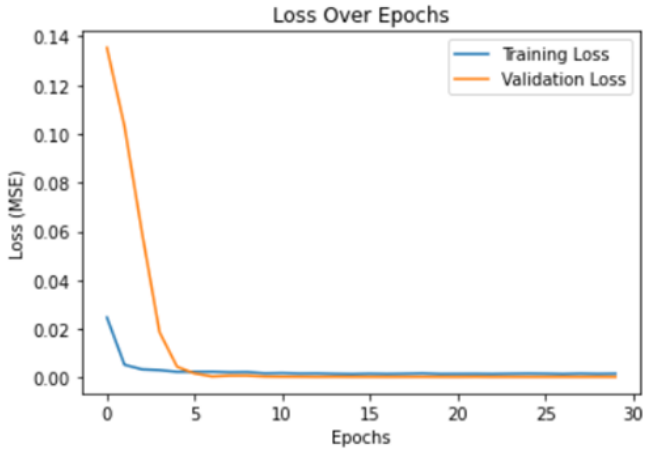


Figure 29: Η πορεία του MSE κατά τις 30 epochs εκπαίδευσης. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή μετά την 5η εποχή και στα δύο σετ δεδομένων

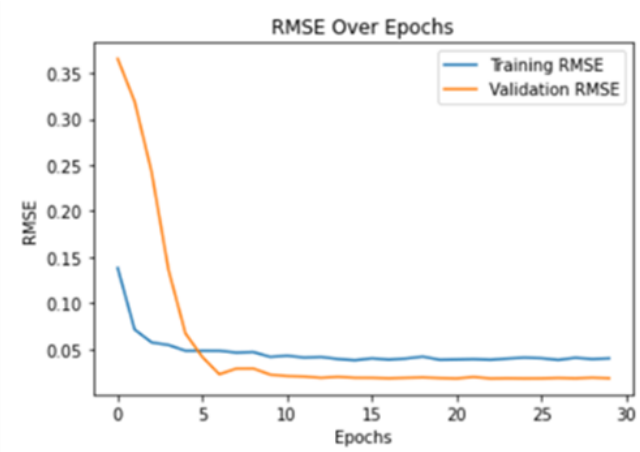
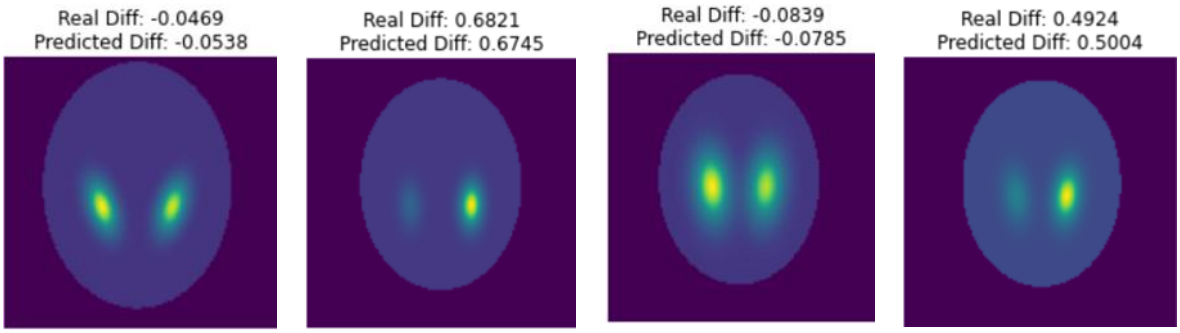


Figure 30: Η πορεία του RMSE κατά τις 30 epochs εκπαίδευσης. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή μετά την 10η εποχή και στα δύο σετ δεδομένων.

Και οι δύο μετρικές MSE και RMSE μειώνονται κατά το πέρας των εποχών ενώ τείνουν να σταθεροποιηθούν σε μία σταθερή τιμή υποδεικνύοντας τη σωστή εκπαίδευση του μοντέλου και στην αυξημένη του ικανότητα στην πρόβλεψη των συνεχών τιμών των ετικετών που αντιστοιχούν σε κάθε ομοίωμα τομογραφικής εικόνας. Ειδικότερα, όσον αφορά την μετρική RMSE, στην 1η εποχή είχε τιμή 0.194, ενώ στην 30η είχε τιμή 0.041 στο σετ εκπαίδευσης ενώ οι τιμές για το σετ επαλήθευσης ήταν 0.369 και 0.019 αντίστοιχα.

Μάλιστα, η συνολική απόδοση στο τεστ αξιολόγησης είναι ίση με $MSE = 0.003$ και $RMSE = 0.017$.



Εικόνα 31: Τυχαίες εικόνες από το δείγμα επαλήθευσης με τις πραγματικές και τις προβλεπόμενες ετικέτες που τους αντιστοιχούν

4.5.5 Συνολική απόδοση του Συνελικτικού Νευρωνικού Δικτύου

Συνολική απόδοση του νευρωνικού δικτύου στην ορθή πρόβλεψη της ποσοτικοποιημένης ανομοιογενούς απορρόφησης του ραδιοφαρμάκου

Παρακάτω παρατίθεται το διάγραμμα των πραγματικών έναντι των προβλεπόμενων τιμών τόσο στο σετ της εκπαίδευσης όσο και στο σετ επαλήθευσης.

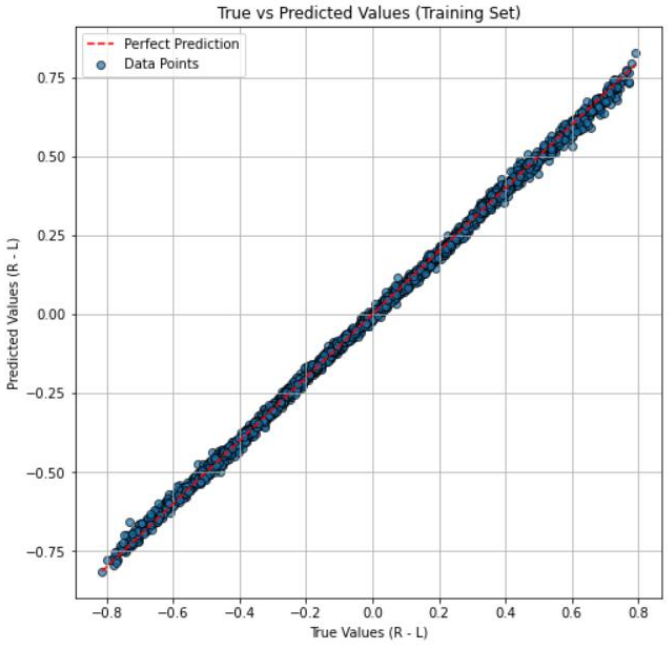


Figure 32: Διάγραμμα πραγματικών – προβλεπόμενων τιμών για τα δεδομένα εκπαίδευσης. Εντοπίζεται η γραμμική συσχέτιση των δύο μεγεθών υποδεικνύοντας την επιτυχή εκπαίδευση του νευρωνικού δικτύου.

Οι τιμές των προβλεπόμενων διαφορών $R' - L'$ συναρτήσει των πραγματικών τιμών στο σετ εκπαίδευσης του μοντέλου παρουσιάζουν γραμμική συσχέτιση καθώς στο αντίστοιχο διάγραμμα $y_{pred} = f(y_{true})$ απεικονίζονται ως μία ευθεία που διέρχεται από την αρχή των αξόνων στο διάστημα $[-1, +1]$ καθώς τόσο οι προβλεπόμενες τιμές όσο

και οι πραγματικές είναι κανονικοποιημένες στο διάστημα $[-1, +1]$ λόγω των κανονικοποιημένων εντάσεων που προκύπτουν από το λογισμικό Simulix. Ανάμεσα στις παραπάνω τιμές ικανοποιείται η σχέση

$$\boldsymbol{y_{pred}} = \boldsymbol{a} \cdot \boldsymbol{y_{true}},$$

Όπου $\boldsymbol{y_{pred}}$: Η διαφορά $R' - L'$ που προβλέπει το μοντέλο

$\boldsymbol{y_{true}}$: Η πραγματική διαφορά $R' - L'$ κάθε εικόνας

$\boldsymbol{a}$: Η κλίση της ευθείας

Ωστόσο υπάρχει ένα σφάλμα πρόβλεψης $error$ καθώς οι τιμές που παράγει το μοντέλο ορίζονται ως $\boldsymbol{y_{pred}} = \boldsymbol{y_{true}} + \boldsymbol{error}$ το οποίο οφείλεται σε τυχαίους παράγοντες.

Οι τιμές (y_{true}, y_{pred}) βρίσκονται συγκεντρωμένες γύρω από την διαγώνιο $y_{pred} = y_{true}$ με την κατανομή των σημείων γύρω από την ευθεία να είναι ισότροπη με τα σφάλματα να κατανέμονται τυχαία και συμμετρικά.

Η συσσώρευση των σημείων γύρω από την ευθεία παρουσιάζεται όταν το μοντέλο έχει υψηλή προβλεπτική ακρίβεια. Ένα μέτρο της διασποράς των προβλέψεων του νευρωνικού δικτύου γύρω από τις πραγματικές ετικέτες των τομογραφικών εικόνων αποτελεί η τυπική απόκλιση. Ορίζεται ως $\boldsymbol{\sigma} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}}$, όπου

$\boldsymbol{x_i} = (\boldsymbol{y_{pred}} - \boldsymbol{y_{true}})$: Η διαφορά των προβλεπόμενων με τις πραγματικές ετικέτες,

$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y_{pred}} - \boldsymbol{y_{true}})$: Ο μέσος όρος των παραπάνω διαφορών για κάθε εικόνα,

$\boldsymbol{N}$: Το συνολικό πλήθος των προβλέψεων

Ειδικότερα, η τυπική απόκλιση των σημείων γύρω από τη διαγώνιο είναι ίση με $\sigma = 0{,}014$ και υποδεικνύει πως το μοντέλο είναι σταθερό και έχει μάθει ορθώς τη σχέση μεταξύ των δεδομένων εισόδου και των δεδομένων εξόδου.

Τα σφάλματα είναι χαμηλά και τυχαία χωρίς σημαντική μεροληψία – προκατάληψη (bias) ή υψηλή διασπορά καθώς είναι ομοιόμορφα και συμμετρικά κατανεμημένα όπως φαίνεται από το αντίστοιχο διάγραμμα.

Η κατανομή των σφαλμάτων $\boldsymbol{error} = \boldsymbol{y_{pred}} - \boldsymbol{y_{true}}$ προκύπτει από την προβολή των σημείων $(\boldsymbol{y_{true}}, \boldsymbol{y_{pred}})$ στην κάθετο ως προς την διαγώνιο $\boldsymbol{y_{pred}} = \boldsymbol{y_{true}}$. Καθώς η κατανομή των σημείων γύρω από τη διαγώνιο είναι ισότροπη, η προβολή τους στην κάθετη διεύθυνση θα έχει μορφή κατανομής Gauss. Απεικονίζεται ως το διάγραμμα $\boldsymbol{frequency} = \boldsymbol{f}(\boldsymbol{error})$ δηλαδή το διάγραμμα σε μορφή ιστογράμματος της συχνότητας εμφάνισης της διαφοράς της προβλεπόμενης τιμής από την πραγματική συναρτήσει της παραπάνω διαφοράς.

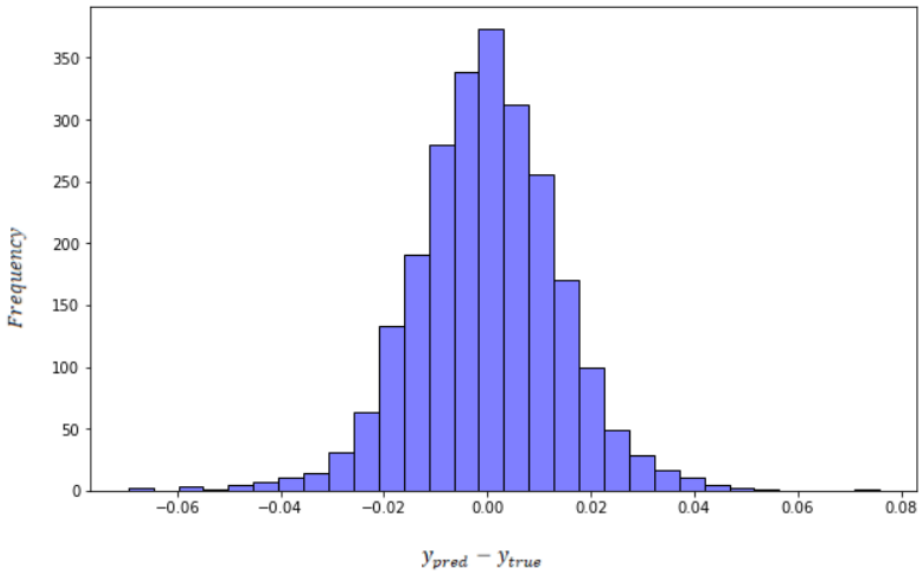


Figure 33: Ιστόγραμμα κατανομής των σφαλμάτων συναρτήσει της συχνότητας εμφάνισης. Παρουσιάζει μορφή κατανομής Gauss με κέντρο το 0.

Η συγκεκριμένη κατανομή Gauss είναι συμμετρική γύρω από το μηδέν, υποδεικνύοντας πως η διακύμανση των σφαλμάτων είναι σταθερή σε όλη την περιοχή των δεδομένων.

Γενικά, η κατανομή Gauss (Γκαουσιανή κατανομή) αποτελεί μία από τις σημαντικότερες κατανομές πιθανότητας τόσο στα μαθηματικά όσο και στις επιστήμες. Έχει χαρακτηριστικό σχήμα καμπάνας και ορίζεται από τη μέση τιμή που καθορίζει το κέντρο της κατανομής και από την τυπική της απόκλιση που εκφράζει τη διασπορά των δεδομένων γύρω από τη μέση τιμή.

Το ιστόγραμμα υποδεικνύει την επιτυχή εκπαίδευση του μοντέλου καθώς οι υψηλές συχνότητες εμφάνισης είναι συσσωρευμένες κοντά στο 0, υποδεικνύοντας πολύ μικρές διαφορές ανάμεσα στις πραγματικές ετικέτες και στις προβλέψεις του μοντέλου. Αντίστοιχα, οι μεγαλύτερες διαφορές παρουσιάζουν μικρότερη συχνότητα εμφάνισης καθώς εντοπίζονται στα άκρα της κατανομής, ενισχύοντας το επιχείρημα μη ύπαρξης σημαντικής μεροληψίας.

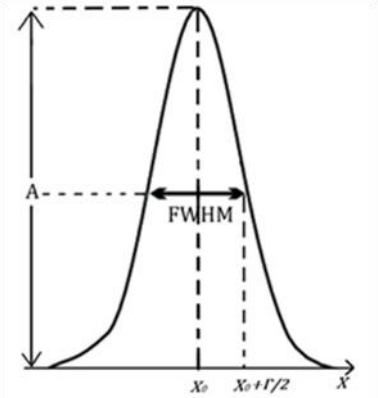
Εκφράζεται από τον τύπο: $\boldsymbol{f(x) = A\, e^{-\frac{(x-x_0)^2}{2\sigma^2}}}$

Όπου: $\boldsymbol{x_0 = \mu}$: η μέση τιμή της κατανομής

$\boldsymbol{\sigma}$: Η τυπική απόκλιση της κατανομής

$\boldsymbol{A = \frac{1}{\sigma\sqrt{2\pi}}}$: το μέγιστο ύψος της κατανομής

Το πλάτος της κατανομής περιγράφεται από το πλήρες πλάτος στη μισή μέγιστη τιμή – Full Width at Half Maximum (FWHM) το οποίο εκφράζει το εύρος της καμπύλης στο ύψος που αντιστοιχεί στο ήμισυ της μέγιστης τιμής του. Ορίζεται ως:

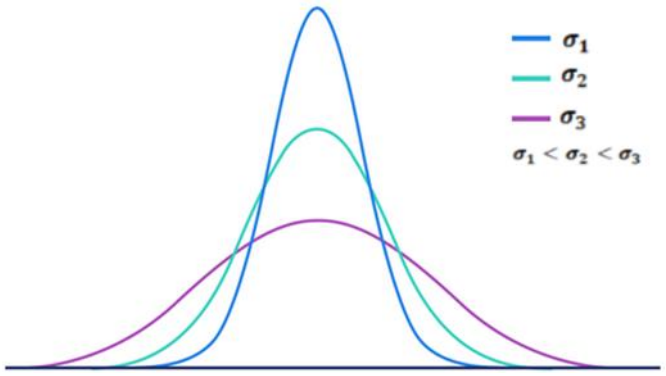


Εικόνα 34: Κατανομή Gauss με τα χαρακτηριστικά μεγέθη

$$\Gamma = FWHM \Rightarrow f\left(x_0 + \frac{\Gamma}{2}\right) = \frac{A}{2} = f\left(x_0 - \frac{\Gamma}{2}\right) \Rightarrow$$

$$A \cdot e^{-\frac{\left(\frac{\Gamma}{2}\right)^2}{2\sigma^2}} = \frac{A}{2} \Rightarrow ln2 = \left(\frac{\Gamma}{2}\right)^2 \cdot \frac{1}{2\sigma^2} \Rightarrow \Gamma = 2\sqrt{2ln2}\sigma \Rightarrow \quad \Gamma \cong 2.355\sigma$$

Συνεπώς, για μεγαλύτερες τυπικές αποκλίσεις, η κατανομή Gauss θα έχει μεγαλύτερο πλάτος.



Εικόνα 35: Η μεταβολή των χαρακτηριστικών της κατανομής Gauss αυξανομένης της τυπικής απόκλισης σ.

4.5.6 Συνολική απόδοση του CNN κατά την ελλιπή εκπαίδευση του δικτύου

Στην περίπτωση ελλιπούς εκπαίδευσης του νευρωνικού δικτύου, θα παρατηρούνται συστηματικές αβεβαιότητες και η αντίστοιχη κατανομή $\boldsymbol{frequency = f(error)}$ θα παρουσιάζει μεγαλύτερο πλάτος λόγω μεγαλύτερης τυπικής απόκλισης. Για την υποστήριξη του παραπάνω επιχειρήματος, πραγματοποιήθηκε εκπαίδευση του μοντέλου για μόνο 5 εποχές.

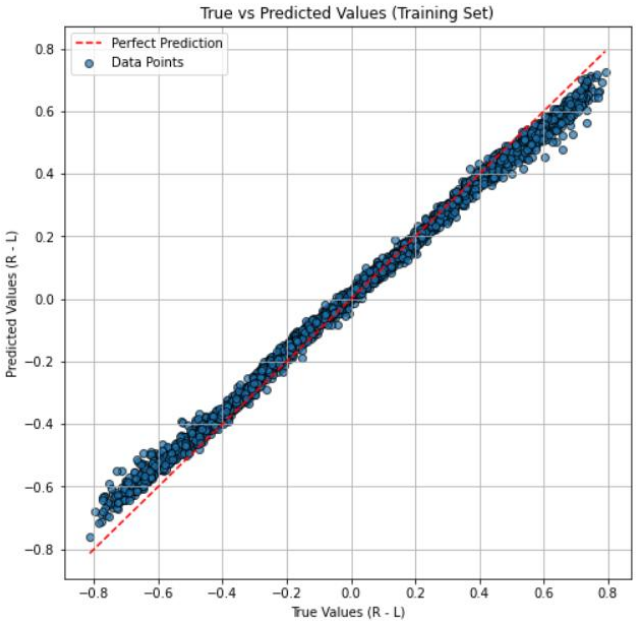


Figure 8: Διάγραμμα πραγματικών – προβλεπόμενων τιμών για τα δεδομένα εκπαίδευσης. Εντοπίζεται ξανά γραμμική συσχέτιση των δύο μεγεθών στραμμένη συμμετρικά δεξιόστροφα με μεγαλύτερη διασπορά των σημείων υποδεικνύοντας συστηματική αβεβαιότητα στις ακραίες τιμές.

Τα σημεία, στην προκείμενη περίπτωση, παρουσιάζουν μεγαλύτερη διασπορά γύρω από την διαγώνιο, δηλαδή είναι τοποθετημένα σε μία ευρεία ζώνη γύρω από την ευθεία $\boldsymbol{y_{pred} = y_{true}}$ γεγονός που επιβεβαιώνεται από τη μεγαλύτερη τυπική απόκλιση:

$\boldsymbol{\sigma = 0,527}$.

Τα σημεία είναι συσσωρευμένα σε μία ευθεία η οποία παρουσιάζει δεξιόστροφη κλίση σε σχέση με τη διαγώνιο στις ακραίες τιμές που υποδεικνύουν μεγαλύτερη διαφορά στις απορροφούμενες εντάσεις. Η απόκλιση αυτή είναι συστηματική καθώς το σφάλμα δεν είναι τυχαίο, παρουσιάζει συγκεκριμένο μοτίβο που εξαρτάται από τη σχέση των y_{pred}, y_{true}.

Πράγματι, η μορφή του ιστογράμματος $\boldsymbol{frequency = f(error)}$ είναι Γκαουσιανή κατανομή μεγαλύτερου πλάτους από το αντίστοιχο ιστόγραμμα της ολοκληρωμένης εκπαίδευσης, υποδεικνύοντας ξανά τη μεγαλύτερη τυπική απόκλιση των προβλεπόμενων τιμών σε σχέση με τις πραγματικές.

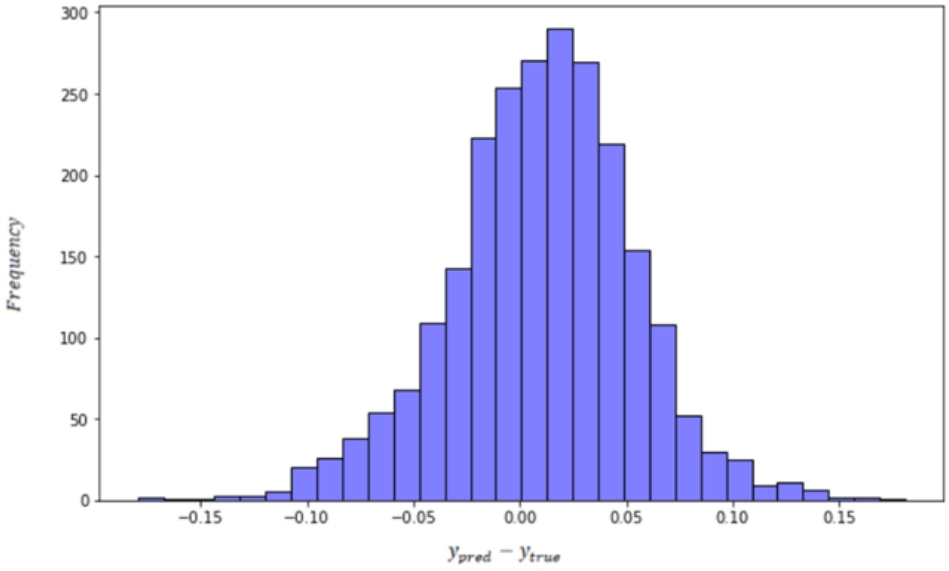


Figure 9: Ιστόγραμμα κατανομής των σφαλμάτων συναρτήσει της συχνότητας εμφάνισης. Παρουσιάζει μορφή κατανομής Gauss μεγαλύτερου πλάτους υποδεικνύοντας μεγαλύτερη τυπική απόκλιση των σφαλμάτων συγκριτικά με την ολοκληρωμένη εκπαίδευση.

Το σφάλμα , παρουσιάζει συγκεκριμένο μοτίβο που εξαρτάται από τη σχέση των y_{pred}, y_{true}, είναι μη τυχαίο και επαναλαμβανόμενο. Είναι δηλαδή συστηματικό με μορφή σταθερά υψηλότερης διασποράς στα άκρα.

Το συστηματικό σφάλμα οφείλεται σε μη επαρκή εκπαίδευση του μοντέλου. Το νευρωνικό δίκτυο στην προκείμενη περίπτωση δεν έχει δει αρκετές φορές τα δεδομένα και παρουσιάζει δυσκολία στην εύρεση της σχέσης μεταξύ των δεδομένων εισόδου και των δεδομένων εξόδου σε μεγαλύτερες διαφορές των απορροφούμενων εντάσεων των εγκεφαλικών ημισφαιρίων και ως εκ τούτου δεν αποδίδει ορθώς.

Ως εκ τούτου, είναι απαραίτητη η ολοκλήρωση της εκπαίδευσης για τις 30 εποχές προκειμένου το νευρωνικό δίκτυο να μάθει τη συσχέτιση των προβλεπόμενων τιμών y_{pred} με τις πραγματικές τιμές y_{true} για να ανιχνεύει και να ποσοτικοποιεί με ακρίβεια την ανομοιογενή απορρόφηση του ραδιοφαρμάκου στους λοβούς του εγκεφάλου και να είναι ικανό να γενικεύει με ακρίβεια σε περιπτώσεις τομογραφικών εικόνων που δεν έχει δει ξανά.

4.6 Ανάπτυξη CNN για την πρόβλεψη της απορροφούμενης έντασης κάθε λοβού και του υποβάθρου

Πρόβλεψη της απορροφούμενης έντασης κάθε λοβού και του υποβάθρου μέσω CNN με τρεις εξόδους

4.6.1 Σκοπός του δεύτερου CNN

Για την επέκταση της αρχικής μελέτης, της ποσοτικοποίησης της ανομοιογενούς απορρόφησης του ραδιοφαρμάκου ^{123}I, κρίθηκε απαραίτητο να κατασκευαστεί ένα

δεύτερο συνελικτικό νευρωνικό δίκτυο το οποίο θα προβλέπει τις απορροφούμενες εντάσεις κάθε εγκεφαλικού λοβού από εξέταση DaTSCAN ξεχωριστά όπως επίσης και την απορροφούμενη ένταση του υποβάθρου.

4.6.2 Μεθοδολογία κατασκευής του CNN

Αρχικά, δίνονται ως δεδομένα εισόδου στο νευρωνικό δίκτυο οι εικόνες των ομοιωμάτων σε μορφή πινάκων και οι παράμετροι που αφορούν την κάθε εικόνα. Από τις τιμές των παραμέτρων απομονώνονται εκείνες των εντάσεων του υποβάθρου όπως και του δεξιού και του αριστερού λοβού. Στη συνέχεια οι πίνακες μετασχηματίζονται σε εικόνες και δημιουργείται μία κενή λίστα για την αποθήκευση τους μαζί με τις αντίστοιχες ετικέτες.

Το σετ δεδομένων, συνολικού όγκου 3000 εικόνων, χωρίζεται σε training data (80% του αρχικού δείγματος) και σε test data (20% του αρχικού δείγματος). Η εκπαίδευση του μοντέλου ολοκληρώνεται σε 50 κύκλους εκπαίδευσης (epochs) κατά την ολοκλήρωση των οποίων παρακολουθείται η εξέλιξη των μετρικών.

Οι μετρικές στο συγκεκριμένο πρόβλημα επιλέχθηκαν να είναι το μέσο τετραγωνικό σφάλμα MSE (Mean Squared Error) και η μέση τετραγωνική απόκλιση RMSE (Root Mean Squared Error) καθώς πρόκειται πάλι για πρόβλημα παλινδρόμησης στο οποίο ζητείται από το νευρωνικό δίκτυο η πρόβλεψη τριών συνεχών τιμών, των εντάσεων των δύο εγκεφαλικών λοβών και την ένταση του υποβάθρου.

4.6.3 Ανάλυση αρχιτεκτονικής του CNN

Το dataset χωρίστηκε με τον εξής τρόπο:

Σετ εκπαίδευσης (Training set): 80% των παραγόμενων εικόνων, δηλαδή 2400 εικόνες χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου

Σετ επαλήθευσης (Validation set): 20% των παραγόμενων εικόνων, δηλαδή 600 εικόνες χρησιμοποιήθηκαν για την επικύρωση του μοντέλου.

Τα δύο σετ δεν αναμειγνύονται προκειμένου η αξιολόγηση να γίνει πάνω σε άγνωστες για το μοντέλο εικόνες.

Η αρχιτεκτονική του μοντέλου είναι όμοια με εκείνη του προηγούμενου συνελικτικού νευρωνικού δικτύου για την ποσοτικοποίηση της ανομοιογενούς απορρόφησης. Η μόνη διαφορά έγκειται στο επίπεδο εξόδου που χρησιμοποιείται η συνάρτηση Dense με 3 νευρώνες για την εξαγωγή των τιμών των τριών εντάσεων. Η ενεργοποίηση είναι και πάλι γραμμική καθώς πρόκειται για νευρωνικό δίκτυο παλινδρόμησης.

Ως συνάρτηση βελτιστοποίησης χρησιμοποιείται η συνάρτηση Adam και ως συνάρτηση ενεργοποίησης ορίστηκε η συνάρτηση ReLU.

4.6.4 Αξιολόγηση μετρικών του CNN

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε σε 50 εποχές κατά τη διάρκεια των οποίων προσαρμόζεται ο ρυθμός μάθησης διαρκώς. Επιλέχθηκαν παραπάνω εποχές εκπαίδευσης του δικτύου CNN λόγω της αυξημένης πολυπλοκότητας καθώς ζητείται η πρόβλεψη τριών τιμών για κάθε εικόνα με την οποία τροφοδοτείται το δίκτυο. Ειδικότερα, ο αρχικός ρυθμός μάθησης ορίζεται να είναι ίσος με $4 \cdot 10^{-5}$, ενώ με διαδοχικές μειώσεις κατά 30%, μειώνεται τελικά σε 10^{-7} επιτυγχάνοντας σταθεροποίηση των μετρικών σε ελάχιστες τιμές.

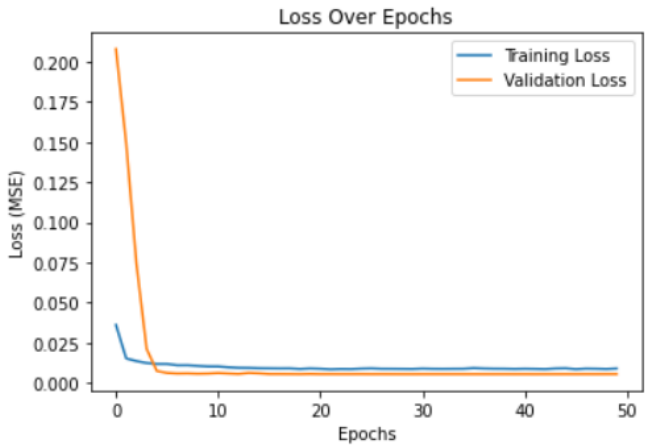


Figure 10: Η πορεία του MSE κατά τις 50 epochs εκπαίδευσης. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή μετά την 5η εποχή και στα δύο σετ δεδομένων.

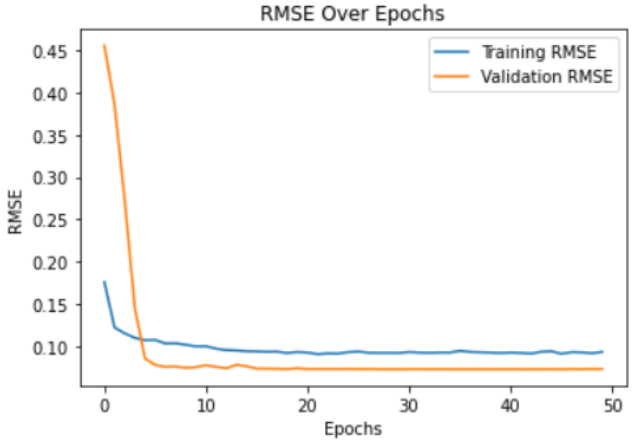


Figure 11: Η πορεία του RMSE κατά τις 50 epochs εκπαίδευσης. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή στην 20η εποχή για το σετ εκπαίδευσης και στην 14η εποχή στο σετ επαλήθευσης.

Και οι δύο μετρικές MSE και RMSE μειώνονται κατά το πέρας των εποχών ενώ τείνουν να σταθεροποιηθούν σε μία σταθερή τιμή υποδεικνύοντας τη σωστή εκπαίδευση του μοντέλου και στην αυξημένη του ικανότητα στην πρόβλεψη των συνεχών τιμών των εντάσεων του αριστερού και του δεξιού λοβού που αντιστοιχεί σε κάθε ομοίωμα εξέτασης DaTSCAN όπως επίσης και στην απορροφούμενη ένταση του υποβάθρου.

Πιο συγκεκριμένα, η μέση τετραγωνική απόκλιση (RMSE) στο σετ δεδομένων εκπαίδευσης ξεκινάει από 0.1760 στην 1η εποχή και ελαχιστοποιείται σε 0.0936 στην 40η εποχή. Αντίστοιχα, αναφορικά με το σετ δεδομένων επαλήθευσης, στην 1η εποχή η μετρική RMSE είναι ίση με 0.455 και στην 50η εποχή μειώνεται σε 0.073.

Συνολικά, η μέση τετραγωνική απόκλιση για την ορθή πρόβλεψη της απορροφούμενης έντασης του αριστερού λοβού είναι $RMSE = 0.089$ όπως επίσης και για τον δεξιά λοβό. Όσον αφορά την ορθή πρόβλεψη της έντασης του υποβάθρου, η μέση τετραγωνική απόκλιση είναι ίση με $RMSE = 0.027$.

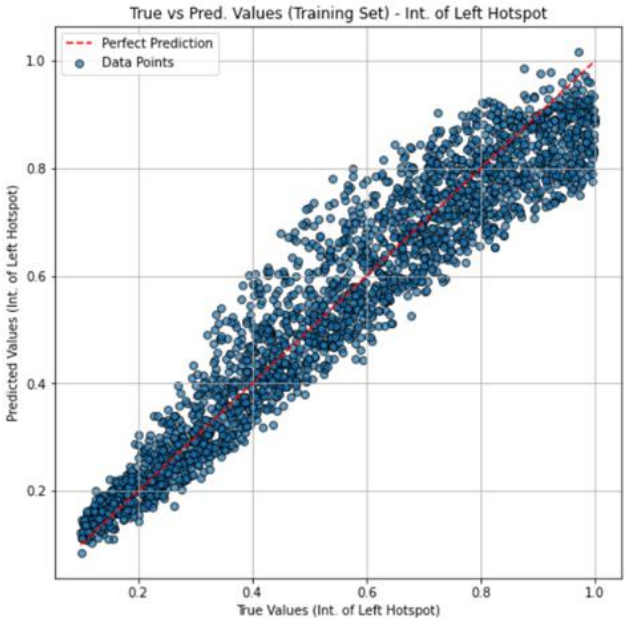


Figure 12: Διάγραμμα πραγματικών – προβλεπόμενων τιμών για τα δεδομένα εκπαίδευσης στην πρόβλεψη της απορροφούμενης έντασης του αριστερού λοβού. Εντοπίζεται γραμμική συσχέτιση των δύο μεγεθών με αυξανόμενη διασπορά γύρω από την διαγώνιο αυξανομένης της έντασης.

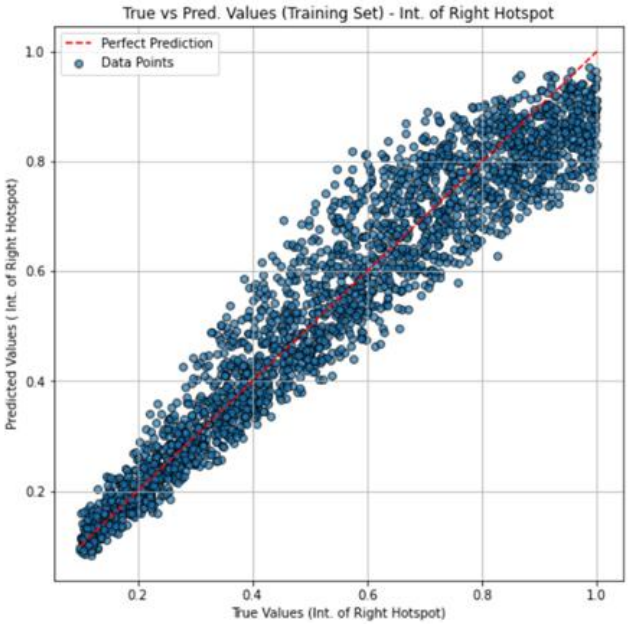


Figure 13: Διάγραμμα πραγματικών – προβλεπόμενων τιμών για τα δεδομένα εκπαίδευσης στην πρόβλεψη της απορροφούμενης έντασης του δεξιού λοβού. Εντοπίζεται παρόμοια συμπεριφορά με τις προβλέψεις του αριστερού hotspot.

Παρατηρώντας τα διαγράμματα $y_{pred} = f(y_{true})$ για κάθε λοβό, οι τιμές των προβλεπόμενων τιμών y_{pred} σε σχέση με τις πραγματικές τιμές y_{true} και για τις δύο εντάσεις βρίσκονται γύρω από την διαγώνιο $\boldsymbol{y_{pred} = y_{true}}$ στο διάστημα $[0, 1]$ με αυξανόμενη τη διασπορά των τιμών γύρω από την διαγώνιο καθώς αυξάνεται η απόστασή τους από αυτήν αυξανομένης της απορροφούμενης έντασης, οδηγώντας σε λιγότερο ακριβείς προβλέψεις για μεγαλύτερες τιμές.

Η μορφή και των δύο διαγραμμάτων είναι όμοια, γεγονός που υποδηλώνει ότι το συνελικτικό νευρωνικό δίκτυο έχει εκπαιδευτεί με παρόμοιο τρόπο και για τα δύο ημισφαίρια και δεν παρουσιάζει συστηματική αβεβαιότητα.

Μάλιστα, η τυπική απόκλιση της διαφοράς των προβλεπόμενων από τις πραγματικές τιμές της απορροφούμενης έντασης του αριστερού λοβού είναι ίση με την τυπική απόκλιση της αντίστοιχης διαφοράς για την απορροφούμενη ένταση του δεξιού λοβού: $\sigma_1 = 0.078 = \sigma_2$ υποδεικνύοντας πως το νευρωνικό δίκτυο δεν υπερεκτιμά ή υποτιμά κάποια πλευρά. Συμπερασματικά, το δίκτυο δεν παρουσιάζει συστηματική απόκλιση και οι προβλέψεις του είναι σχετικά ακριβείς χωρίς να υπάρχει μεροληψία προς έναν από τους δύο λοβούς.

Καθώς οι τιμές της έντασης του υποβάθρου είναι πολύ μικρές, το συνελικτικό νευρωνικό δίκτυο δυσκολεύεται να ανιχνεύσει τις μικρές διαφοροποιήσεις ανάμεσα στις πραγματικές και στις προβλεπόμενες τιμες, γεγονός που οδηγεί σε συστηματική υποτίμηση των τιμών που προβλέπει, όπως φαίνεται στο αντίστοιχο διάγραμμα $y_{pred} = f(y_{true})$. Η τυπική απόκλιση των πραγματικών από τις προβλεπόμενες τιμές είναι ίση με $\sigma_3 = 0.025$, γεγονός που υποδεικνύει πως οι προβλέψεις του νευρωνικού δικτύου είναι πολύ κοντά στις πραγματικές τιμές και έτσι η συστηματική υποτίμηση των τιμών είναι μικρή.

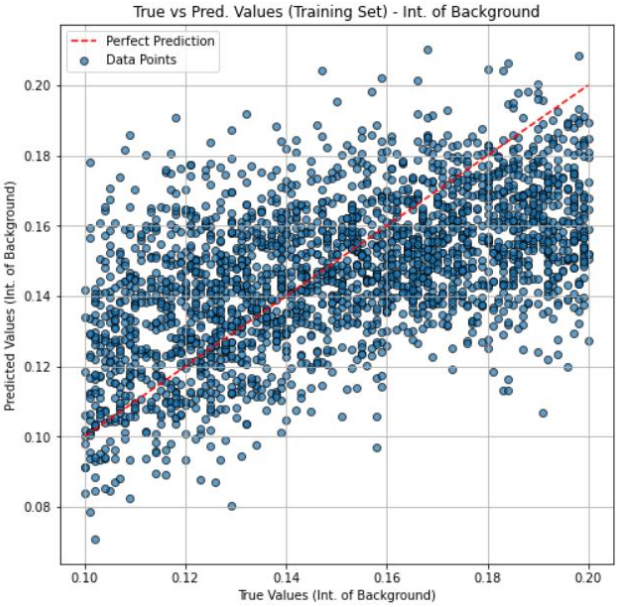
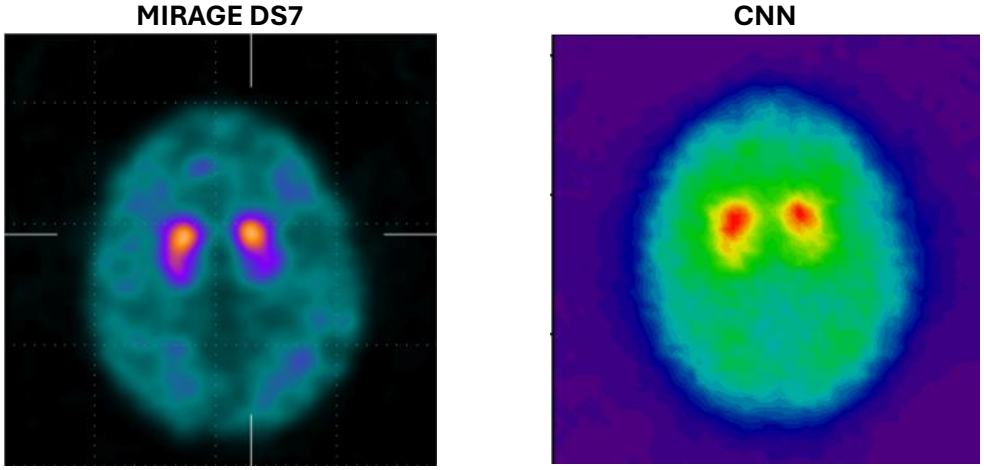


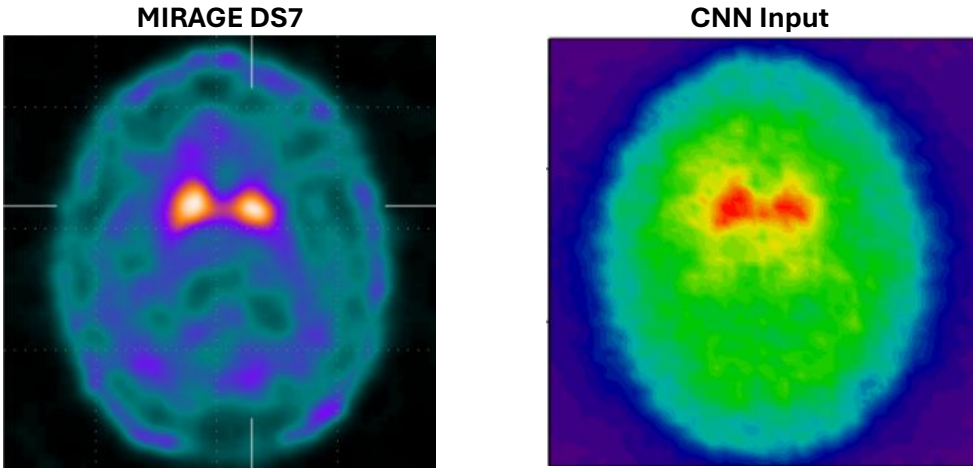
Figure 14: Διάγραμμα πραγματικών – προβλεπόμενων τιμών για τα δεδομένα εκπαίδευσης στην πρόβλεψη της απορροφούμενης έντασης του υποβάθρου. Εντοπίζεται ελαφριά συστηματική υποτίμηση των τιμών.

4.7 Εφαρμογή των CNN σε πραγματικά δεδομένα DaTSCAN

Για τους σκοπούς της παρούσης εργασίας δόθηκαν δύο ιατρικές εικόνες DaTSCAN από γ- Camera MIRAGE DS7 οι οποίες τροφοδοτήθηκαν στα δύο συνελικτικά νευρωνικά δίκτυα, οι εικόνες Exam A και Exam B.



Εικόνα 43: Οι απεικονίσεις της εικόνας της εξέτασης Α. Αριστερά φαίνεται η εικόνα από τη γ – Camera και δεξιά η εικόνα που εμφανίζεται στο δίκτυο κατά την επεξεργασία του αντίστοιχου πίνακα.



Εικόνα 44: Οι απεικονίσεις της εικόνας της εξέτασης Β. Αριστερά φαίνεται η εικόνα από τη γ – Camera και δεξιά η εικόνα που εμφανίζεται στο δίκτυο κατά την επεξεργασία του αντίστοιχου πίνακα.

Το συνελικτικό νευρωνικό δίκτυο για την ποσοτικοποίηση της ανομοιογενούς απορρόφησης του ραδιοφαρμάκου για κάθε ιατρική εικόνα έδωσε τις εξής προβλέψεις:

$$\boldsymbol{R' - L'} = \begin{cases} \mathbf{-0.1673}, & \boldsymbol{Exam\ A} \\ \mathbf{+0.0504}, & \boldsymbol{Exam\ B} \end{cases}$$

Αντίστοιχα, το συνελικτικό νευρωνικό δίκτυο για τις προβλέψεις των εντάσεων του υποβάθρου και των δύο λοβών έδωσε τις παρακάτω προβλέψεις:

$$\boldsymbol{Exam\ A}: \begin{cases} \mathbf{0.0922}, & \boldsymbol{Υπόβαθρο} \\ \mathbf{0.6786}, & \boldsymbol{Αριστερός\ λοβός}, \\ \mathbf{0.6531}, & \boldsymbol{Δεξιός\ λοβός} \end{cases}$$

$$\boldsymbol{Exam\ B}: \begin{cases} \mathbf{0.0675}, & \boldsymbol{Υπόβαθρο} \\ \mathbf{0.5928}, & \boldsymbol{Αριστερός\ λοβός} \\ \mathbf{0.6378}, & \boldsymbol{Δεξιός\ λοβός} \end{cases}$$

Προκειμένου να γίνει σύγκριση με τις πραγματικές τιμές των ιατρικών εικόνων, απομονώθηκαν σε υποπίνακες οι περιοχές ενδιαφέροντος από τις εικόνες της γ – Camera, δηλαδή το κάθε hotspot και το υπόβαθρο με σκοπό τη λήψη του μέσου (mean) και της μέγιστης τιμής (max)..

Στις προβλεπόμενες τιμές έντασης κάθε λοβού από το δεύτερο νευρωνικό δίκτυο προστίθεται η τιμή του υποβάθρου, καθώς το CNN λειτουργεί αθροιστικά. Το μοντέλο έχει εκπαιδευτεί να δέχεται ως είσοδο τις επιμέρους εντάσεις κάθε λοβού χωρίς το υπόβαθρο και να τις προβλέπει ξεχωριστά. Ωστόσο, στις πραγματικές μετρήσεις περιλαμβάνεται και το υπόβαθρο και προκειμένου η σύγκριση να έχει νόημα, η τιμή αυτή πρέπει να προστεθεί και στις προβλέψεις του CNN.

Προκύπτουν οι παρακάτω συγκεντρωτικοί πίνακες αποτελεσμάτων:

Exam A								
Υπόβαθρο (B)			**Δεξιός Λοβός (R)**			**Αριστερός Λοβός (L)**		
CNN output	**Mean**	**Max**	**CNN output**	**Mean**	**Max**	**CNN output**	**Mean**	**Max**
0.0922	0.0726	0.1105	0.7453	0.7823	1.079	0.7708	0.8150	1.072

Exam B								
Υπόβαθρο (B)			Δεξιός Λοβός (R)			Αριστερός Λοβός (L)		
CNN output	Mean	Max	CNN output	Mean	Max	CNN output	Mean	Max
0.0675	0.0541	0.0640	0.7053	0.6851	0.7786	0.6603	0.6274	0.7340

Κατά τη σύγκριση της εξόδου του συνελικτικού νευρωνικού δικτύου με τη μέση τιμή των υποπινάκων με τις περιοχές ενδιαφέροντος προκύπτουν οι παρακάτω ποσοστιαίες αποκλίσεις:

	Exam A	Exam B
Δεξιός Λοβός (R)	4,73%	2,95%
Αριστερός Λοβός (L)	5,42%	5,24%
Υπόβαθρο (B)	27%	24,76%

Και στις δύο ιατρικές εικόνες παρατηρείται πως η προβλεπόμενη τιμή του υποβάθρου δεν προσεγγίζει τη μέση τιμή αλλά την μέγιστη, δηλαδή υπάρχει μία υπερεκτίμηση των τιμών, γεγονός που υποδεικνύει τη δυσκολία του CNN να ανιχνεύσει τις μικρές διαφοροποιήσεις ανάμεσα στις πραγματικές και στις προβλεπόμενες τιμες του υποβάθρου και συνεπώς παρατηρούνται οι αποκλίσεις 27% και 24.76% για κάθε ιατρική εικόνα αντίστοιχα.

Η μέγιστη απόκλιση των προβλεπόμενων τιμών των εντάσεων των δύο εγκεφαλικών λοβών είναι της τάξης του 5%, γεγονός που υποδηλώνει πως το μοντέλο είναι σε θέση να κάνει ορθές προβλέψεις με μία μικρή ποσοστιαία απόκλιση χωρίς υποτίμηση ή υπερεκτίμηση κάποιας πλευράς. Οι αποκλίσεις αυτές αφενός οφείλονται στην αυξημένη πολυπλοκότητα του νευρωνικού δικτύου καθώς καλείται να προβλέψει τρεις διαφορετικές εξόδους. Αφετέρου, οι πραγματικές τομοσπινθηρογραφικές εικόνες δεν ανταποκρίνονται με απόλυτο τρόπο στα θεωρητικά ομοιώματα λόγω μικρών μορφολογικών αποκλίσεων ή ιδιαιτεροτήτων εντός των περιοχών ενδιαφέροντος, οι οποίες είναι αναμενόμενες σε πραγματικές ιατρικές εικόνες αλλά και λόγω πιθανού θορύβου κατά τη διάρκεια της εξέτασης.

Καταληκτικά, για την επιχειρηματολογία που αναπτύχθηκε παραπάνω, οι παρατηρούμενες ποσοστιαίες αποκλίσεις θεωρούνται αποδεκτές και ως εκ τούτου το συνελικτικό νευρωνικό δίκτυο που αναπτύχθηκε μπορεί να λειτουργήσει επικουρικά στην διάγνωση από τον αντίστοιχο ειδικό θεράποντα ιατρό.

5. Σύνοψη

Στην παρούσα εργασία μελετήθηκαν τα νευρωνικά δίκτυα και η ικανότητά τους να ταξινομούν ορθώς εικόνες με βάση τα χαρακτηριστικά τους αλλά και να πραγματοποιούν προβλέψεις.

Αρχικά, αναπτύχθηκαν εισαγωγικά νευρωνικά δίκτυα που δέχονταν ως είσοδο εικόνες συγκεκριμένων χαρακτηριστικών με γεωμετρικά σχήματα οι οποίες κατασκευάστηκαν από τα ίδια τα νευρωνικά δίκτυα.

Ειδικότερα, το πρώτο εισαγωγικό πρόβλημα αφορά τη δυαδική ταξινόμηση δείγματος συνολικού όγκου 3000 εικόνων διαστάσεων 128x128, από τις οποίες οι 1500 περιλαμβάνουν έναν κύκλο μεταβλητών διαστάσεων τυχαία τοποθετημένο στο χώρο και οι άλλες 1500 περιλαμβάνουν ένα τετράγωνο επίσης μεταβλητών διαστάσεων και τυχαία τοποθετημένο στο χώρο. Σκοπό του προβλήματος αποτέλεσε η ταξινόμηση των εικόνων σε δύο κατηγορίες, ανάλογα με το γεωμετρικό σχήμα που περιλαμβάνει η κάθε εικόνα. Αρχικά κατασκευάστηκε το απαιτούμενο δείγμα και στη συνέχεια το νευρωνικό δίκτυο εκπαιδεύτηκε με στόχο την επίτευξη του σκοπού. Γενικά, η απόδοση του μοντέλου στην δυαδική ταξινόμηση των εικόνων ως προς το είδος του απλού γεωμετρικού σχήματος που περιλαμβάνουν μπορεί να θεωρηθεί επιτυχής καθώς η συνολική ακρίβεια μετά το πέρας των εποχών εκπαίδευσης είναι 100% ενώ η συνάρτηση της απώλειας είναι της τάξης του $\sim 4{,}7 \cdot 10^{-4}$.

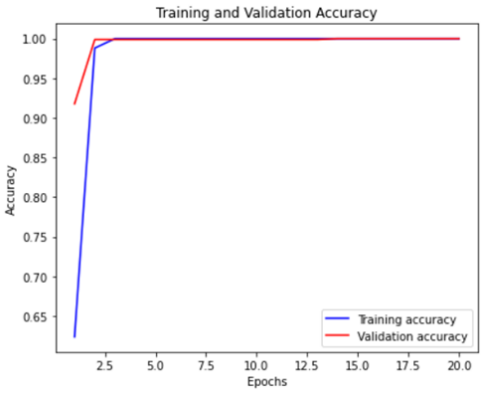


Figure 45: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται σταθεροποίηση σε μέγιστη τιμή

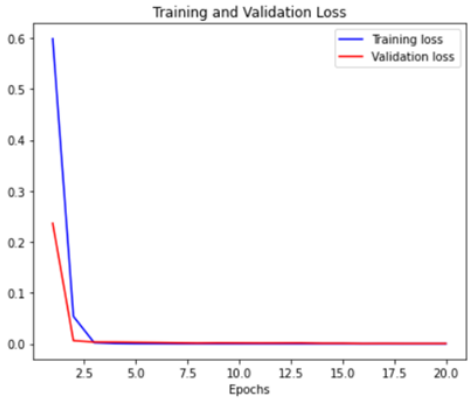


Figure 46: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 20 εποχών. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή

Το δεύτερο εισαγωγικό πρόβλημα αφορά την ταξινόμηση εικόνων που περιλαμβάνουν 1 έως 4 ίδια γεωμετρικά σχήματα ίδιου μεγέθους ως προς το πλήθος αυτών σε κάθε εικόνα. Κατασκευάστηκε δείγμα συνολικού όγκου 3000 εικόνων διαστάσεων 128x128 οι οποίες περιλαμβάνουν 1 έως 4 τετράγωνα ίδιου μεγέθους χωρίς να επικαλύπτονται τα σχήματα μεταξύ τους και το νευρωνικό δίκτυο εκπαιδεύτηκε ως προς την ταξινόμηση των εικόνων σε 4 κατηγορίες ανάλογα με το πλήθος των γεωμετρικών σχημάτων που περιλαμβάνει η κάθε εικόνα. Στη συνέχεια, κατασκευάστηκε δείγμα συνολικού όγκου 3000 εικόνων διαστάσεων 128x128 οι οποίες περιλαμβάνουν 1 έως 4 τετράγωνα ίδιου μεγέθους με επιτρεπτή την επικάλυψη μεταξύ τους και το νευρωνικό δίκτυο εκπαιδεύτηκε με τον ίδιο σκοπό. Σκοπό του συγκεκριμένου προβλήματος δεν ήταν μόνο η διαδικασία ορθής εκπαίδευσης των μοντέλων αλλά και η

σύγκριση των τυπικών αποκλίσεων των δύο νευρωνικών δικτύων κατά την αύξηση της πολυπλοκότητας.

Συνολικά, στις εικόνες αξιολόγησης του μοντέλου με επιτρεπτή την αλληλοεπικάλυψη μεταξύ των γεωμετρικών σχημάτων, η συνάρτηση της απώλειας είναι της τάξης του 0,408 ενώ η ακρίβεια φτάνει την τιμή 0,825. Αντίστοιχα, για το μοντέλο χωρίς αλληλοεπικάλυψη μεταξύ των γεωμετρικών σχημάτων η συνάρτηση της απώλειας είναι της τάξης του 0,058 ενώ η ακρίβεια φτάνει την τιμή 1,00.

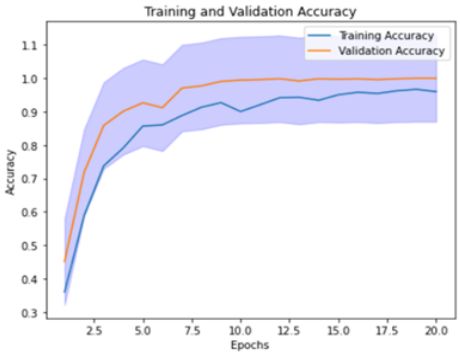


Figure 47: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 20 εποχών στο ANN χωρίς αλληλοεπικάλυψη. Παρατηρείται ανοδική πορεία και σταθεροποίηση στην 16η εποχή

Figure 48: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 20 εποχών στο ANN χωρίς αλληλοεπικάλυψη. Παρατηρείται συνεχής μείωση και έπειτα σταθεροποίηση στην 16η εποχή

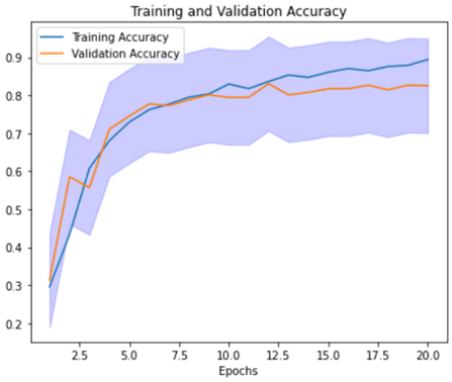


Figure 49: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 20 εποχών στο ANN με αλληλοεπικάλυψη. Παρατηρείται ανοδική πορεία και σταθεροποίηση στις τελευταίες εποχές

Figure 50: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 20 εποχών στο ANN με αλληλοεπικάλυψη. Παρατηρείται συνεχής μείωση και έπειτα σταθεροποίηση στην 15η εποχή

Παρατηρήθηκε πως οι ζώνες των τυπικών αποκλίσεων είναι ξεκάθαρα διαχωρισμένες στις περισσότερες εποχές. Αυτό σημαίνει πως το μοντέλο το οποίο τροφοδοτείται με λιγότερο πολύπλοκα δεδομένα αποδίδει πιο αποτελεσματικά και μπορεί να αποδώσει πιο σταθερά από όταν τα δεδομένα είναι περισσότερο πολύπλοκα, όταν δηλαδή περιλαμβάνουν επικάλυψη μεταξύ των γεωμετρικών σχημάτων. Συνολικά, η σαφής διαφορά στις τυπικές αποκλίσεις και η μη επικάλυψη των δύο ζωνών υπογραμμίζει την ανώτερη απόδοση του μοντέλου που τροφοδοτείται με εικόνες χωρίς επικάλυψη των γεωμετρικών σχημάτων. Καταληκτικά, παρατηρήθηκε πως αυξανομένης

της πολυπλοκότητας των δεδομένων, αυξάνεται και η δυσκολία που αντιμετωπίζει το νευρωνικό δίκτυο στην εκμάθηση και ως εκ τούτου στην σωστή πρόβλεψη των κατηγοριών.

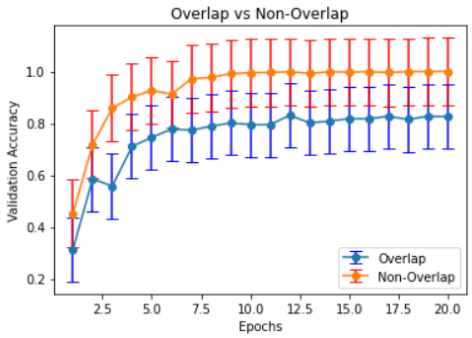


Figure 51: Διάγραμμα των ακριβειών επαλήθευσης με τις αντίστοιχες τυπικές αποκλίσεις για τα δύο νευρωνικά δίκτυα.

Το τρίτο εισαγωγικό πρόβλημα αφορά τη δυαδική ταξινόμηση δείγματος συνολικού όγκου 3000 εικόνων διαστάσεων 128x128, η καθεμία από τις οποίες περιλαμβάνει 2 κύκλους μεταβλητών μεγεθών, τυχαία τοποθετημένων στο χώρο εντός του αρχικού πλαισίου. Ζητούμενο αποτέλεσε η δυαδική ταξινόμηση των εικόνων ως προς την αλληλοεπικάλυψη μεταξύ των γεωμετρικών σχημάτων. Συνολικά, το μοντέλο δυαδικής ταξινόμησης επιτυγχάνει ακρίβεια ορθών προβλέψεων 96% και η λογαριθμική συνάρτηση της απώλειας φτάνει την τιμή $0{,}118$ καθιστώντας την εκπαίδευση επιτυχή.

Figure 52: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται ανοδική πορεία και σταθεροποίηση μετά την 28η εποχή και για τα δύο σετ δεδομένων.

Figure 53: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται συνεχής μείωση και σταθεροποίηση στην 22η και στην 24η εποχή για τα δύο σετ δεδομένων.

Το τέταρτο εισαγωγικό πρόβλημα είχε ως στόχο τη δυαδική ταξινόμηση εικόνων που περιλαμβάνουν δύο απλά γεωμετρικά σχήματα, ένα τετράγωνο και έναν κύκλο μεταβλητών διαστάσεων ως προς την αλληλοεπικάλυψη αυτών. Κατασκευάστηκε το δείγμα συνολικού όγκου 3000 εικόνων με ένα τετράγωνο και έναν κύκλο μεταβλητών διαστάσεων εντός του αρχικού πλαισίου και με επιτρεπτή την αλληλοεπικάλυψη μεταξύ των γεωμετρικών σχημάτων και στη συνέχεια το νευρωνικό δίκτυο εκπαιδεύτηκε με αυτόν τον σκοπό. Γενικά, επιτεύχθη συνολική ακρίβεια 92% στην ορθή πρόβλεψη των ετικετών ως προς την αλληλοεπικάλυψη των γεωμετρικών σχημάτων και μπορεί να θεωρηθεί πως το μοντέλο αποδίδει ικανοποιητικά παρά την αυξημένη πολυπλοκότητα

των δεδομένων εισόδου λόγω της διαφορετικής φύσης των γεωμετρικών σχημάτων, των διαφορετικών διαστάσεων και της επιτρεπτής επικάλυψης.

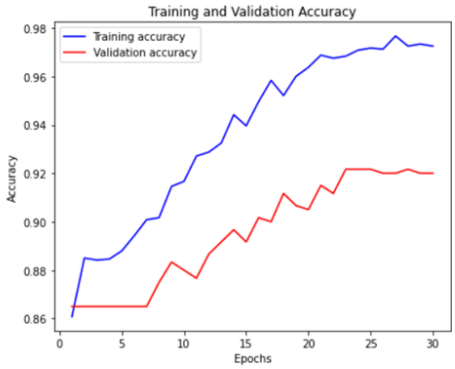


Figure 54: Η πορεία της μετρικής της ακρίβειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται ανοδική πορεία και σταθεροποίηση στην 28η εποχή και για τα δύο σετ δεδομένων.

Figure 55: Η πορεία της μετρικής της απώλειας κατά τη διάρκεια των 30 εποχών. Παρατηρείται συνεχής μείωση και σταθεροποίηση στην 24η εποχή και για τα δύο σετ δεδομένων.

Τα νευρωνικά δίκτυα αυτά επέτρεψαν τη μελέτη της διαδικασίας μάθησης ενός νευρωνικού δικτύου, μέσω της κατάλληλης προσαρμογής των παραμέτρων παρατηρώντας την εξέλιξη των αντίστοιχων μετρικών. Σε όλα τα εισαγωγικά νευρωνικά δίκτυα που κατασκευάστηκαν παρατηρείται αύξηση της μετρικής της ακρίβειας και σταθεροποίηση προς τη μέγιστη τιμή με τη μέγιστη αυτή τιμή να είναι μικρότερη για τα προβλήματα μεγαλύτερης πολυπλοκότητας. Παράλληλα, η μετρική της απώλειας παρουσιάζει εκθετική μείωση στις πρώτες εποχές και μείωση με μικρότερο ρυθμό στη συνέχεια όσο πλησιάζει ένα σημείο σταθεροποίησης που τείνει στο 0. Ως εκ τούτου, και στα τέσσερα εισαγωγικά προβλήματα ταξινόμησης εικόνων η διαδικασία της εκπαίδευσης μπορεί να θεωρηθεί επιτυχής και ότι το ζητούμενο κάθε προβλήματος έχει επιτευχθεί.

Καθώς η παρούσα εργασία εστιάζει στην προσέγγιση της ιατρικής φυσικής με τη χρήση νευρωνικών δικτύων επιλέχθηκε η διάκριση της νόσου Alzheimer μέσω της εξέτασης DaTSCAN για περαιτέρω μελέτη. Για τις ανάγκες της εκπαίδευσης του συνελικτικού νευρωνικού δικτύου, χρησιμοποιήθηκε το πρόγραμμα Simulix με δυνατότητα να παράγει ομοιώματα εγκεφαλικών τομογραφιών με ανομοιογενή απορρόφηση του ραδιοφαρμάκου ^{123}I. Κατασκευάστηκαν 3000 ομοιώματα εγκεφαλικών τομογραφιών διαστάσεων 128×128 τα οποία παρουσίαζαν την κεφαλή και τους εγκεφαλικούς λοβούς με ελλειψοειδή σχήματα τα οποία παρουσιάζουν διαφοροποιήσεις ως προς το μέγεθος, τη θέση και την απορρόφηση, μέσω στοχαστικών, τυχαίων παραλλαγών. Έπειτα, το δίκτυο εκπαιδεύτηκε για την πρόβλεψη της ανομοιογενούς απορρόφησης του ραδιοφαρμάκου.

Επίσης, κατασκευάστηκε άλλο ένα νευρωνικό δίκτυο το οποίο προβλέπει τις συνεχείς τιμές του υποβάθρου και της απορροφούμενης έντασης κάθε εγκεφαλικού λοβού ξεχωριστά. Ως μετρική των δύο συνελικτικών νευρωνικών δικτύων επιλέχθηκε να είναι η μέση τετραγωνική απόκλιση (RMSE). Στο δεύτερο CNN επιλέχθηκαν 50 εποχές εκπαίδευσης σε αντίθεση με το πρώτο που επιλέχθηκαν μόνο 30 εποχές. Η επιλογή αυτή έγινε λόγω της αύξησης των ζητούμενων μεγεθών προς πρόβλεψη και ενισχύθηκε από την παρατήρηση πως σε περίπτωση ελλιπούς εκπαίδευσης του πρώτου νευρωνικού δικτύου, το μοντέλο παρουσίασε συστηματική αβεβαιότητα στις προβλέψεις των ακραίων τιμών.

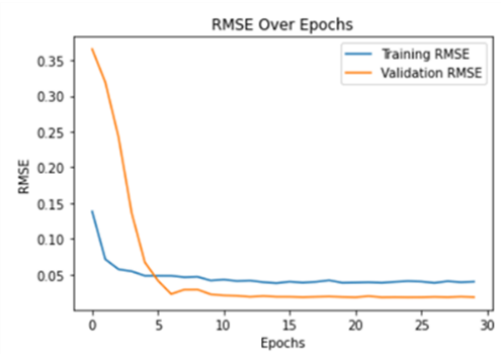


Figure 56: Η πορεία του RMSE κατά τις 30 epochs εκπαίδευσης στο CNN ποσοτικοποίησης της ανομοιογενούς απορρόφησης του ^{123}I. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή μετά την 10η εποχή και στα δύο σετ δεδομένων.

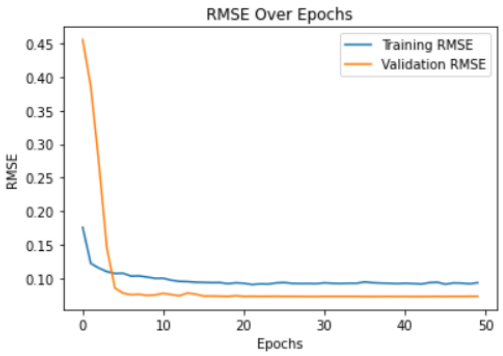


Figure 57: Η πορεία του RMSE κατά τις 50 epochs εκπαίδευσης στο CNN πρόβλεψης της απορροφούμενης έντασης του ^{123}I κάθε λοβού και του υποβάθρου. Παρατηρείται σταθεροποίηση σε ελάχιστη τιμή στην 20η εποχή για το σετ εκπαίδευσης και στην 14η εποχή στο σετ επαλήθευσης.

Και στα δύο συνελικτικά νευρωνικά δίκτυα παρατηρήθηκε μείωση της μέσης τετραγωνικής απόκλισης και σταθεροποίηση σε ελάχιστη τιμή κοντά στο 0 με το πέρας των εποχών, καθιστώντας την εκπαίδευση του μοντέλου επιτυχή και τα δύο νευρωνικά δίκτυα ικανά να γενικεύουν σε άγνωστες περιπτώσεις και να τις αξιολογούν.

Τέλος, χορηγήθηκαν δύο ιατρικές εικόνες εξέτασης DaTSCAN από τη γ – Camera MIRAGE DS7 σε μορφή πινάκων οι οποίες τροφοδοτήθηκαν στα δύο νευρωνικά δίκτυα με σκοπό την εξαγωγή προβλέψεων για τις αντίστοιχες ετικέτες και ως εκ τούτου την αξιολόγηση του νευρωνικού δικτύου. Προκειμένου να συγκριθούν οι προβλέψεις των νευρωνικών δικτύων με τις πραγματικές τιμές, από τις εικόνες τις γ – Camera απομονώθηκαν ως περιοχές ενδιαφέροντος το hotspot κάθε λοβού και μία περιοχή του υποβάθρου από όπου εξάχθηκαν οι μέσες τιμές των υποπινάκων για σύγκριση με τις προβλεπόμενες τιμές.

Αναφορικά με την απορροφούμενη ένταση του υποβάθρου, παρατηρήθηκε ποσοστιαία απόκλιση της τάξης του 27% η οποία οφείλεται στη δυσκολία που αντιμετωπίζει το νευρωνικό δίκτυο να ανιχνεύσει τις μικρές διαφοροποιήσεις των πραγματικών και των προβλεπόμενων τιμών του υποβάθρου. Οι παρατηρούμενες ποσοστιαίες διαφορές κυμαίνονταν από 2.95% έως 5.42% για τις προβλεπόμενες απορροφούμενες εντάσεις των δύο εγκεφαλικών λοβών, οι οποίες θεωρούνται αποδεκτές καθώς οι πραγματικές ιατρικές εικόνες δεν ανταποκρίνονται απόλυτα στα θεωρητικά ομοιώματα αλλά παρουσιάζουν μικρές διαφοροποιήσεις στη φυσιολογία και τη μορφολογία τους. Ως εκ τούτου, το CNN καθίσταται ικανό να λειτουργήσει επικουρικά στην διάγνωση από τον αντίστοιχο ειδικό θεράποντα ιατρό.

Βιβλιογραφία

[Abu 23] Abut, Serdar & Okut, Hayrettin & Kallail, K.. (2023), “Paradigm Shift from Artificial Neural Networks (ANNs) to Deep Convolutional Neural Networks (DCNNs) in the Field of Medical Image Processing”, Expert Systems with Applications, 244, (σελ. 2,7,12).

[Blo 96] Matthew Bloom, Stefanie Jacobs, John Pile-Spellman, Theodore A. Pozniakoff, Marissa I. Mabutas, Rashid A. Fawwaz and Ronald L. Van Heertum (1996), “Cerebral SPECT Imaging: Effect on Clinical Management”, Journal of nuclear medicine : official publication, Society of Nuclear Medicine vol. 37,7 (1996), (σελ. 1070).

[Cas 24] Juan A. Castro-Silva, María N. Moreno-García, Lorena Guachi Guachi, Diego H. Peluffo-Ordóñez (2024), “Novel hippocampus-centered methodology for informative instance selection in Alzheimer's disease data”, Heliyon, Volume 10, Issue 19, (σελ. 1,2).

[Chi 04] Chisté, V., & Bé, M. M. (2004), Table de radionucléides: I-123. BNM – LNHB/CEA, (σελ.1).

[GE 20] GE Healthcare Limited, (2020). DaTSCAN 74 MBq/ml ενέσιμο διάλυμα – Περίληψη των χαρακτηριστικών του προϊόντος, European Medicines Agency.

[Goo 16] Goodfellow, I., Bengio, Y., & Courville, A. (2016), “Deep learning”, ISBN 978-0262035613, MIT Press, (σελ. 305, 308-9).

[Kin 15] Kingma, D. P., & Ba, J. (2015), “Adam: A method for stochastic optimization”, In International Conference on Learning Representations (ICLR), (σελ. 1-3).

[Mat 15] Mattsson, S., Johansson, L., Leide Svegborn, S., Liniecki, J., Noßke, D., Riklund, K. Å., Stabin, M., Taylor, D., Bolch, W., Carlsson, S., Eckerman, K., Giussani, A., Söderberg, L., & Valind, S. (2015), “Radiation dose to patients from radiopharmaceuticals: A compendium of current information related to frequently used substances”, ICRP Publication 128, Annals of the ICRP, 44(2S), (σελ. 242-3).

[Med 11] Medeiros, R., Baglietto-Vargas, D., & LaFerla, F. M. (2011), “The role of tau in Alzheimer’s disease and related disorders”, CNS Neuroscience & Therapeutics, 17(6), (σελ. 514).

[Mey 14] Meyer, Philipp T, and Sabine Hellwig, “Update on SPECT and PET in parkinsonism - part 1: imaging for differential diagnosis”, Current opinion in neurology vol. 27,4 (2014): 390-7, (σελ. 390).

[Oro 23] Orobets, K.S., Karamyshev, A.L. (2023 – 24), “Amyloid Precursor Protein and Alzheimer’s Disease”, International Journal of Molecular Science, 14794, (σελ. 1 -3)

[O’ Sh 15] Keiron O’Shea, Ryan Nash,(2015), “An Introduction to Convolutional Neural Networks” , arXiv, (σελ. 1,3)

[O' Sh 24] O'Shea, Deirdre M et al, "Practical use of DAT SPECT imaging in diagnosing dementia with Lewy bodies: a US perspective of current guidelines and future directions.", Frontiers in neurology vol. 15 1395413, 22 Apr. 2024, (σελ. 1-3,14).

[Vil 05] Victor L. Villemagne, C.C. Rowe, S. Macfarlane, Novakovic, C.L. Masters (2005), "Imaginem oblivionis: the prospects of neuroimaging for early detection of Alzheimer's disease", Journal of Clinical Neuroscience 12(3), (σελ. 221, 223).

Ηλεκτρονικές πηγές

[CIN 20] The Cyprus Institute of Neurology & Genetics (CING), Neuropathology Department (2020), "ΕΡΕΥΝΑ, Στη μάχη κατά της Νόσου Αλτσχάιμερ'' :

https://www.cing.ac.cy/en/about-us/clinical-sciences/nca/newsletter-articles/feb20-art06

[EMA] European Medicines Agency, (n.d.), DaTSCAN (Ioflupane (123I)) [Product information]: http://www.ema.europa.eu

[Kec nd] Keck School of Medicine of USC (Alzheimer's Therapeutic Research Institute): https://atrinews.usc.edu/how-alzheimers-disease-changes-the-brain

[NNDC] National Nuclear Data Center, MIRD data, Brookhaven National Laboratory: https://www.nndc.bnl.gov/nudat3/mird/

[Phy 24] Brain Anatomy, (2024, July 4), Physiopedia: https://www.physiopedia.com/Brain_Anatomy

[Rel 25] ReLU Activation Function in Deep Learning (2025, January 29) GeeksforGeeks: https://www.geeksforgeeks.org/relu-activation-function-in-deep-learning/

Παράρτημα

Παρακάτω παρουσιάζονται οι αλγόριθμοι που κατασκευάστηκαν στα πλαίσια της παρούσας πτυχιακής εργασίας σε γλώσσα προγραμματισμού Python.

1ο Πρόβλημα Ταξινόμησης

```
import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
import random
import matplotlib.patches as patches
from PIL import Image
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Dropout
from tensorflow.keras.utils import to_categorical

# Functions to create square and circle images
def create_square(size=128):
    array = np.zeros((size, size))
    bottom_edge = int(np.random.uniform(0, 1) * 100)
    left_edge = int(np.random.uniform(0, 1) * 100)
    square_size = 14 + int(np.random.uniform(0, 1) * 14)
    top_edge = bottom_edge + square_size
    right_edge = left_edge + square_size
    array[bottom_edge:top_edge, left_edge:right_edge] = 1
    return array

def create_circle(size=128):
    array = np.zeros((size, size))
    max_radius = 25
    min_radius = 5
    radius = np.random.randint(min_radius, max_radius)
    center_x = np.random.randint(radius, size - radius)
    center_y = np.random.randint(radius, size - radius)
    y, x = np.ogrid[:size, :size]
    mask = (x - center_x)**2 + (y - center_y)**2 <= radius**2
    array[mask] = 1
    return array

# Paths for storing data
dataset_path_circles = 'dataset/just_circles'
dataset_path_squares = 'dataset/just_squares'

# Clear and recreate directories for circles
if os.path.exists(dataset_path_circles):
    for filename in os.listdir(dataset_path_circles):
        file_path = os.path.join(dataset_path_circles, filename)
        os.remove(file_path)
    print("Previous images of circles cleared.")
if not os.path.exists(dataset_path_circles):
    os.makedirs(dataset_path_circles)
```

```
Previous images of circles cleared.
Previous images of squares cleared.
```

```
labels_path_circles = os.path.join(dataset_path_circles,
'labels_circles.npy')
if os.path.exists(labels_path_circles):
    os.remove(labels_path_circles)

# Clear and recreate directories for squares
if os.path.exists(dataset_path_squares):
    for filename in os.listdir(dataset_path_squares):
        file_path = os.path.join(dataset_path_squares, filename)
        os.remove(file_path)
    print("Previous images of squares cleared.")
if not os.path.exists(dataset_path_squares):
    os.makedirs(dataset_path_squares)

labels_path_squares = os.path.join(dataset_path_squares,
'labels_squares.npy')
if os.path.exists(labels_path_squares):
    os.remove(labels_path_squares)

# Generate images
def save_images():
    images = []
    labels = []

    # Generate squares
    for i in range(1500):
        img = create_square()
        images.append(img)
        labels.append(1)  # Label for square

    # Generate circles
    for i in range(1500):
        img = create_circle()
        images.append(img)
        labels.append(0)  # Label for circle

    images = np.array(images).reshape(-1, 128, 128, 1)
    labels = np.array(labels)
    return images, labels

images, labels = save_images()

# Split dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split(images, labels,
test_size=0.3, random_state=42)


# Define the model architecture with binary crossentropy
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 1)),
    MaxPooling2D((2, 2)),
    Dropout(0.25),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
```

```
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')  # Single output with sigmoid
activation for binary classification
])

# Compile the model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, epochs=20, batch_size=32,
validation_split=0.2)
```

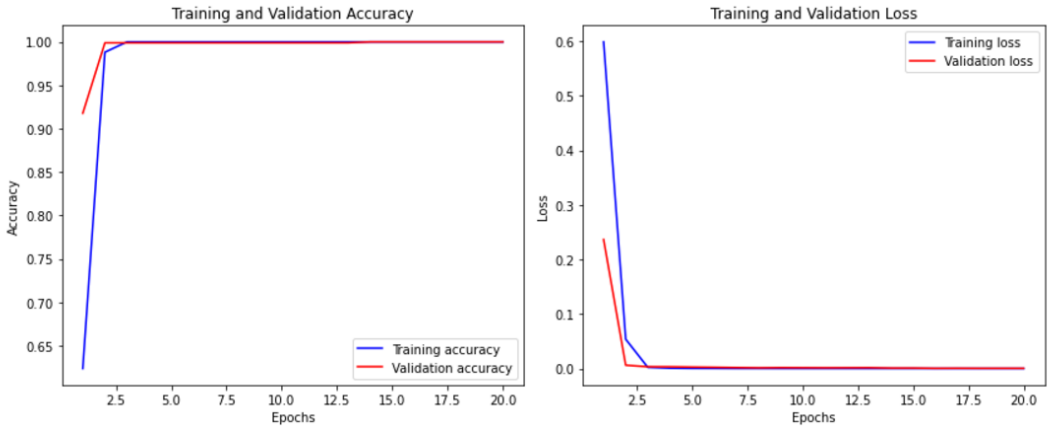
```
Epoch 1/20
66/66 [==============================] - 36s 529ms/step - loss: 0.5987 - accuracy: 0.6238 - val_loss: 0.2365 - val_accuracy: 0.
9178
Epoch 2/20
66/66 [==============================] - 31s 471ms/step - loss: 0.0538 - accuracy: 0.9881 - val_loss: 0.0061 - val_accuracy: 0.
9989
```

....

```
Epoch 19/20
66/66 [==============================] - 34s 508ms/step - loss: 1.7818e-05 - accuracy: 1.0000 - val_loss: 4.7914e-04 - val_accu
racy: 1.0000
Epoch 20/20
66/66 [==============================] - 33s 499ms/step - loss: 2.3233e-05 - accuracy: 1.0000 - val_loss: 4.6569e-04 - val_accu
racy: 1.0000
```

```
# Plot the training and validation accuracy and loss
def plot_history(history):
    # Extract accuracy and loss data
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs = range(1, len(acc) + 1)

    # Plot training and validation accuracy
    plt.figure(figsize=(12, 5))

    # Subplot for accuracy
    plt.subplot(1, 2, 1)
    plt.plot(epochs, acc, 'b', label='Training accuracy')
    plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
    plt.title('Training and Validation Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()
    # Subplot for loss
    plt.subplot(1, 2, 2)
    plt.plot(epochs, loss, 'b', label='Training loss')
    plt.plot(epochs, val_loss, 'r', label='Validation loss')
    plt.title('Training and Validation Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()
```

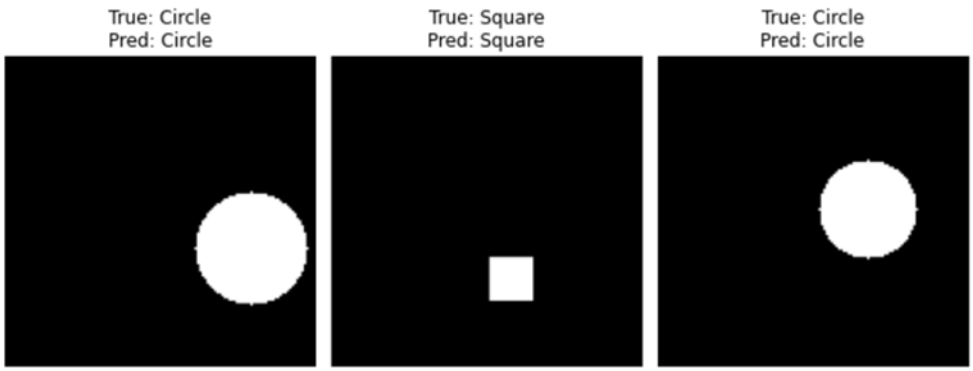
```
    plt.tight_layout()
    plt.show()

# Assuming `history` contains the model history, call the plot
function
plot_history(history)
```




```
# Evaluate the model
model.evaluate(X_test, y_test)
```

```
29/29 [==============================] - 2s 78ms/step - loss: 4.6569e-04 - accuracy: 1.0000
```

```
num_samples_to_display = 10
validation_images_sample = X_test[:num_samples_to_display]
validation_labels_sample = y_test[:num_samples_to_display]

# Πρόβλεψη ετικετών
predictions = model.predict(validation_images_sample)
predicted_labels = (predictions > 0.5).astype(int).flatten()
plt.figure(figsize=(15, 10))
for i in range(num_samples_to_display):
    plt.subplot(2, 5, i + 1)
    plt.imshow(validation_images_sample[i].reshape(128, 128),
cmap='gray')
    true_label = "Circle" if validation_labels_sample[i] == 0 else
"Square"
    predicted_label = "Circle" if predicted_labels[i] == 0 else
"Square"
    plt.title(f"True: {true_label}\nPred: {predicted_label}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```



2ο Πρόβλημα Ταξινόμησης

Α) Αλγόριθμος χωρίς επικάλυψη μεταξύ των γεωμετρικών σχημάτων

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import os
from tensorflow.keras import layers, models
import math
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D,
Flatten, Dense, Dropout, BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.regularizers import l2
from tensorflow.keras import regularizers
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.callbacks import ReduceLROnPlateau

# Clear previous dataset if it exists
dataset_path = 'dataset/square_v2'
if os.path.exists(dataset_path):
    for filename in os.listdir(dataset_path):
        file_path = os.path.join(dataset_path, filename)
        os.remove(file_path)
    print("Previous images cleared.")
```

```
Previous images cleared.
```

```
# Δημιουργία φακέλου αν δεν υπάρχει
if not os.path.exists(dataset_path):
    os.makedirs(dataset_path)

labels_path = os.path.join(dataset_path, 'labels.npy')
if os.path.exists(labels_path):
    os.remove(labels_path)
    print("Previous labels cleared.")

def check_overlap(centers, new_center, square_size):
    xi, yi = new_center
    for (center_xi, center_yi) in centers:
        # Υπολογισμός απόστασης μεταξύ του νέου κέντρου και του
υπάρχοντος
        d_square = ((center_xi - xi) ** 2) + ((center_yi - yi) ** 2)
        d = math.sqrt(d_square)

        if center_xi == xi:
            a = np.pi / 2
        else:
            a = (center_yi - yi) / (center_xi - xi)
        tanf = math.tan(a)
        f = math.atan(tanf)
        cosf = math.cos(f)
        if d < (square_size / cosf):
```

```
            return True


def create_squares(size=128, square_size=30, num_squares=None):
    array = np.zeros((size, size))

    # If num_squares is None, randomly choose between 1 and 4
squares.
    if num_squares is None:
        num_squares = np.random.randint(1, 5)
    centers = []
    for _ in range(num_squares):
        while True:
            bottom_edge = int(np.random.uniform(0, size -
square_size))
            left_edge = int(np.random.uniform(0, size - square_size))
            top_edge = bottom_edge + square_size
            right_edge = left_edge + square_size

            # Calculate the center of the square
            xi = left_edge + square_size // 2
            yi = bottom_edge + square_size // 2
            current_center = (xi, yi)

            if not check_overlap(centers, current_center,
square_size):
                centers.append(current_center)
                array[bottom_edge:top_edge, left_edge:right_edge] = 1
                break
    return array, len(centers)

num_samples = 3000
square_size = 30   # Size of each square
labels = []        # List to store labels for each image
padding_length = len(str(num_samples))
for img_idx in range(num_samples):
    # Randomly choose the number of squares for this image
    num_squares = np.random.randint(1, 5)
    array, _ = create_squares(size=128, square_size=square_size,
num_squares=num_squares)

    # Save the generated image with zero-padded filename
    image_filename = f'dataset/square_v2/square_{str(img_idx +
1).zfill(padding_length)}.png'
    plt.imsave(image_filename, array, cmap='gray', vmin=0, vmax=1)

    # Save the label immediately after creating the image
    labels.append(num_squares)
# Convert labels to numpy array
labels = np.array(labels)
# Save all labels as a numpy array
labels_path = 'dataset/square_v2/labels.npy'
np.save(labels_path, labels)

print(f"Labels saved successfully to: {labels_path}")
print("Images and labels saved successfully.")
```

```
Labels saved successfully to: dataset/square_v2/labels.npy
Images and labels saved successfully.
```

```
# Φόρτωση και κανονικοποίηση των εικόνων
def load_and_preprocess_images(directory):
    images = []
    for filename in os.listdir(directory):
        if filename.endswith(".png"):
            img = Image.open(os.path.join(directory,
filename)).convert('L')
            img = img.resize((128, 128))
            img = np.array(img) / 255.0
            img = np.expand_dims(img, axis=-1)
            images.append(img)
    return np.array(images)

# Φόρτωση δεδομένων
X = load_and_preprocess_images('dataset/square_v2')
y = np.load('dataset/square_v2/labels.npy')

# Διαχωρισμός
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# One-hot encoding των ετικετών
n_categories = 4  # Αριθμός κατηγοριών
y_train_adjusted = y_train - 1
y_test_adjusted = y_test - 1
y_train_ohe = to_categorical(y_train_adjusted,
num_classes=n_categories)
y_test_ohe = to_categorical(y_test_adjusted,
num_classes=n_categories)

#model
initial_lr = 1e-5
optimizer = Adam(learning_rate=initial_lr)

# μοντέλο
model = models.Sequential()
model.add(Input(shape=(128, 128, 1)))
model.add(layers.Conv2D(16, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(layers.Conv2D(32, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
# Fully Connected Layers
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(n_categories, activation='softmax'))
model.compile(optimizer=optimizer, loss='categorical_crossentropy',
metrics=['accuracy'])
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
```

```
    factor=0.3,
    patience=2,
    min_lr=1e-7,
    verbose=1
)

# Print the shape of the training data
print("Shape of train_data (X_train):", X_train.shape)
print("Shape of train_labels (y_train):", y_train.shape)
```

```
Shape of train_data (X_train): (2250, 128, 128, 1)
Shape of train_labels (y_train): (2250,)
```

```
# Train the model
history = model.fit(X_train, y_train_ohe,
                    validation_data=(X_test, y_test_ohe),
                    epochs=20,
                    batch_size=16,
                    verbose=1,
                    callbacks=[reduce_lr])
val_accuracy_non_overlap = history.history['val_accuracy']
np.save('val_accuracy_non_overlap.npy', val_accuracy_non_overlap)

train_loss = history.history['loss']
val_loss = history.history['val_loss']
train_accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
std_deviation = np.std(val_accuracy)

# number of epochs
epochs = range(1, len(train_loss) + 1)
```

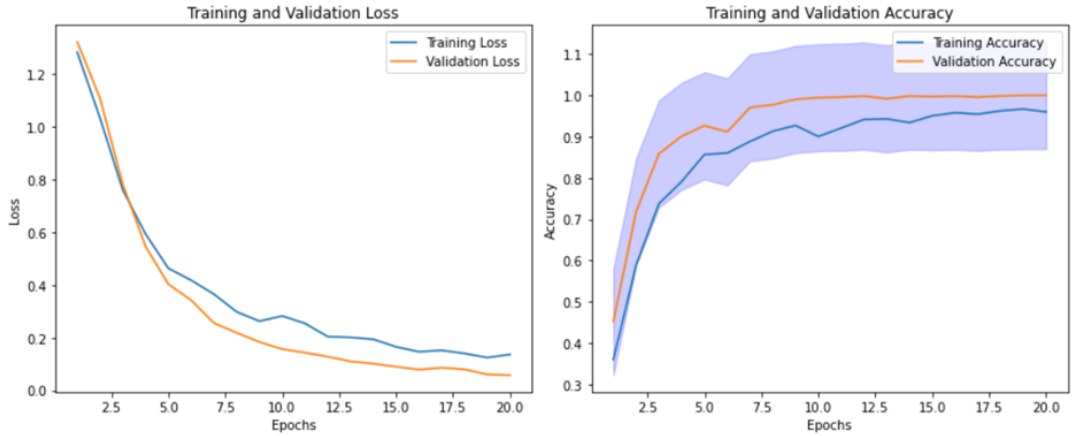
```
Epoch 1/20
141/141 [==============================] - 54s 372ms/step - loss: 1.2824 - accuracy: 0.3604 - val_loss: 1.3212 - val_accuracy:
0.4520 - lr: 1.0000e-05
Epoch 2/20
141/141 [==============================] - 53s 375ms/step - loss: 1.0327 - accuracy: 0.5889 - val_loss: 1.1074 - val_accuracy:
0.7187 - lr: 1.0000e-05
```

....

```
Epoch 19/20
141/141 [==============================] - 54s 383ms/step - loss: 0.1245 - accuracy: 0.9671 - val_loss: 0.0606 - val_accuracy:
1.0000 - lr: 3.0000e-06
Epoch 20/20
141/141 [==============================] - 54s 384ms/step - loss: 0.1363 - accuracy: 0.9600 - val_loss: 0.0578 - val_accuracy:
1.0000 - lr: 3.0000e-06
```

```
# Plot training and validation loss
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(epochs, train_loss, label='Training Loss')
plt.plot(epochs, val_loss, label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, train_accuracy, label='Training Accuracy')
plt.plot(epochs, val_accuracy, label='Validation Accuracy')
plt.fill_between(epochs,
                 np.array(val_accuracy) - std_deviation,
```

```
                    np.array(val_accuracy) + std_deviation,
                    color='b', alpha=0.2)
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
# Display the plots
plt.tight_layout()
plt.show()
```




```
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test_ohe)
print(f"Test accuracy: {accuracy:.2f}")
```

```
24/24 [==============================] - 3s 137ms/step - loss: 0.0578 - accuracy: 1.0000
Test accuracy: 1.00
```

Β) Αλγόριθμος με επιτρεπτή την επικάλυψη μεταξύ των γεωμετρικών σχημάτων

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import os
from tensorflow.keras import layers, models
import math
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D,
Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.regularizers import l2
from tensorflow.keras import regularizers
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.callbacks import ReduceLROnPlateau

# Clear previous dataset if it exists
dataset_path = 'dataset/square'
if os.path.exists(dataset_path):
    for filename in os.listdir(dataset_path):
        file_path = os.path.join(dataset_path, filename)
        os.remove(file_path)
    print("Previous images cleared.")
```

```
Previous images cleared.
```

```
# Δημιουργία φακέλου αν δεν υπάρχει
if not os.path.exists(dataset_path):
    os.makedirs(dataset_path)

labels_path = os.path.join(dataset_path, 'labels.npy')
if os.path.exists(labels_path):
    os.remove(labels_path)
    print("Previous labels cleared.")


def create_square(size=128, square_size=30, num_squares=None):
    array = np.zeros((size, size))
    if num_squares is None:
        num_squares = np.random.randint(1, 5)

    for _ in range(num_squares):
        # Random position for each square
        bottom_edge = int(np.random.uniform(0, size - square_size))
        left_edge = int(np.random.uniform(0, size - square_size))
        top_edge = bottom_edge + square_size
        right_edge = left_edge + square_size
        array[bottom_edge:top_edge, left_edge:right_edge] = 1
```

```
    return array, num_squares

# Συνολικός αριθμός εικόνων και μέγεθος τετραγώνου
num_samples = 3000
square_size = 30
labels = []
padding_length = len(str(num_samples))

# Δημιουργία εικόνων και ετικετών
for img_idx in range(num_samples):
    num_squares = np.random.randint(1, 5)
    array, _ = create_square(size=128, square_size=square_size,
num_squares=num_squares)

    # Αποθήκευση εικόνας
    image_filename = f'dataset/square/square_{str(img_idx +
1).zfill(padding_length)}.png'
    plt.imsave(image_filename, array, cmap='gray', vmin=0, vmax=1)

    labels.append(num_squares)

# Αποθήκευση ετικετών
labels = np.array(labels)
labels_path = 'dataset/square/labels.npy'
np.save(labels_path, labels)
print("Images and labels saved successfully.")
```

```
Images and labels saved successfully.
```

```
# Φόρτωση και κανονικοποίηση των εικόνων
def load_and_preprocess_images(directory):
    images = []
    for filename in os.listdir(directory):
        if filename.endswith(".png"):
            img = Image.open(os.path.join(directory,
filename)).convert('L')
            img = img.resize((128, 128))
            img = np.array(img) / 255.0
            img = np.expand_dims(img, axis=-1)
            images.append(img)
    return np.array(images)

# Φόρτωση δεδομένων
X = load_and_preprocess_images('dataset/square')
y = np.load('dataset/square/labels.npy')

# Διαχωρισμός δεδομένων
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=42)

# One-hot encoding των ετικετών
n_categories = 4
y_train_adjusted = y_train - 1
y_test_adjusted = y_test - 1
y_train_ohe = to_categorical(y_train_adjusted,
num_classes=n_categories)
```

```
y_test_ohe = to_categorical(y_test_adjusted,
num_classes=n_categories)

#model
initial_lr = 1e-5
optimizer = Adam(learning_rate=initial_lr)

# Ορισμός του μοντέλου
model = models.Sequential()
model.add(Input(shape=(128, 128, 1)))
model.add(layers.Conv2D(16, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(layers.Conv2D(32, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(n_categories, activation='softmax'))
model.compile(optimizer=optimizer, loss='categorical_crossentropy',
metrics=['accuracy'])

reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.3,
    patience=2,
    min_lr=1e-7,
    verbose=1
)
print("Shape of train_data (X_train):", X_train.shape)
print("Shape of train_labels (y_train):", y_train.shape)
```

```
Shape of train_data (X_train): (2250, 128, 128, 1)
Shape of train_labels (y_train): (2250,)
```

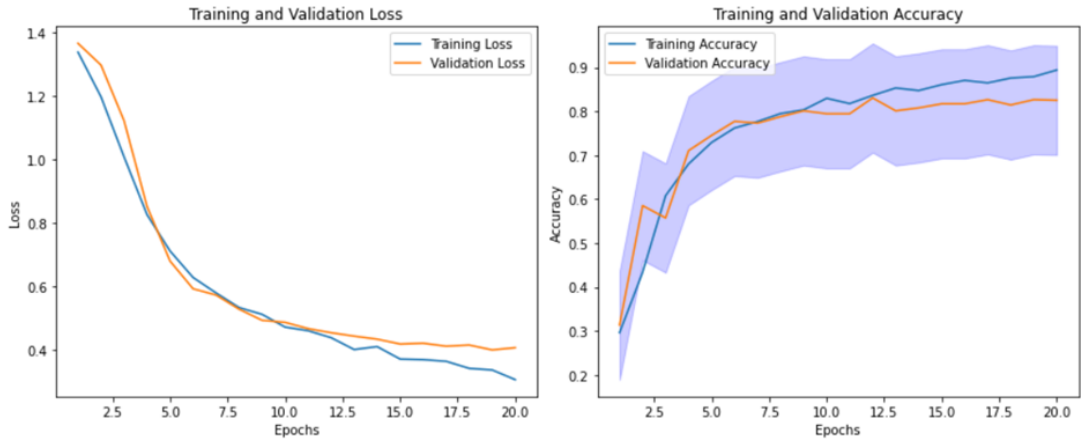
```
# Train the model
history = model.fit(X_train, y_train_ohe,
                    validation_data=(X_test, y_test_ohe),
                    epochs=20,
                    batch_size=16,
                    verbose=1,
                    callbacks=[reduce_lr])
val_accuracy_overlap = history.history['val_accuracy']
np.save('val_accuracy_overlap.npy', val_accuracy_overlap)

train_loss = history.history['loss']
val_loss = history.history['val_loss']
train_accuracy = history.history['accuracy']
val_accuracy = history.history['val_accuracy']
std_deviation = np.std(val_accuracy)

epochs = range(1, len(train_loss) + 1)
```

```
Epoch 1/20
141/141 [==============================] - 136s 929ms/step - loss: 1.3388 - accuracy: 0.2960 - val_loss: 1.3666 - val_accuracy:
0.3133 - lr: 1.0000e-05
Epoch 2/20
141/141 [==============================] - 205s 1s/step - loss: 1.1981 - accuracy: 0.4333 - val_loss: 1.2981 - val_accuracy: 0.
5853 - lr: 1.0000e-05
```

....

```
Epoch 19/20
141/141 [==============================] - 106s 756ms/step - loss: 0.3378 - accuracy: 0.8791 - val_loss: 0.4010 - val_accuracy:
0.8267 - lr: 1.0000e-05
Epoch 20/20
141/141 [==============================] - 54s 383ms/step - loss: 0.3073 - accuracy: 0.8942 - val_loss: 0.4079 - val_accuracy:
0.8253 - lr: 1.0000e-05
```

```
# Plot training and validation loss
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(epochs, train_loss, label='Training Loss')
plt.plot(epochs, val_loss, label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(epochs, train_accuracy, label='Training Accuracy')
plt.plot(epochs, val_accuracy, label='Validation Accuracy')
plt.fill_between(epochs,
                 np.array(val_accuracy) - std_deviation,
                 np.array(val_accuracy) + std_deviation,
                 color='b', alpha=0.2)
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

# Display the plots
plt.tight_layout()
plt.show()
```




```
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test_ohe)
print(f"Test accuracy: {accuracy:.2f}")
```

```
24/24 [==============================] - 3s 125ms/step - loss: 0.4079 - accuracy: 0.8253
Test accuracy: 0.83
```

3ο Πρόβλημα Ταξινόμησης

```
import os
import numpy as np
import matplotlib.pyplot as plt
import math
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Dropout, BatchNormalization, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
from PIL import Image
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.layers import Input

# Clear previous dataset if it exists
dataset_path = 'dataset/circle'
if os.path.exists(dataset_path):
    for filename in os.listdir(dataset_path):
        file_path = os.path.join(dataset_path, filename)
        os.remove(file_path)
    print("Previous images cleared.")
```

```
Previous images cleared.
```

```
if not os.path.exists(dataset_path):
    os.makedirs(dataset_path)

labels_path = os.path.join(dataset_path, 'labels.npy')
if os.path.exists(labels_path):
    os.remove(labels_path)

# Συνάρτηση δημιουργίας εικόνας με επικαλυπτόμενα σχήματα
def create_both(size=128):
    array = np.zeros((size, size))
    max_radius = 25
    min_radius = 5
    radius1 = np.random.randint(min_radius, max_radius)
    radius2 = np.random.randint(min_radius, max_radius)

    center_x1 = np.random.randint(radius1, size-radius1)
    center_y1 = np.random.randint(radius1, size-radius1)
    center_x2 = np.random.randint(radius2, size-radius2)
    center_y2 = np.random.randint(radius2, size-radius2)

    d_square = (((center_x2-center_x1)**2) + (center_y2-
center_y1)**2)
    d = math.sqrt(d_square)

    if d <= (radius1 + radius2):
        overlap = 1
    else:
        overlap = 0
```

```
    y, x = np.ogrid[:size, :size]
    mask1 = (x - center_x1)**2 + (y - center_y1)**2 <= radius1**2
    array[mask1] = 1
    mask2 = (x - center_x2)**2 + (y - center_y2)**2 <= radius2**2
    array[mask2] = 1
    return array, overlap

# Δημιουργία εικόνων και ετικετών
num_samples = 3000
labels = []  # Λίστα για τις ετικέτες
padding_length = len(str(num_samples))

for img_idx in range(num_samples):
    image, overlap = create_both()
    image_filename = f'dataset/circle/circle_{str(img_idx +
1).zfill(padding_length)}.png'
    plt.imsave(image_filename, image, cmap='gray', vmin=0, vmax=1)
    labels.append(overlap)
labels = np.array(labels)
labels_path = 'dataset/circle/labels.npy'
# Αποθήκευση των ετικετών ως numpy array
np.save(labels_path, np.array(labels))
print("Dataset and labels saved successfully.")
```

```
Dataset and labels saved successfully.
```

```
# Φόρτωση των εικόνων και των ετικετών
def load_and_preprocess_images(directory):
    images = []
    for filename in os.listdir(directory):
        if filename.endswith(".png"):
            img = Image.open(os.path.join(directory,
filename)).convert('L')
            img = img.resize((128, 128))
            img = np.array(img) / 255.0
            img = np.expand_dims(img, axis=-1)
            images.append(img)
    return np.array(images)

# Φόρτωση εικόνων και ετικετών
circle_images = load_and_preprocess_images(dataset_path)
circle_labels = np.load(labels_path)
# Διαχωρισμός σε training και testing datasets
X_train, X_test, y_train, y_test = train_test_split(circle_images,
circle_labels, test_size=0.2, random_state=42)

print(f"Loaded {circle_images.shape[0]} images with shape
{circle_images.shape[1:]}")
print(f"Labels shape: {circle_labels.shape}")
```

```
Loaded 3000 images with shape (128, 128, 1)
Labels shape: (3000,)
```

```
initial_lr = 4e-5
optimizer = Adam(learning_rate=initial_lr)
# Δημιουργία μοντέλου
model = Sequential([
    Input((128, 128, 1)),
    Conv2D(32, (3, 3), activation='relu'),
    BatchNormalization(),
```

```
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')
])
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.3,
    patience=2,
    min_lr=1e-6,
    verbose=1
)

model.compile(optimizer=optimizer,
              loss='binary_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train, y_train,
                    validation_data=(X_test, y_test),
                    epochs=30,
                    batch_size=16,
                    verbose=1,
                    callbacks=[reduce_lr])
```

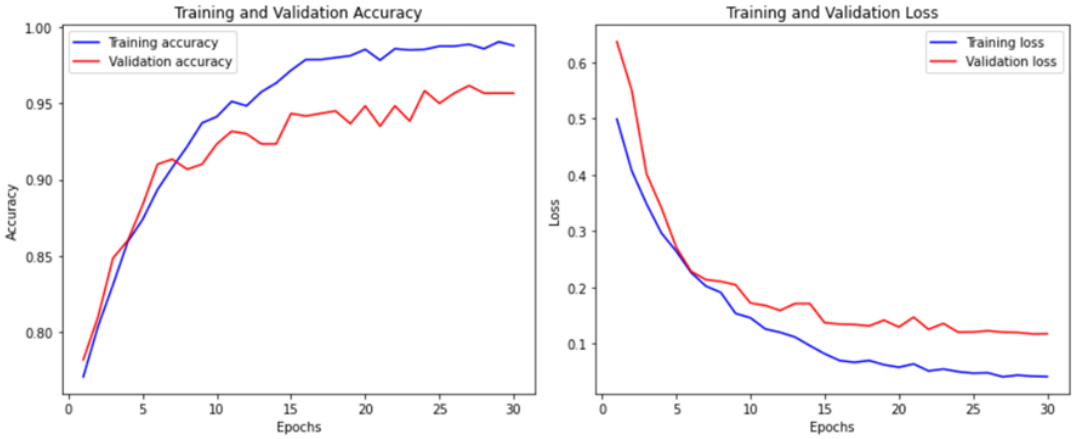
```
Epoch 1/30
150/150 [==============================] - 42s 269ms/step - loss: 0.4988 - accuracy: 0.7704 - val_loss: 0.6361 - val_accuracy:
0.7817 - lr: 4.0000e-05
Epoch 2/30
150/150 [==============================] - 40s 270ms/step - loss: 0.4070 - accuracy: 0.8037 - val_loss: 0.5501 - val_accuracy:
0.8100 - lr: 4.0000e-05
```

...

```
Epoch 29/30
150/150 [==============================] - 43s 290ms/step - loss: 0.0422 - accuracy: 0.9904 - val_loss: 0.1173 - val_accuracy:
0.9567 - lr: 3.6000e-06
Epoch 30/30
150/150 [==============================] - 41s 274ms/step - loss: 0.0415 - accuracy: 0.9879 - val_loss: 0.1177 - val_accuracy:
0.9567 - lr: 3.6000e-06
```

```
def plot_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs = range(1, len(acc) + 1)
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(epochs, acc, 'b', label='Training accuracy')
    plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
    plt.title('Training and Validation Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()
    plt.subplot(1, 2, 2)
```

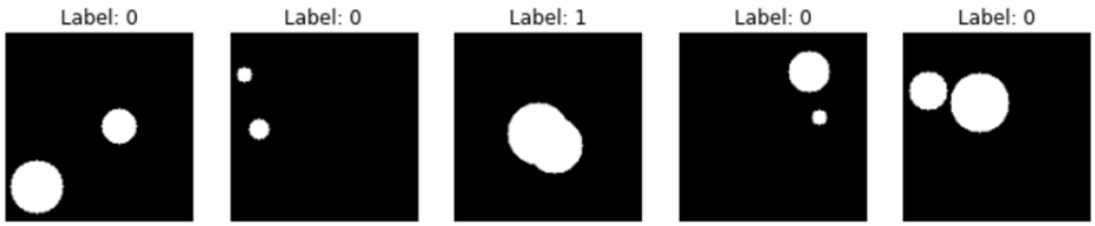
```
    plt.plot(epochs, loss, 'b', label='Training loss')
    plt.plot(epochs, val_loss, 'r', label='Validation loss')
    plt.title('Training and Validation Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()
    plt.tight_layout()
    plt.show()
plot_history(history)
```




```
# Αξιολόγηση του μοντέλου
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test accuracy: {accuracy:.2f}")
```

```
19/19 [==============================] - 2s 119ms/step - loss: 0.1177 - accuracy: 0.9567
Test accuracy: 0.96
```

```
# Προβολή εικόνων από το validation set μαζί με τις ετικέτες τους
def show_random_validation_images(X, y, num_images=5):
    idx = np.random.choice(len(X), num_images, replace=False)
    plt.figure(figsize=(12, 6))
    for i, idx in enumerate(idx):
        plt.subplot(1, num_images, i + 1)
        plt.imshow(X[idx].reshape(128, 128), cmap='gray')
        plt.title(f"Label: {y[idx]}")
        plt.axis('off')
    plt.show()
show_random_validation_images(X_test, y_test)
```



4ο Πρόβλημα Ταξινόμησης

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import os
import math
from PIL import Image
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D,
Flatten, Dense, Dropout, BatchNormalization, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam, SGD
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras import regularizers
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.layers import Input

num_samples = 3000
labels = []
mix = 'dataset/create_mix'

# Clear previous dataset if it exists
if os.path.exists(mix):
    for filename in os.listdir(mix):
        file_path = os.path.join(mix, filename)
        os.remove(file_path)
    print("Previous images cleared.")
if not os.path.exists(mix):
    os.makedirs(mix)

def create_mix(size=128):
    array = np.zeros((size, size))
    # Circle
    max_radius = 25
    min_radius = 5
    radius1 = np.random.randint(min_radius, max_radius)
    center_x1 = np.random.randint(radius1, size - radius1)
    center_y1 = np.random.randint(radius1, size - radius1)
    # Square
    bottom_edge = int(np.random.uniform(0, 1) * (size - 28))
    left_edge = int(np.random.uniform(0, 1) * (size - 28))
    square_size = 14 + int(np.random.uniform(0, 1) * 14)
    top_edge = bottom_edge + square_size
    right_edge = left_edge + square_size
    center_x2 = (left_edge + right_edge) / 2
    center_y2 = (bottom_edge + top_edge) / 2

    # Distance and overlap calculation
    if center_x2 == center_x1:
        f = np.pi / 2
    else:
```

```
        f = math.atan((center_y2 - center_y1) / (center_x2 -
center_x1))
    cosf = math.cos(f)
    b = (square_size * cosf) / 2

    # Distance between centers
    d_square = (center_x2 - center_x1) ** 2 + (center_y2 - center_y1)
** 2
    d = math.sqrt(d_square)
    if d <= (radius1 + b):
        overlap = 1
    else:
        overlap = 0

    y, x = np.ogrid[:size, :size]
    mask_circle = (x - center_x1) ** 2 + (y - center_y1) ** 2 <=
radius1 ** 2
    array[mask_circle] = 1
    array[bottom_edge:top_edge, left_edge:right_edge] = 1
    return array, overlap

padding_length = len(str(num_samples))
for img_idx in range(num_samples):
    random_both, overlap = create_mix()
    image_filename = f'{mix}/create_mix_{str(img_idx +
1).zfill(padding_length)}.png'

    plt.imsave(image_filename, random_both, cmap='gray', vmin=0,
vmax=1)

    # Append label
    labels.append(overlap)

# Save labels to a numpy file
labelsmix_path = 'dataset/create_mix/labels.npy'
if not os.path.exists(os.path.dirname(labelsmix_path)):
    os.makedirs(os.path.dirname(labelsmix_path))
np.save(labelsmix_path, np.array(labels))
print("Images and labels saved successfully.")
```

```
Previous images cleared.
Images and labels saved successfully.
```

```
def load_and_preprocess_images(directory):
    images = []
    for filename in os.listdir(directory):
        if filename.endswith(".png"):
            img = Image.open(os.path.join(directory,
filename)).convert('L')
            img = img.resize((128, 128))
            img = np.array(img) / 255.0
            img = np.expand_dims(img, axis=-1)
            images.append(img)
    return np.array(images)

# Load images
mix_images = load_and_preprocess_images('dataset/create_mix')
```

```
# Load labels
mix_labels = np.load('dataset/create_mix/labels.npy')

print(f"Number of images: {len(mix_images)}")
print(f"Number of labels: {len(mix_labels)}")
X_train, X_test, y_train, y_test = train_test_split(mix_images,
                                                    mix_labels,
                                                    test_size=0.2,
                                                    random_state=42)
print(f"Loaded {mix_images.shape[0]} images with shape
{mix_images.shape[1:]}")
print(f"Labels shape: {mix_labels.shape}")
```

```
Number of images: 3000
Number of labels: 3000
Loaded 3000 images with shape (128, 128, 1)
Labels shape: (3000,)
```

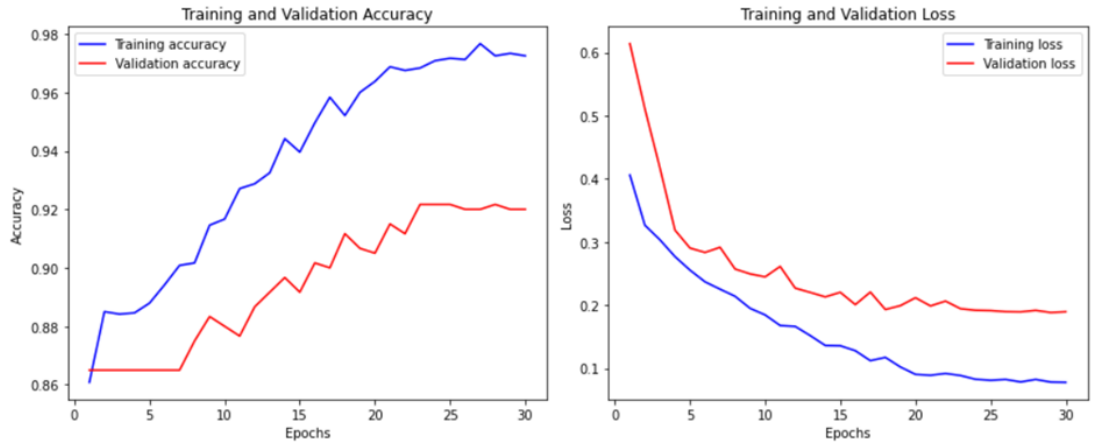
```
initial_lr = 2e-5
optimizer = Adam(learning_rate=initial_lr)
# Δημιουργία μοντέλου
model = Sequential([
    Input((128, 128, 1)),
    Conv2D(32, (3, 3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.3,
    patience=2,
    min_lr=1e-6,
    verbose=1
)

model.compile(optimizer=optimizer,
              loss='binary_crossentropy',
              metrics=['accuracy'])

history = model.fit(X_train, y_train,
                    validation_data=(X_test, y_test),
                    epochs=30,
                    batch_size=16,
                    verbose=1,
                    callbacks=[reduce_lr])
```

```
Epoch 1/30
150/150 [==============================] - 46s 301ms/step - loss: 0.4061 - accuracy: 0.8608 - val_loss: 0.6141 - val_accuracy:
0.8650 - lr: 2.0000e-05
Epoch 2/30
150/150 [==============================] - 42s 282ms/step - loss: 0.3265 - accuracy: 0.8850 - val_loss: 0.5109 - val_accuracy:
0.8650 - lr: 2.0000e-05
```

...

```
Epoch 29/30
150/150 [==============================] - 42s 280ms/step - loss: 0.0782 - accuracy: 0.9733 - val_loss: 0.1883 - val_accuracy:
0.9200 - lr: 1.8000e-06
Epoch 30/30
150/150 [==============================] - 41s 270ms/step - loss: 0.0778 - accuracy: 0.9725 - val_loss: 0.1895 - val_accuracy:
0.9200 - lr: 1.8000e-06
```

```python
def plot_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    epochs = range(1, len(acc) + 1)
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    plt.plot(epochs, acc, 'b', label='Training accuracy')
    plt.plot(epochs, val_acc, 'r', label='Validation accuracy')
    plt.title('Training and Validation Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()
    plt.subplot(1, 2, 2)
    plt.plot(epochs, loss, 'b', label='Training loss')
    plt.plot(epochs, val_loss, 'r', label='Validation loss')
    plt.title('Training and Validation Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()
    plt.tight_layout()
    plt.show()
plot_history(history)
```

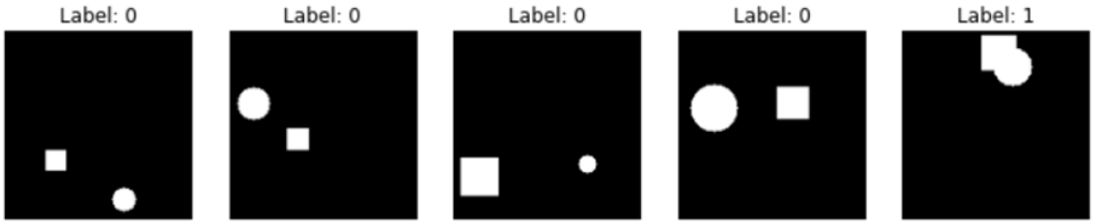



```python
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test accuracy: {accuracy:.2f}")
```

```
19/19 [==============================] - 2s 107ms/step - loss: 0.1895 - accuracy: 0.9200
Test accuracy: 0.92
```

```python
# Προβολή εικόνων από το validation set μαζί με τις ετικέτες τους
def show_random_validation_images(X, y, num_images=5):
    idx = np.random.choice(len(X), num_images, replace=False)
    plt.figure(figsize=(12, 6))
```

```
    for i, idx in enumerate(idx):
        plt.subplot(1, num_images, i + 1)
        plt.imshow(X[idx].reshape(128, 128), cmap='gray')
        plt.title(f"Label: {y[idx]}")
        plt.axis('off')
    plt.show()
show_random_validation_images(X_test, y_test)
```



Αλγόριθμος CNN ποσοτικοποίησης της ανομοιογενούς απορρόφησης του ^{123}I

```
import numpy as np
import os
import glob
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import load_img,
img_to_array
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D,
MaxPooling2D, Dropout, BatchNormalization, Input
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
from tensorflow.keras.optimizers import Adam
from PIL import Image
from tensorflow.keras.callbacks import ReduceLROnPlateau

input_folder = r"C:\Users\hlian\OneDrive\Υπολογιστής\Simulix"
output_folder = "images_for_training"

if os.path.exists(output_folder):
    files_to_delete = glob.glob(os.path.join(output_folder, "*"))
    for file in files_to_delete:
        os.remove(file)
    print(f"The data in file {output_folder} were deleted.")
else:
    os.makedirs(output_folder)
    print(f"The file was created: {output_folder}")
```

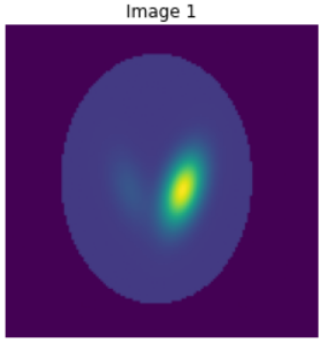
```
The data in file images_for_training were deleted.
```

```
# Φόρτωση των αρχείων .mtx
file_pattern = os.path.join(input_folder, "*.mtx")
file_list = sorted(glob.glob(file_pattern))

# Μετατροπή των αρχείων .mtx σε εικόνες
for idx, file_path in enumerate(file_list):
    try:
        matrix = np.loadtxt(file_path).reshape((128, 128))
        output_image_path = os.path.join(output_folder,
f"image_{idx:04d}.png")
        plt.imsave(output_image_path, matrix, cmap='viridis')
        #print(f"Saved image: {output_image_path}")
    except Exception as e:
        print(f"Error while loading {file_path}: {e}")
data_file_path =
r'C:\Users\hlian\OneDrive\Υπολογιστής\Simulix\simulix_prmt.dat'
data = pd.read_csv(data_file_path, delim_whitespace=True,
header=None)

filtered_data = data.iloc[:, [-3, -2, -1]].copy()

filtered_data.columns = ['Int. of Head Background', 'Int. of Left
HotSpot', 'Int. of Right HotSpot']
```

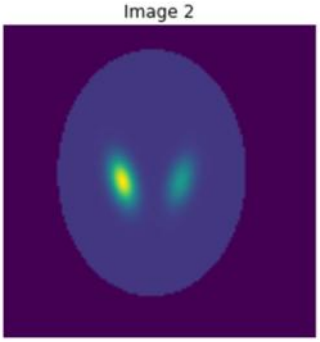
```
filtered_data['Right'] = filtered_data['Int. of Right HotSpot'] +
filtered_data['Int. of Head Background']
filtered_data['Left'] = filtered_data['Int. of Left HotSpot'] +
filtered_data['Int. of Head Background']
filtered_data['max_int'] = filtered_data[['Right',
'Left']].max(axis=1)
# Υπολογισμός των R και L
filtered_data['R'] = filtered_data['Right'] /
filtered_data['max_int']
filtered_data['L'] = filtered_data['Left'] / filtered_data['max_int']

# Υπολογισμός της διαφοράς R - L
filtered_data['R_minus_L'] = filtered_data['R'] - filtered_data['L']

# Αποθήκευση μόνο της στήλης R_minus_L σε αρχείο CSV στον φάκελο
output_folder
csv_file_path = os.path.join(output_folder,
"filtered_data_with_difference.csv")
filtered_data[['R_minus_L']].to_csv(csv_file_path, index=False)
print(f"Data saved at: {csv_file_path}")
```

```
Data saved at: images_for_training\filtered_data_with_difference.csv
```

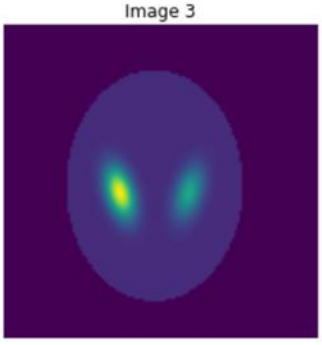
```
for i in range(5):
    image_path = os.path.join(output_folder, f"image_{i:04d}.png")


    if os.path.exists(image_path):
        img = plt.imread(image_path)
        plt.imshow(img, cmap='viridis')
        plt.title(f"Image {i + 1}")
        plt.axis('off')
        plt.show()
        row_data = filtered_data.iloc[i]
        print(f"Data for Image {i + 1}:")
        print(f"  Left HotSpot Intensity: {row_data['Int. of Left
HotSpot']}")
        print(f"  Right HotSpot Intensity: {row_data['Int. of Right
HotSpot']}")
        print(f"  R-L: {row_data['R_minus_L']}\n")
    else:
        print(f"Image {i + 1} not found.")
labels = []
```



Data for Image 1:
  Left HotSpot Intensity: 0.114
  Right HotSpot Intensity: 0.956
  R-L: 0.7302688638334779



Data for Image 2:
  Left HotSpot Intensity: 0.984
  Right HotSpot Intensity: 0.46
  R-L: -0.4463373083475297



Data for Image 3:
  Left HotSpot Intensity: 0.99
  Right HotSpot Intensity: 0.543
  R-L: -0.39487632508833914

```
for i in range(len(file_list)):
    image_path = os.path.join(output_folder, f"image_{i:04d}.png")

        if os.path.exists(image_path):
        difference = filtered_data.iloc[i]['R_minus_L']
        labels.append(difference)
    else:
        print(f"Image {i + 1} not found.")

# Εμφάνιση των labels για έλεγχο
print("Labels list:")
print(labels[:5])

# Φόρτωση των εικόνων
def load_and_preprocess_images(directory):
    images = []
    for filename in sorted(os.listdir(directory)):
        if filename.endswith(".png"):
            img = Image.open(os.path.join(directory,
filename)).convert('L')
            img = img.resize((128, 128))
            img = np.array(img) / 255.0
            img = np.expand_dims(img, axis=-1)
            images.append(img)
    return np.array(images)

def load_labels(csv_file_path):
    data = pd.read_csv(csv_file_path)
    labels = data['R_minus_L'].to_numpy()
    return labels
output_folder = "images_for_training"
csv_file_path = os.path.join(output_folder,
"filtered_data_with_difference.csv")

# Κλήση συναρτήσεων για φόρτωση εικόνων και ετικετών
images = load_and_preprocess_images(output_folder)
labels = load_labels(csv_file_path)

import tensorflow.keras.backend as K
X_train, X_test, y_train, y_test = train_test_split(images, labels,
test_size=0.2, random_state=42)

# Συνάρτηση RMSE
def rmse(y_true, y_pred):
    return K.sqrt(K.mean(K.square(y_pred - y_true)))

from tensorflow.keras.metrics import MeanSquaredError

# Δημιουργία του μοντέλου CNN
model = Sequential([
    Input(shape=(128, 128, 1)),
    Conv2D(32, (3, 3), activation='relu'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
```

```
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(1, activation='linear')
])

# Επιλογή του κατάλληλου optimizer
initial_lr = 4e-5
optimizer = Adam(learning_rate=initial_lr)
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.3,
    patience=2,
    min_lr=1e-6,
    verbose=1
)

model.compile(optimizer=optimizer, loss='mean_squared_error',
metrics=[MeanSquaredError(), rmse])

# Εκπαίδευση
history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=30,
    batch_size=16,
    verbose=1,
    callbacks=[reduce_lr]
)
```

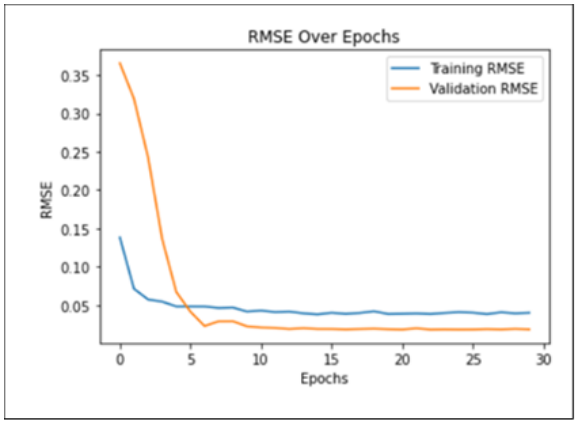
```
Epoch 1/30
150/150 [==============================] - 44s 280ms/step - loss: 0.0247 - mean_squared_error: 0.0247 - rmse: 0.1382 - val_los
s: 0.1354 - val_mean_squared_error: 0.1354 - val_rmse: 0.3657 - lr: 4.0000e-05
Epoch 2/30
150/150 [==============================] - 47s 315ms/step - loss: 0.0053 - mean_squared_error: 0.0053 - rmse: 0.0711 - val_los
s: 0.1032 - val_mean_squared_error: 0.1032 - val_rmse: 0.3193 - lr: 4.0000e-05
```

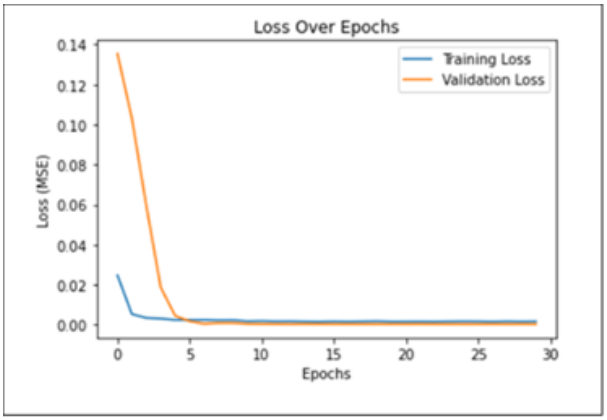
...

```
Epoch 29/30
150/150 [==============================] - 43s 290ms/step - loss: 0.0016 - mean_squared_error: 0.0016 - rmse: 0.0390 - val_los
s: 3.5962e-04 - val_mean_squared_error: 3.5962e-04 - val_rmse: 0.0187 - lr: 1.0000e-06
Epoch 30/30
150/150 [==============================] - 43s 288ms/step - loss: 0.0017 - mean_squared_error: 0.0017 - rmse: 0.0397 - val_los
s: 3.3782e-04 - val_mean_squared_error: 3.3782e-04 - val_rmse: 0.0181 - lr: 1.0000e-06
```

```
# Οπτικοποίηση της εξέλιξης του RMSE και Loss
plt.plot(history.history['rmse'], label='Training RMSE')
plt.plot(history.history['val_rmse'], label='Validation RMSE')
plt.title('RMSE Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('RMSE')
plt.legend()
plt.grid(False)
plt.show()

plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
```

```
plt.title('Loss Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss (MSE)')
plt.legend()
plt.grid(False)
plt.show()
```






```
# Υπολογισμός απόδοσης στο test set
from sklearn.metrics import mean_squared_error, mean_absolute_error

y_pred = model.predict(X_test)

# Υπολογισμός στα κανονικοποιημένα δεδομένα
mse = mean_squared_error(y_test, y_pred.flatten())
mae = mean_absolute_error(y_test, y_pred.flatten())
rmse_value = np.sqrt(mse)

print(f"Test Set Performance:")
print(f"MSE: {mse:.4f}")
print(f"RMSE: {rmse_value:.4f}")
```

```
19/19 [==============================] - 3s 125ms/step
Test Set Performance:
MSE: 0.0003
RMSE: 0.0184
```

```
# Τυχαία δείγματα από το test set
import random

random_indices = random.sample(range(len(X_test)), 10)

plt.figure(figsize=(15, 15))
for i, idx in enumerate(random_indices):
    img = X_test[idx].reshape(128, 128)
    real_diff = y_test[idx]
    predicted_diff = y_pred[idx][0]

    plt.subplot(5, 2, i+1)
    plt.imshow(img, cmap='viridis')
    plt.title(f"Real Diff: {real_diff:.4f}\nPredicted Diff:
{predicted_diff:.4f}")
    plt.axis('off')

plt.tight_layout()
plt.show()
```

```
# Υπολογισμός των προβλεπόμενων τιμών για το training set
y_train_pred = model.predict(X_train)

# Scatter plot για πραγματικές vs προβλεπόμενες τιμές
plt.figure(figsize=(8, 8))
plt.scatter(y_train, y_train_pred.flatten(), alpha=0.7,
edgecolors='k', label='Data Points')
plt.plot([min(y_train), max(y_train)], [min(y_train), max(y_train)],
color='red', linestyle='--', label='Perfect Prediction')
plt.title('True vs Predicted Values (Training Set)')
plt.xlabel('True Values (R - L)')
plt.ylabel('Predicted Values (R - L)')
plt.legend()
plt.grid(True)
plt.show()
residuals = y_train_pred - y_train
#Υπολογισμος τυπικής απόκλισης
std_dev = np.std(residuals)
print(f"Standard Deviation of Residuals: {std_dev:.4f}")
```

```
Standard Deviation of Residuals: 0.5614
```

```
y_train_pred = np.array(y_train_pred).flatten()
y_train = np.array(y_train).flatten()

# Υπολογισμός residuals
residuals = y_train_pred - y_train
residuals = np.ravel(residuals)
residuals = np.array(residuals)
residuals = np.array(residuals).flatten()

# Δημιουργία histogram
plt.figure(figsize=(10, 6))
sns.histplot(residuals, kde=True, color='blue', bins=30,
label='Residuals Distribution')
plt.axvline(0, color='red', linestyle='--', label='Zero Error Line')
plt.title('Residuals Distribution (Predicted - True Values)')
plt.xlabel('Residuals (Predicted - True)')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.show()
```

Αλγόριθμος CNN πρόβλεψης των εντάσεων κάθε λοβού και του υποβάθρου

```
import numpy as np
import os
import glob
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import load_img,
img_to_array
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D,
MaxPooling2D, Dropout, BatchNormalization, Input
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
from tensorflow.keras.optimizers import Adam
from PIL import Image
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.metrics import MeanSquaredError
import tensorflow.keras.backend as K

input_folder = r"C:\Users\hlian\OneDrive\Υπολογιστής\Simulix2"
output_folder = "images_for_training4"
# Δημιουργία φακέλου για εικόνες αν δεν υπάρχει
if os.path.exists(output_folder):
    files_to_delete = glob.glob(os.path.join(output_folder, "*"))
    for file in files_to_delete:
        os.remove(file)
    print(f"The data in file {output_folder} were deleted.")
else:
    os.makedirs(output_folder)
    print(f"The file was created: {output_folder}")
file_pattern = os.path.join(input_folder, "*.mtx")
file_list = sorted(glob.glob(file_pattern))
 The data in file images_for_training4 were deleted.
# Μετατροπή των αρχείων .mtx σε εικόνες και αποθήκευση
image_data = []
for idx, file_path in enumerate(file_list):
    try:
        matrix = np.loadtxt(file_path).reshape((128, 128))
        image_data.append(matrix)
        output_image_path = os.path.join(output_folder,
f"image_{idx:04d}.png")
        plt.imsave(output_image_path, matrix, cmap='viridis')
    except Exception as e:
        print(f"Error while loading {file_path}: {e}")
data_file_path =
r'C:\Users\hlian\OneDrive\Υπολογιστής\Simulix2\simulix_prmt.dat'
data = pd.read_csv(data_file_path, delim_whitespace=True,
header=None)
filtered_data = data.iloc[:, [0] + list(range(-3, 0))].copy()

# Επανετικέτα στις στήλες
filtered_data.columns = [
    'enumerate', 'Background', 'Int. of Left HotSpot', 'Int. of Right
HotSpot']
```

```
csv_file_path = os.path.join(output_folder, "filtered_data.csv")
filtered_data.to_csv(csv_file_path, index=False)
print(f"All filtered data saved at: {csv_file_path}")
X = np.array(image_data)
X = X.reshape((-1, 128, 128, 1))
# Ετικέτες (y)
y = filtered_data[['Background','Int. of Left HotSpot', 'Int. of
Right HotSpot']].values
```

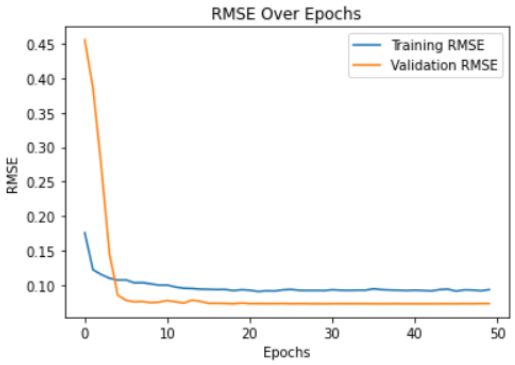
```
All filtered data saved at: images_for_training4\filtered_data.csv
```

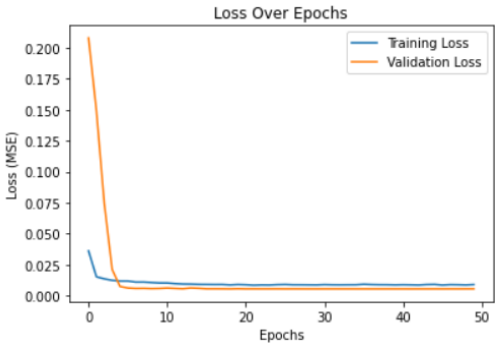
```
# Διαχωρισμός των δεδομένων σε train και test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Συνάρτηση RMSE
def rmse(y_true, y_pred):
    return K.sqrt(K.mean(K.square(y_pred - y_true)))
model = Sequential([
    Input(shape=(128, 128, 1)),

    Conv2D(32, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu', padding='same'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu', padding='same'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu', padding='same'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu', padding='same'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.2),
    Dense(3, activation='linear')
])
initial_lr = 4e-5
optimizer = Adam(learning_rate=initial_lr)
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.3,
    patience=2,
    min_lr=1e-7,
    verbose=1
)

# Συγκέντρωση του μοντέλου με MSE ως metric
model.compile(optimizer=optimizer, loss='mean_squared_error',
metrics=[MeanSquaredError(), rmse])

# Εκπαίδευση
history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=50,
    batch_size=16,
```

```
    verbose=1,
    callbacks=[reduce_lr]
)
```

```
Epoch 1/50
150/150 [==============================] - 35s 220ms/step - loss: 0.0361 - mean_squared_error: 0.0361 - rmse: 0.1760 - val_los
s: 0.2078 - val_mean_squared_error: 0.2078 - val_rmse: 0.4554 - lr: 4.0000e-05
Epoch 2/50
150/150 [==============================] - 38s 251ms/step - loss: 0.0152 - mean_squared_error: 0.0152 - rmse: 0.1223 - val_los
s: 0.1491 - val_mean_squared_error: 0.1491 - val_rmse: 0.3858 - lr: 4.0000e-05
                                                                ...
Epoch 49/50
150/150 [==============================] - 28s 187ms/step - loss: 0.0086 - mean_squared_error: 0.0086 - rmse: 0.0920 - val_los
s: 0.0055 - val_mean_squared_error: 0.0055 - val_rmse: 0.0734 - lr: 1.0000e-07
Epoch 50/50
150/150 [==============================] - 28s 184ms/step - loss: 0.0089 - mean_squared_error: 0.0089 - rmse: 0.0936 - val_los
s: 0.0055 - val_mean_squared_error: 0.0055 - val_rmse: 0.0734 - lr: 1.0000e-07
```

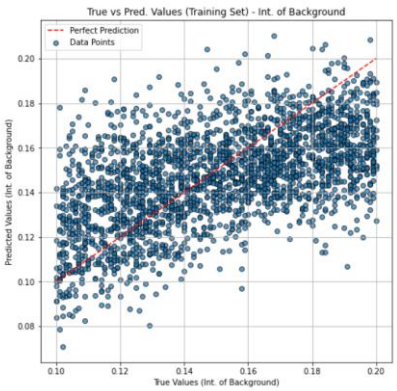
```
# Οπτικοποίηση της εξέλιξης του RMSE και Loss
plt.plot(history.history['rmse'], label='Training RMSE')
plt.plot(history.history['val_rmse'], label='Validation RMSE')
plt.title('RMSE Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('RMSE')
plt.legend()
plt.grid(False)
plt.show()

plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Loss Over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss (MSE)')
plt.legend()
plt.grid(False)
plt.show()
```

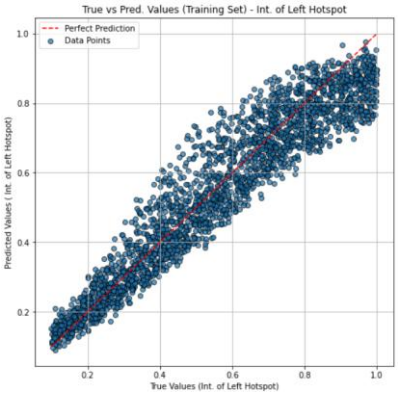


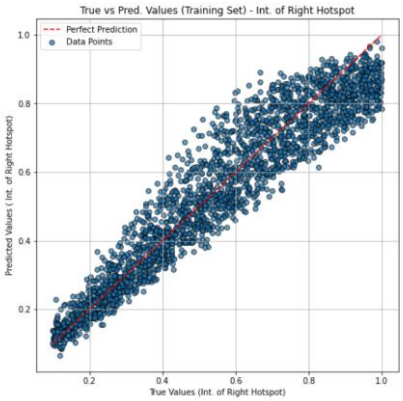



```
# Υπολογισμός των προβλεπόμενων τιμών για το training set
y_train_pred = model.predict(X_train)

y_train_1, y_train_2,  y_train_3= y_train[:, 0], y_train[:, 1],
y_train[:, 2]
y_train_pred_1, y_train_pred_2, y_train_pred_3 = y_train_pred[:, 0],
y_train_pred[:, 1], y_train_pred[:, 2]

plt.figure(figsize=(8, 8))
plt.scatter(y_train_1, y_train_pred_1, alpha=0.7, edgecolors='k',
label='Data Points')
plt.plot([min(y_train_1), max(y_train_1)], [min(y_train_1),
max(y_train_1)], color='red', linestyle='--', label='Perfect
Prediction')
plt.title('True vs Pred. Values (Training Set) - Int. of Background')
```

```
plt.xlabel('True Values (Int. of Background)')
plt.ylabel('Predicted Values (Int. of Background)')
plt.legend()
plt.grid(True)
plt.show()

# Scatter plot για το δεύτερο μέγεθος
plt.figure(figsize=(8, 8))
plt.scatter(y_train_2, y_train_pred_2, alpha=0.7, edgecolors='k',
label='Data Points')
plt.plot([min(y_train_2), max(y_train_2)], [min(y_train_2),
max(y_train_2)], color='red', linestyle='--', label='Perfect
Prediction')
plt.title('True vs Pred. Values (Training Set) - Int. of Left
Hotspot')
plt.xlabel('True Values (Int. of Left Hotspot)')
plt.ylabel('Predicted Values ( Int. of Left Hotspot)')
plt.legend()
plt.grid(True)
plt.show()

# Scatter plot για το 3ο μέγεθος
plt.figure(figsize=(8, 8))
plt.scatter(y_train_3, y_train_pred_3, alpha=0.7, edgecolors='k',
label='Data Points')
plt.plot([min(y_train_3), max(y_train_3)], [min(y_train_3),
max(y_train_3)], color='red', linestyle='--', label='Perfect
Prediction')
plt.title('True vs Pred. Values (Training Set) - Int. of Right
Hotspot')
plt.xlabel('True Values (Int. of Right Hotspot)')
plt.ylabel('Predicted Values ( Int. of Right Hotspot)')
plt.legend()
plt.grid(True)
plt.show()
```








```
residuals1 = y_train_pred_1 - y_train_1
#Υπολογισμος τυπικής απόκλισης
std_dev = np.std(residuals1)
print(f"Standard Deviation of Residuals/Int. of Background:
{std_dev:.4f}")
residuals2 = y_train_pred_2 - y_train_2
#Υπολογισμος τυπικής απόκλισης
std_dev = np.std(residuals2)
```

```
print(f"Standard Deviation of Residuals/Int. of Left Hotspot:
{std_dev:.4f}")
residuals3 = y_train_pred_3 - y_train_3
#Υπολογισμος τυπικής απόκλισης
std_dev = np.std(residuals3)
print(f"Standard Deviation of Residuals/Int. of Right Hotspot:
{std_dev:.4f}")
```

```
Standard Deviation of Residuals/Int. of Background: 0.0245
Standard Deviation of Residuals/Int. of Left Hotspot: 0.0776
Standard Deviation of Residuals/Int. of Right Hotspot: 0.0779
```

```
import numpy as np
from sklearn.metrics import mean_squared_error, mean_absolute_error

y_pred = model.predict(X_test)

y_test_1, y_test_2, y_test_3  = y_test[:, 0], y_test[:, 1], y_test[:,
2]
y_pred_1, y_pred_2,y_pred_3  = y_pred[:, 0], y_pred[:, 1], y_pred[:,
2]

mse_1 = mean_squared_error(y_test_1, y_pred_1)
rmse_1 = np.sqrt(mse_1)
mse_2 = mean_squared_error(y_test_2, y_pred_2)
rmse_2 = np.sqrt(mse_2)
mse_3 = mean_squared_error(y_test_3, y_pred_3)
rmse_3 = np.sqrt(mse_3)
# Εμφάνιση αποτελεσμάτων
print("Test Set Performance for Int of Background:")
print(f"MSE: {mse_1:.4f}")
print(f"RMSE: {rmse_1:.4f}")

print("\nTest Set Performance for Int. of Right Hotspot:")
print(f"MSE: {mse_2:.4f}")
print(f"RMSE: {rmse_2:.4f}")

print("\nTest Set Performance for Int. of Right Hotspot:")
print(f"MSE: {mse_2:.4f}")
print(f"RMSE: {rmse_2:.4f}")
```

```
19/19 [==============================] - 2s 78ms/step
Test Set Performance for Int of Background:
MSE: 0.0007
RMSE: 0.0268

Test Set Performance for Int. of Left Hotspot:
MSE: 0.0080
RMSE: 0.0893

Test Set Performance for Int. of Right Hotspot:
MSE: 0.0080
RMSE: 0.0893
```