



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

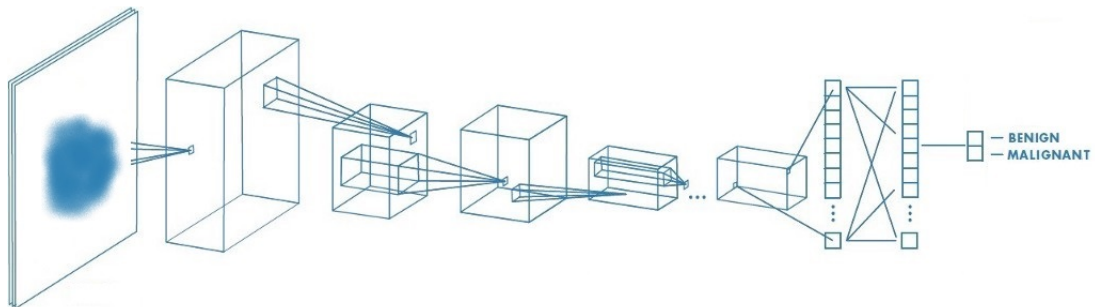
# Ταξινόμηση Μελαγχρωματικών Βλαβών Με Μεθόδους Μηχανικής Μάθησης

---

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

**ΛΑΔΑΚΗ ΓΕΩΡΓΙΟΥ**



**Επιβλέπων:** Ανδρέας - Γεώργιος Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

**Συνεπιβλέπων:** Σιόλας Γεώργιος  
ΕΔΙΠ Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2020

---





## Ταξινόμηση Μελαγχρωματικών Βλαβών Με Μεθόδους Μηχανικής Μάθησης

---

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

**ΛΑΔΑΚΗ ΓΕΩΡΓΙΟΥ**

**Επιβλέπων:** Ανδρέας - Γεώργιος Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

**Συνεπιβλέπων:** Σιόλας Γεώργιος  
ΕΔΙΠ Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 10η Σεπτεμβρίου 2020.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Ανδρέας - Γεώργιος Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

.....  
Κωνσταντίνα Νικήτα  
Καθηγήτρια Ε.Μ.Π.

.....  
Γεώργιος Στάμου  
Αναπληρωτής Καθηγητής Ε.Μ.Π.





ΕΘΝΙΚΟ ΜΕΤΕΩΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Copyright © – All rights reserved. Με την επιφύλαξη παντός δικαιώματος.  
Λαδάκης Γεώργιος, 2020.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.



# Περίληψη

---

Την τελευταία δεκαετία οι εξελίξεις στον κλάδο της Τεχνητής Νοημοσύνης (Artificial Intelligence) παρουσιάζουν εκθετική αύξηση. Η διεύρυνση του όγκου δεδομένων σε συνδυασμό με την σημαντική ανάπτυξη των υπολογιστικών πόρων δημιούργησαν γόνιμο έδαφος για την άνθηση της Βαθιάς Μάθησης (Deep Learning) και εν συνεχεία της Όρασης Υπολογιστών (Computer Vision). Ένα πολλά υποσχόμενο παρακλάδι της Όρασης Υπολογιστών αφορά την υποβοηθούμενη διάγνωση (computer-aided diagnosis) με την χρήση ιατρικών εικόνων. Σε συνέχεια των παραπάνω, αποφασίσαμε να ασχοληθούμε με την ταξινόμηση μελαγχρωματικών βλαβών ως καλοήθεις ή κακοήθεις με την χρήση Συνελκτικών Νευρωνικών Δικτύων (Convolutional Neural Networks - CNNs).

Σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη των δομικών στοιχείων των CNNs, η ανάλυση των μεθόδων εκπαίδευσης και αξιολόγησης τους, καθώς και η παρουσίαση των σημαντικότερων CNN αρχιτεκτονικών. Στη συνέχεια, ακολουθεί η υλοποίηση και βελτιστοποίηση μοντέλων με στόχο την επίτευξη της καλύτερης δυνατής απόδοσης για το συγκεκριμένο πρόβλημα. Τα δεδομένα για την υλοποίηση των εν λόγω μοντέλων λήφθηκαν από τον διαγωνισμό “ISIC Challenge 2016 - Task 3: Lesion Classification”. Με τη χρήση state-of-the-art αρχιτεκτονικών και μεθόδων Βαθιάς Μάθησης καταφέραμε να ξεπεράσουμε τις επιδόσεις των νικητών του διαγωνισμού. Τέλος, καταλήγουμε σε ορισμένα συμπεράσματα ενώ προτείνουμε κατευθύνσεις για την περαιτέρω βελτίωση της επίδοσης των μοντέλων.

## Λέξεις Κλειδιά

Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Βαθιά Μάθηση, Ταξινόμηση, Συνελκτικά Νευρωνικά Δίκτυα, VGG, ResNet, EfficientNet, Transfer Learning, Μελάνωμα





# Abstract

---

Over the last decade, advances in the world of Artificial Intelligence have grown exponentially. The availability of large data acquisition and the advancement of computing power has motivated the flourish of Deep Learning, and consequently the rise of Computer Vision. A very promising subfield of Computer Vision is computer-aided diagnosis through medical imaging. Taking into account the developments mentioned above, we decided to work on classifying pigmented skin lesions in benign (non-melanoma) and malignant (melanoma) using convolutional neural networks (CNNs).

The purpose of this graduation thesis is to study the building blocks of CNNs, to analyze their training and evaluation methods and to present some of the most important CNN architectures. Additionally, we implement and optimize some models in pursuit of the highest possible score. The dataset used in our implementations was provided in the competition “ISIC Challenge 2016 - Task 3: Lesion Classification”. By utilizing state-of-the-art architectures and deep learning methods we achieved higher scores than the competition winners. Finally, we draw some conclusions and suggest ways to further improve our models in future work.

## Keywords

Artificial Intelligence, Machine Learning, Deep Learning, Classification, Convolutional Neural Networks, VGG, ResNet, EfficientNet, Transfer Learning, Melanoma



## Ευχαριστίες

---

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης στα πλαίσια του προπτυχιακού προγράμματος σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Ξεκινώντας, θα ήθελα να ευχαριστήσω τον επιβλέπων μου κ. Ανδρέα-Γεώργιο Σταφυλοπάτη, Καθηγητή Ε.Μ.Π, ο οποίος μου έδωσε τη δυνατότητα να διευρύνω τις γνώσεις μου στο πεδίο της Μηχανικής Μάθησης.

Παράλληλα, θα ήθελα να ευχαριστήσω τον συνεπιβλέπων κ. Γεώργιο Σιόλα, ΕΔΙΠ Ε.Μ.Π, για την άψογη συνεργασία μας στα πλαίσια της εργασίας αλλά και για τον ζήλο με τον οποίο ανταποκρινόταν στις απορίες μου κατά τη διάρκεια των σπουδών μου.

Κλείνοντας, καθώς η παρούσα διπλωματική εργασία σηματοδοτεί την ολοκλήρωση των σπουδών μου, δεν θα μπορούσα να παραλείψω την οικογένειά μου καθώς και όλα τα κοντινά και αγαπημένα μου πρόσωπα για την αμέριστη στήριξη που έδειξαν όλα αυτά τα χρόνια. Τους ευχαριστώ για τη συμπαράστασή και για όλες τις θυσίες που έκαναν για μένα.

Αθήνα, Σεπτέμβριος 2020

*Λαδάκης Γεώργιος*



# Περιεχόμενα

---

<b>Περίληψη</b>	<b>1</b>
<b>Abstract</b>	<b>3</b>
<b>Ευχαριστίες</b>	<b>5</b>
<b>1 Εισαγωγή</b>	<b>17</b>
1.1 Τεχνίτη Νοημοσύνη (Artificial Intelligence) - Ιστορική Αναδρομή . . . . .	17
1.2 Μηχανική Μάθηση (Machine Learning) . . . . .	19
1.2.1 Επιβλεπόμενη Μάθηση (Supervised Learning) . . . . .	19
1.2.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) . . . . .	20
1.2.3 Ενισχυτική Μάθηση (Reinforcement Learning) . . . . .	20
1.2.4 Βαθιά Μάθηση (Deep Learning) . . . . .	20
1.3 Τεχνητός Νευρώνας (Artificial Neuron) . . . . .	20
1.4 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) . . . . .	22
1.5 Μελάνωμα . . . . .	22
1.5.1 Αίτια Εμφάνισης Μελανώματος . . . . .	23
1.5.2 Διάγνωση μελανώματος - Ο κανόνας ABCD . . . . .	23
1.5.3 Δερματοσκόπηση . . . . .	25
<b>I Θεωρητικό Μέρος</b>	<b>27</b>
<b>2 Συνελκτικά Νευρωνικά Δίκτυα - Convolutional Neural Networks (CNN)</b>	<b>29</b>
2.1 Συνελκτικό επίπεδο (Convolutional layer) . . . . .	30
2.1.1 Φίλτρα . . . . .	30
2.1.2 Συνέλιξη . . . . .	30
2.1.3 Υπερπαραμέτροι - Διαστάσεις εξόδου . . . . .	31
2.2 Επίπεδο Συγκέντρωσης (Pooling Layer) . . . . .	32
2.3 Πλήρως Συνδεδεμένο Επίπεδο (Fully Connected Layer) . . . . .	33
<b>3 Εκπαίδευση CNN</b>	<b>35</b>
3.1 Συναρτήσεις Κόστους (Cost Functions) . . . . .	35
3.1.1 Μέσο τετραγωνικό σφάλμα (Mean squared error) . . . . .	35
3.1.2 Hinge Loss . . . . .	35
3.1.3 Cross entropy loss . . . . .	36

3.2	Συναρτήσεις Ενεργοποίησης (Activation Functions)	36
3.2.1	Βηματική (Step function)	37
3.2.2	Γραμμική (Linear function)	37
3.2.3	Σιγμοειδής (Sigmoid function)	37
3.2.4	Softmax	38
3.2.5	Υπερβολική Εφαπτομένη (Tanh function)	38
3.2.6	Ανορθωμένη Γραμμική Μονάδα (ReLU)	39
3.2.7	Swish	39
3.3	Κανονικοποίηση (Regularization)	40
3.3.1	Προσαύξηση δεδομένων (Data augmentation)	40
3.3.2	Μέσος όρος μοντέλων (Ensemble model averaging)	40
3.3.3	Dropout	40
3.3.4	Πρόωρη διακοπή εκπαίδευσης (Early stopping)	41
3.4	Αλγόριθμοι Βελτιστοποίησης (Optimizers)	42
3.4.1	Backpropagation	42
3.4.2	Gradient Descent	42
3.4.3	Momentum	44
3.4.4	AdaGrad	45
3.4.5	RMSprop	45
3.4.6	Adam	46
3.5	Μετρικές (Metrics)	47
3.5.1	Confusion Matrix	47
3.5.2	Καμπύλη ROC - AUC	48
3.5.3	Average Precision	49
<b>4</b>	<b>CNN Αρχιτεκτονικές</b>	<b>51</b>
4.1	LeNet	51
4.2	AlexNet	51
4.3	VGG	51
4.4	ResNet	52
4.4.1	Residual Block	52
4.5	EfficientNet	55
4.5.1	EfficientNet-B0	55
4.5.2	Inverted Residual Block	55
4.5.3	Squeeze & Excitation Block (SE block)	56
<b>5</b>	<b>Transfer Learning</b>	<b>57</b>
<b>II</b>	<b>Πρακτικό Μέρος</b>	<b>59</b>
<b>6</b>	<b>Εργαλεία και Βιβλιοθήκες</b>	<b>61</b>
6.1	Python Libraries	61
6.1.1	NumPy	61

6.1.2	Matplotlib	61
6.1.3	Scikit-learn	61
6.2	Keras	61
6.3	Google Colaboratory	62
<b>7</b>	<b>Υλοποίηση μοντέλων</b>	<b>63</b>
7.1	Συλλογή Δεδομένων	63
7.2	Προεπεξεργασία Δεδομένων	64
7.2.1	Min-Max Normalization	64
7.2.2	Standardization	65
7.2.3	Histogram Stretching	65
7.2.4	Color Constancy - Gray World	65
7.3	Επιλογή Αρχιτεκτονικής Δικτύου	66
7.4	Εκπαίδευση Δικτύου	67
7.4.1	Η callback κλάση ModelCheckpoint	67
7.4.2	Η callback κλάση ReduceLROnPlateau	67
7.4.3	Η κλάση ImageDataGenerator	68
7.4.4	Class Weights	68
7.5	Αξιολόγηση Συστήματος	69
7.5.1	ConvNet	69
7.5.2	VGG	71
7.5.3	ResNet	72
7.5.4	EfficientNet	73
<b>8</b>	<b>Επίλογος</b>	<b>79</b>
8.1	Συμπεράσματα	79
8.2	Μελλοντικές Επεκτάσεις	80
	<b>Βιβλιογραφία</b>	<b>85</b>





# Κατάλογος Σχημάτων

---

1.1	Τεχνητός νευρώνας . . . . .	21
1.2	Επίπεδα ANN . . . . .	22
2.1	Δομή συνελκτικού νευρωνικού δικτύου . . . . .	30
3.1	Hinge Loss . . . . .	36
3.2	Βηματική συνάρτηση . . . . .	37
3.3	Σιγμοειδής συνάρτηση . . . . .	38
3.4	Υπερβολική Εφαπτομένη . . . . .	39
3.5	ReLU . . . . .	39
3.6	Swish . . . . .	40
3.7	Dropout . . . . .	41
3.8	Early stopping . . . . .	41
3.9	Gradient Descent . . . . .	43
3.10	Saddle Point . . . . .	44
3.11	Αλγόριθμοι Βελτιστοποίησης . . . . .	47
3.12	Καμπύλη ROC-AUC . . . . .	49
4.1	VGG16 . . . . .	52
4.2	Residual Block . . . . .	53
4.3	ResNet . . . . .	54
4.4	Residual Block . . . . .	55
4.5	Inverted Residual Block . . . . .	56
5.1	Transfer Learning . . . . .	58
7.1	Πρίν την εφαρμογή Histogram Stretching . . . . .	65
7.2	Μετά την εφαρμογή Histogram Stretching . . . . .	65
7.3	EfficientNet on Imagenet . . . . .	67
7.4	Γραφικές παραστάσεις εκπαιδεύσεις ConvNet . . . . .	70
7.5	Γραφικές παραστάσεις εκπαιδεύσεις VGG . . . . .	71
7.6	Γραφικές παραστάσεις εκπαιδεύσεις ResNet . . . . .	73
7.7	Γραφικές παραστάσεις εκπαιδεύσεις EfficientNet . . . . .	74



## Κατάλογος Εικόνων

---

1.1	Bombe	17
1.2	Deep Blue	18
1.3	Deep Learning subset of Machine Learning subset of Artificial Intelligence	20
1.4	Ασυμμετρία βλάβης	24
1.5	Όρια βλάβης	24
1.6	Χρώμα βλάβης	24
1.7	Δερματοσκοπικές δομές	25
1.8	Δερματοσκόπηση	26
2.1	Συνέλιξη	31
2.2	Zero Padding	32
2.3	Max Pooling	32
3.1	Backpropagation	42
3.2	Momentum	45
3.3	Confusion Matrix	47
7.1	RGB Εικόνα	64
7.2	Εικόνες στην αρχική τους μορφή	66
7.3	Εικόνες μετά το Histogram Stretching	66
7.4	Εικόνες μετά τον αλγόριθμο Color Constancy - Gray World	66
7.5	Επαύξηση Δεδομένων	68
7.6	ConvNet Confusion Matrix	70
7.7	VGG Confusion Matrix	72
7.8	ResNet Confusion Matrix	73
7.9	EfficientNet Confusion Matrix	75
7.10	Mislabeled images	77



## Κατάλογος Πινάκων

---

1.1	Βαθμολόγηση ABCD κριτηρίων . . . . .	25
7.1	ConvNet μετρικές . . . . .	70
7.2	VGG μετρικές . . . . .	72
7.3	ResNet μετρικές . . . . .	73
7.4	EfficientNet μετρικές . . . . .	75
7.5	Leaderboard . . . . .	76



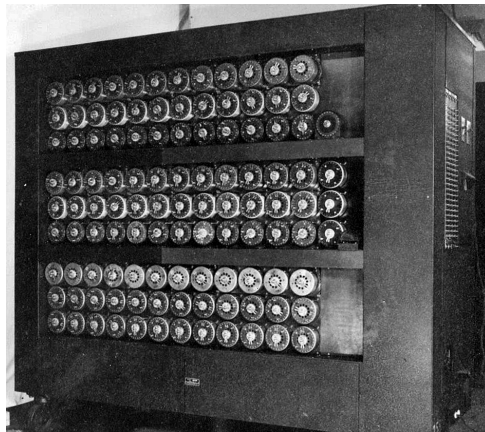
# Κεφάλαιο **1**

## Εισαγωγή

---

### 1.1 Τεχνητή Νοημοσύνη (Artificial Intelligence) - Ιστορική Αναδρομή

Παρότι είναι δύσκολο να προσδιοριστεί με ακρίβεια, η γέννηση του κλάδου της τεχνητής νοημοσύνης εντοπίζεται τη δεκαετία του 1940. Την εποχή εκείνη, ο Άγγλος μαθηματικός Alan Turing με την Ομάδα 8 (Hut 8) κατασκεύασε την Bombe, μία μηχανή για την αποκρυπτογράφηση του κώδικα Enigma που χρησιμοποιούσε ο γερμανικός στρατός κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου. Το 1950 ο Alan Turing δημοσίευσε το άρθρο "Computing Machinery and Intelligence" [1] στο οποίο πρότεινε ένα κριτήριο για τον προσδιορισμό των ευφύων μηχανών, γνωστό ως Turing Test: Αν μία μηχανή καταφέρει να ξεγελάσει έναν άνθρωπο πείθοντας τον πώς συνομιλεί με έναν άλλον άνθρωπο τότε θεωρείται ότι η μηχανή διαθέτει νοημοσύνη.



Εικόνα 1.1: Η μηχανή Bombe [2]

Ο όρος "τεχνητή νοημοσύνη" (Artificial Intelligence - AI) καθιερώθηκε όταν έξι χρόνια αργότερα, το 1956, οι Marvin Minsky και John McCarthy φιλοξένησαν το συνέδριο "Dartmouth Summer Research Project on Artificial Intelligence" (DSRPAI) [3] στο Dartmouth College. Ο σκοπός του συνεδρίου ήταν η δημιουργία ενός νέου ερευνητικού πεδίου που στόχευε στην υλοποίηση μηχανών ικανών να προσομοιώσουν την ανθρώπινη νοημοσύνη. Οι συνέδριοι του DSRPAI θεωρούνται οι θεμελιωτές της τεχνητής νοημοσύνης.

Τα χρόνια που ακολούθησαν βρήκαν τον κλάδο της τεχνητής νοημοσύνης σε ανάπτυξη

με αρκετές επιτυχίες. Χαρακτηριστικό παράδειγμα αποτελεί το πρόγραμμα Eliza [4] που ανέπτυξε το 1966 ο Joseph Weizenbaum στο MIT με το οποίο κατάφερε να πείσει την γραμματέα του πώς συνομιλούσε με τον ίδιο. Το πρόγραμμα αυτό θεωρείται το πρώτο που πέρασε επιτυχώς το Turing Test. Μία ακόμα μεγάλη επιτυχία θεωρείται το πρόγραμμα General Problem Solver του νομπελίστα Herbert Simon το οποίο έλυne αυτόματα μία σειρά προβλημάτων όπως ο πύργος του Hanoi [5] κ.ά.. Οι παραπάνω επιτυχίες συνέβαλαν στη αύξηση της χρηματοδότησης του ερευνητικού πεδίου AI αλλά και στην επικράτηση ενός κλίματος αισιοδοξίας για το μέλλον του κλάδου. Το 1970, σε συνέντευξη του στο περιοδικό Life Magazine, ο Marvin Minsky δήλωσε ότι σε βάθος οκταετίας οι μηχανές θα είναι εξίσου ευφυείς με τον άνθρωπο. Ωστόσο, μόλις τρία χρόνια αργότερα, το 1973, άρχισαν να εμφανίζονται οι πρώτοι επικριτές, με τον Βρετανό μαθηματικό James Lighthill να ισχυρίζεται ότι οι μηχανές δεν θα καταφέρουν να ξεπεράσουν το ερασιτεχνικό επίπεδο σε παιχνίδια όπως το σκάκι ενώ δεν θα μπορέσουν ποτέ να αναπτύξουν συλλογιστική. Οι παραπάνω φωνές οδήγησαν σε μείωση της χρηματοδότησης και την εγκατάλειψη ερευνητικών προγραμμάτων, σηματοδοτώντας την αρχή μιας περιόδου που ονομάστηκε "Χειμώνας της Τεχνητής Νοημοσύνης".

Ένας από τους λόγους που δεν ικανοποιήθηκαν οι προσδοκίες της εποχής οφείλεται στην προσέγγιση που ακολουθούσαν οι προγραμματιστές κατά την ανάπτυξη συστημάτων όπως το Eliza και το General Problem Solver για την προσομοίωση της ανθρώπινης νοημοσύνης. Συγκεκριμένα τα προγράμματα αυτά ήταν Expert Systems και έκαναν τη θεώρηση ότι η ανθρώπινη νοημοσύνη μπορεί να τυποποιηθεί και να ανακατασκευαστεί με μία σειρά από "if-then" εντολές. Τα Expert systems πράγματι αποδίδουν εξαιρετικά σε προβλήματα που μπορούν να τυποποιηθούν. Χαρακτηριστικό παράδειγμα πετυχημένου Expert System αποτελεί το Deep Blue [6] της IBM το οποίο, το 1997, νίκησε τον παγκόσμιο πρωταθλητή στο σκάκι Gary Kasparov αλλά και την πρόβλεψη του James Lighthill πως κάτι τέτοιο ήταν ανέφικτο.



Εικόνα 1.2: Η αναμέτρηση μεταξύ Gary Kasparov και Deep Blue [7]

Αντιθέτως, τα Expert System δεν αποδίδουν καλά σε προβλήματα που δεν μπορούν να τυποποιηθούν. Για παράδειγμα, δεν είναι εύκολο με μία σειρά εντολών "if-then" να υλοποιηθεί επιτυχώς ένα πρόγραμμα αναγνώρισης προσώπου. Για την επίλυση τέτοιων προβλη-



μάτων ένα σύστημα πρέπει να μαθαίνει από εξωτερικά δεδομένα και να χρησιμοποιεί αυτήν τη γνώση για την επίτευξη συγκεκριμένων στόχων και την ανάπτυξη προσαρμοστικότητας - ικανότητας γενίκευσης. Τα Expert Systems δεν διαθέτουν αυτά τα χαρακτηριστικά και τεχνικά δεν μπορούν να προσδιοριστούν ως τεχνητή νοημοσύνη. Στατιστικές μέθοδοι για την επίτευξη πραγματικής τεχνητής νοημοσύνης είχαν ήδη εμφανιστεί το 1940 όταν ο Καναδός ψυχολόγος Donald Hebb ανέπτυξε μία θεωρία μάθησης γνωστή ως Hebbian Learning που προσομοιώνει την λειτουργία των νευρώνων του ανθρώπινου εγκεφάλου. Η θεωρία του Hebb αποτέλεσε ένα επιμέρους ερευνητικό πεδίο της τεχνητής νοημοσύνης το οποίο όμως το 1969 διεκόπη όταν οι Marvin Minsky και Seymour Papert απέδειξαν ότι η επεξεργαστική ισχύς των υπολογιστών δεν μπορούσε να ανταπεξέλθει στις απαιτήσεις των λεγόμενων τεχνητών νευρωνικών δικτύων (Artificial Neural Networks) [8] [9].

Τα τεχνητά νευρωνικά δίκτυα επανήλθαν στα πλαίσια της βαθιάς μάθησης όταν το 2015 το AlfaGo, κέρδισε τον παγκόσμιο πρωταθλητή του Go, Ke Jie. Το Go είναι ένα παιχνίδι πολύ πιο περίπλοκο από το σκάκι για το οποίο επικρατούσε η άποψη ότι οι υπολογιστές δεν θα καταφέρουν πότε να κερδίσουν τον άνθρωπο. Σήμερα, τα νευρωνικά δίκτυα και η βαθιά μάθηση χρησιμοποιούνται για την ανάπτυξη των περισσότερων εφαρμογών που χαρακτηρίζονται ως AI [9]. Τέτοιες εφαρμογές είναι η αναγνώριση προσώπου (face recognition), η αναγνώριση λόγου (speech recognition) κ.ά.. *“Αυτή η πρόταση για παράδειγμα δεν πληκτρολογεί θηκε αλληλά εκφωνήθηκε και στη συνέχεια μετατράπηκε σε κείμενο μέσω της υπηρεσίας speech to text Recognition της Google”*.

## 1.2 Μηχανική Μάθηση (Machine Learning)

Η μηχανική μάθηση αποτελεί παρακλάδι της τεχνητής νοημοσύνης. Το 1959, ο Arthur Samuel εισάγει τον όρο “μηχανική μάθηση” και την ορίζει ως “το ερευνητικό πεδίο που δίνει τη δυνατότητα στους υπολογιστές να μαθαίνουν, λύνοντας προβλήματα για τα οποία δεν έχουν προγραμματιστεί ρητά” [10].

Ένας τυπικός ορισμός για τη μηχανική μάθηση δόθηκε από τον Tom M. Mitchell το 1997: “Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία  $E$  ως προς κάποια κλάση εργασιών  $T$  και μέτρο απόδοσης  $P$ , αν η απόδοσή του σε εργασίες από το  $T$ , όπως μετριέται από το  $P$ , βελτιώνεται μέσω της εμπειρίας  $E$ ” [11].

Οι τεχνικές μηχανικής μάθησης που έχουν αναπτυχθεί μπορούν να κατηγοριοποιηθούν ανάλογα με τη φύση του προβλήματος που καλούνται να αντιμετωπίσουν:

### 1.2.1 Επιβλεπόμενη Μάθηση (Supervised Learning)

Το υπολογιστικό σύστημα καλείται να “μάθει” μία συνάρτηση από ένα σύνολο δεδομένων. Η τεχνική αυτή χαρακτηρίζεται ως επιβλεπόμενη μάθηση επειδή θεωρείται ότι υπάρχει κάποιος “επιβλέπων” ο οποίος παρέχει τη σωστή τιμή εξόδου της συνάρτησης. Βρίσκει εφαρμογή σε προβλήματα:

- Ταξινόμησης (Classification)
- Παρεμβολής (Regression)

### 1.2.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

Σε αυτή την περίπτωση το σύστημα καλείται να ανακαλύψει μόνο του συσχετίσεις μεταξύ του συνόλου δεδομένων χωρίς να είναι εκ των προτέρων γνωστό αν υπάρχουν και ποιες είναι. Βρίσκει εφαρμογή σε προβλήματα :

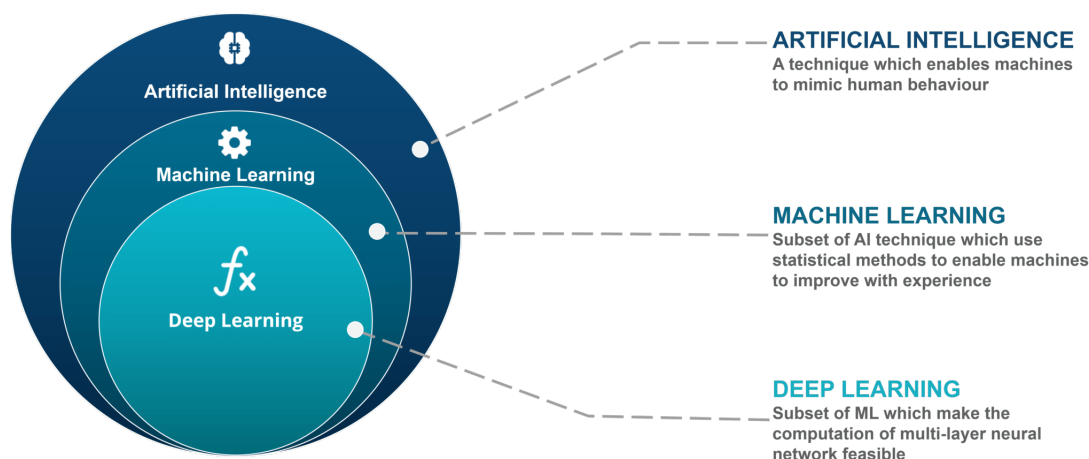
- Ανάλυσης Συσχετίσεων (Association Analysis)
- Ομαδοποίησης (Clustering)

### 1.2.3 Ενισχυτική Μάθηση (Reinforcement Learning)

Το υπολογιστικό σύστημα προσπαθεί να μάθει μέσα από την άμεση αλληλεπίδραση με το περιβάλλον. Αυτό επιτυγχάνεται μέσω "επιβράβευσης" ή "τιμωρίας" του συστήματος κατά τη διάρκεια της αλληλεπιδράσεως του με το περιβάλλον. Στόχος του είναι η μεγιστοποίηση της επιβράβευσης. Βρίσκει εφαρμογή σε προβλήματα Σχεδιασμού (Planning).

### 1.2.4 Βαθιά Μάθηση (Deep Learning)

Η βαθιά μάθηση είναι ένα παρακλάδι της μηχανικής μάθησης που τα τελευταία χρόνια παρουσιάζει ραγδαία ανάπτυξη. Η βαθιά μάθηση χρησιμοποιεί δίκτυα πολλαπλών επιπέδων, δηλαδή αυξημένου βάθους, για την εξαγωγή χαρακτηριστικών υψηλού επιπέδου από τα δεδομένα. Η χρήση πολλαπλών επιπέδων επέτρεψε την αποτελεσματική εκπαίδευση δικτύων σε μη-δομημένα δεδομένα όπως οι εικόνες, η φωνή κ.ά.. Οι μέθοδοι βαθιάς μάθησης έδωσαν μεγάλη ώθηση στην όραση υπολογιστών (computer vision), την επεξεργασία φυσικής γλώσσας (natural language processing) και σε πολλές άλλες εφαρμογές [12].



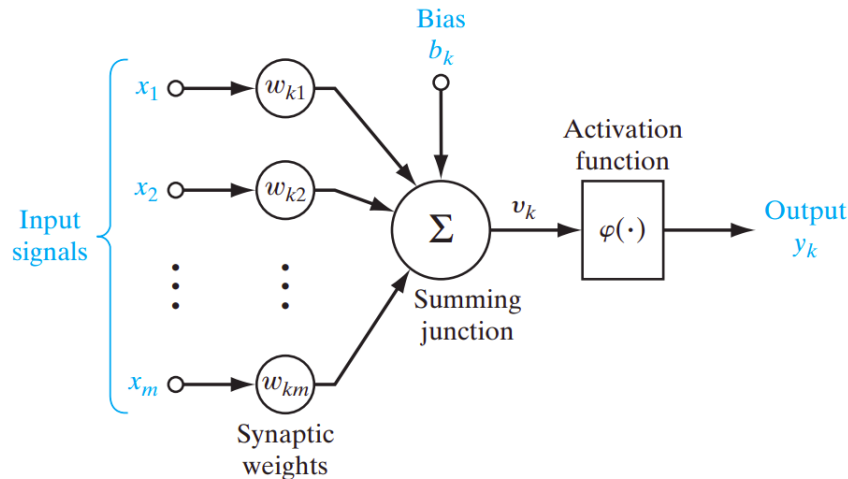
Εικόνα 1.3: Η σχέση μεταξύ Artificial Intelligence, Machine Learning και Deep Learning [13]

## 1.3 Τεχνητός Νευρώνας (Artificial Neuron)

Η πρώτη προσπάθεια μοντελοποίησης ενός νευρώνα έγινε το 1943 από τους Αμερικανούς επιστήμονες McCulloch Pitts, γνωστό ως μοντέλο MCP [14]. Στη συνέχεια, το 1957, ο Frank Rosenblatt πρότεινε το perceptron [15] βελτιώνοντας σημαντικά το MCP ενώ ανέπτυξε και

έναν αλγόριθμο επιβλεπόμενης μάθησης που επέτρεπε την προσαρμογή των παραμέτρων του μοντέλου για την ταξινόμηση γραμμικά διαχωρίσιμων δειγμάτων.

Γενικεύοντας το μοντέλο του Rosenblatt ένας τεχνητός νευρώνας μπορεί να αναπαρασταθεί ως εξής [16]:



Σχήμα 1.1: Τεχνητός νευρώνας [16]

Όπως διακρίνουμε στο σχήμα 1.1, το μοντέλο διαθέτει ένα σύνολο συνάψεων ή διασυνδέσεων, κάθε μία εκ των οποίων έχει ένα συντελεστή  $w$  που ονομάζεται συναπτικό βάρος (synaptic weight). Κάθε σήμα εισόδου  $x_j$  που συνδέεται με το νευρώνα  $k$  σταθμίζεται, δηλαδή πολλαπλασιάζεται, με το αντίστοιχο συναπτικό βάρος  $w_{kj}$ . Ο πρώτος δείκτης ενός συναπτικού βάρους  $w_{kj}$  αναφέρεται στο νευρώνα  $k$  ενώ ο δεύτερος δείκτης  $j$  αναφέρεται στην είσοδο της σύναψης. Στη συνέχεια, αφού τα σήματα εισόδου πολλαπλασιαστούν με τα αντίστοιχα συναπτικά βάρη, αθροίζονται για να δώσουν το δυναμικό ενεργοποίησης  $u_k$  (activation potential). Για την λήψη της τελικής εξόδου  $y$  εφαρμόζεται στην  $u_k$  μία συνάρτηση ενεργοποίησης (activation function)  $\varphi(\cdot)$ . Η συνάρτηση ενεργοποίησης αποκαλείται και συνάρτηση περιορισμού (squashing function) καθώς περιορίζει το επιτρεπτό εύρος πλάτους του σήματος εξόδου σε κάποια πεπερασμένη τιμή συνήθως  $y_k \in [0, 1]$  ή  $y_k \in [-1, 1]$ . Επιπλέον, το μοντέλο του νευρώνα περιλαμβάνει μία εξωτερικά εφαρμοζόμενη πόλωση (bias)  $b_k$  η οποία έχει ως αποτέλεσμα την αύξηση ή τη μείωση της  $u_k$  ανάλογα με το αν είναι θετική ή αρνητική.

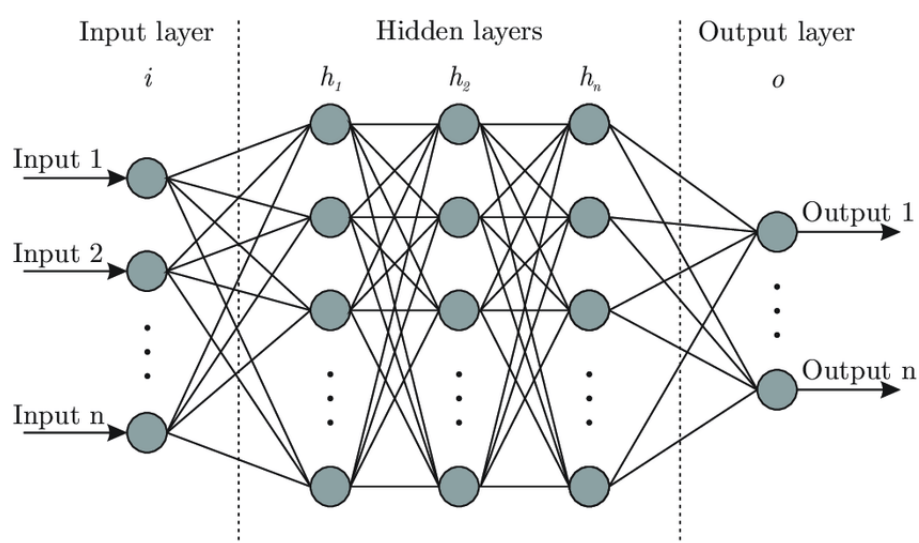
Με μαθηματικούς όρους, το μοντέλο του νευρώνα περιγράφεται από τις ακόλουθες εξισώσεις:

$$u_k = \sum_{j=1}^m w_{kj} x_j + b_k$$

$$y_k = \varphi(u_k + b_k)$$

## 1.4 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)

Με τη χρήση του τεχνητού νευρώνα ως βασικό δομικό στοιχείο μπορούν να αναπτυχθούν τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks - ANNs). Τα μαθηματικά μοντέλα των ANNs αποτελούνται από έναν αριθμό τεχνητών νευρώνων οργανωμένων σε επίπεδα (layers). Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από πολλά επίπεδα τα οποία διακρίνονται σε τρεις βασικές κατηγορίες ανάλογα με την διεργασία που επιτελούν. Συγκεκριμένα, το πρώτο επίπεδο (input layer) ονομάζεται επίπεδο εισόδου και ο ρόλος του είναι η τροφοδότηση των εισόδων στο δίκτυο χωρίς την εκτέλεση υπολογισμών. Το τελευταίο επίπεδο ή επίπεδο εξόδου (output layer) αναλαμβάνει τον υπολογισμό και την τροφοδότηση των τελικών εξόδων του δικτύου στο περιβάλλον. Τέλος όλα τα ενδιάμεσα επίπεδα ονομάζονται κρυφά επίπεδα (hidden layers) και είναι αυτά που εκτελούν τους υπολογισμούς.



Σχήμα 1.2: Τα επίπεδα ενός τεχνητού νευρωνικού δικτύου [17]

## 1.5 Μελάνωμα

Το μελάνωμα αποτελεί έναν εξαιρετικά επιθετικό και θανατηφόρο τύπο καρκίνου του δέρματος. Προέρχεται από τα μελανοκύτταρα που βρίσκονται κάτω από την επιφανειακή στοιβάδα του δέρματος και είναι υπεύθυνα για την παραγωγή της μελανίνης, ουσίας που προστατεύει το δέρμα από την ηλιακή ακτινοβολία και προσδίδει μελανό χρώμα στο δέρμα. Για αυτόν το λόγο το μελάνωμα είναι συνήθως σκούρο καφέ. Στα αρχικά στάδια περιορίζεται στο δέρμα. Ωστόσο, κατά την πρόοδο της νόσου μπορεί να δώσει μεταστάσεις στους λεμφαδένες και σε όργανα, όπως στους πνεύμονες, στο ήπαρ, στα οστά και στον εγκέφαλο. Εκτός από το δέρμα, το μελάνωμα εμφανίζεται σπανιότερα στο μάτι, στο βλεννογόνο της στοματικής κοιλότητας, στο βλεννογόνο του πεπτικού σωλήνα, στα γεννητικά όργανα και στο ουροποιητικό σύστημα.

Η εμφάνιση του μελανώματος συνεχώς αυξάνεται καθιστώντας το ως ένα από τα ταχύτερα εξαπλωμένα κακοήγη νοσήματα στον άνθρωπο. Σύμφωνα με στατιστικές, το μελάνωμα είναι ο έκτος συχνότερος τύπος καρκίνου σε άνδρες και γυναίκες παγκοσμίως. Κρούσματα μελα-

νώματος εντοπίζονται συχνότερα στην Αυστραλία και τη Νέα Ζηλανδία με συχνότητα έως και 3 φορές μεγαλύτερη από ότι στην Ευρώπη. Αντιθέτως, σε αφρικανικές και ασιατικές χώρες το μελάνωμα είναι πιο σπάνιο. Στην Ευρώπη, είναι πιο συχνό στην Ελβετία, την Ολλανδία, την Νορβηγία και τις υπόλοιπες Σκανδιναβικές χώρες όπου ετησίως 20 περίπου άτομα ανά 100.00 κατοίκους διαγιγνώσκονται με μελάνωμα. Λιγότερο συχνό είναι στις Μεσογειακές χώρες, συμπεριλαμβανομένης και της Ελλάδας, με την διάγνωση να τίθεται 4 φορές ανα 100.000 κατοίκους.

### 1.5.1 Αίτια Εμφάνισης Μελανώματος

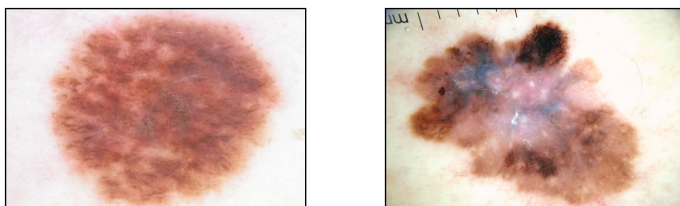
Τα αίτια εμφάνισης μελανώματος όπως και στις περισσότερες κακοήθειες δεν είναι απολύτως γνωστά. Ωστόσο, έχουν εντοπιστεί ορισμένοι ενδογενείς και περιβαλλοντολογικοί παράγοντες που αυξάνουν τον κίνδυνο ανάπτυξης μελανώματος:

- **Ανοιχτόχρωμο δέρμα:** Ο σημαντικότερος ενδογενής παράγοντας είναι το ανοιχτόχρωμο δέρμα. Άτομα με ξανθά ή κόκκινα μαλλιά, με φακίδες στο πρόσωπο και ανοιχτόχρωμα μάτια έχουν αυξημένη πιθανότητα να αναπτύξουν μελάνωμα του δέρματος. Αντιθέτως άτομα με σκούρο ή μαύρο δέρμα και αντίστοιχο χρώμα ματιών δεν αντιμετωπίζουν τον ίδιο κίνδυνο.
- **Παρουσία πολλαπλών σπύλων (ελιών):** Σύμφωνα με μελέτες ο μεγάλος αριθμός σπύλων ενός ατόμου συσχετίζεται με την εμφάνιση μελανώματος. Όσο μεγαλύτερος είναι αυτός ο αριθμός τόσο μεγαλύτερη είναι και η πιθανότητα ανάπτυξης κακοήθειας. Ένα άτομο με 100 σπύλους έχει 7 φορές μεγαλύτερο κίνδυνο από ένα άτομο με 10 σπύλους.
- **Ιστορικό ηλιακών εγκαυμάτων:** Η εμφάνιση μελανώματος συνδέεται άμεσα με το ιστορικό σοβαρών ηλιακών εγκαυμάτων, ιδιαίτερα στην παιδική ηλικία. Τα πρώτα χρόνια ζωής του ατόμου είναι μια πολύ ευαίσθητη χρονική περίοδος όσον αφορά την υπέρμετρη έκθεση στην ηλιακή ακτινοβολία καθώς συμπίπτει με την ανάπτυξη των σπύλων.
- **Οικογενειακό ιστορικό:** Τα άτομα με ιστορικό μελανώματος στην οικογένειά τους έχουν σαφώς αυξημένη πιθανότητα ανάπτυξης μελανώματος στο μέλλον.

### 1.5.2 Διάγνωση μελανώματος - Ο κανόνας ABCD

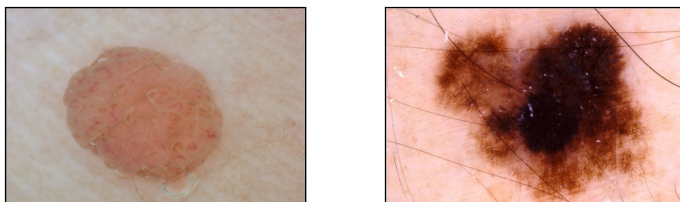
Οι άνθρωποι συνήθως εμφανίζουν κάποιες μελανές κηλίδες στο δέρμα που είτε είναι σπύλοι είτε σημάδια εκ γενετής και αποτελούν μελαγχρωματικές βλάβες. Στην πλειονότητα τους οι παραπάνω βλάβες είναι καλοήθειες, υπάρχει ωστόσο ο κίνδυνος κάποια να αποτελεί μελάνωμα. Το μελάνωμα μπορεί να αναπτυχθεί σε μια υπάρχουσα μελαγχρωματική βλάβη ή ακόμα πιθανότερα σε μια νέα. Ιδιαίτερα σημαντική κρίνεται η παρακολούθηση των αλλαγών και για τις δύο παραπάνω κατηγορίες βλαβών χωρίς αυτό να σημαίνει ότι η παραμικρή αλλαγή στο χρώμα, σχήμα ή μέγεθος συνιστά μελάνωμα. Για την ασφαλέστερη κλινική διάγνωση του μελανώματος έχουν υιοθετηθεί διάφοροι αλγόριθμοι, εμείς θα εστιάσουμε στα κριτήρια ABCD [18]:

- **A (Asymmetry) – Ασυμμετρία:** Η μελαγχρωματική βλάβη είναι ασύμμετρη ως προς τον οριζόντιο ή τον κάθετο άξονα.



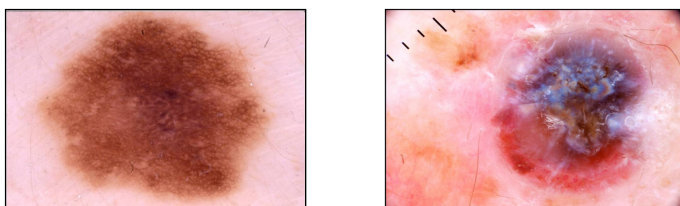
Εικόνα 1.4: Αριστερά διακρίνουμε μία καλοήγη μελαγχρωματική βλάβη ενώ δεξιά μία κακοήγη.

- **B (Border) – Όρια της βλάβης:** Τα όρια της βλάβης είναι ασαφή, θολωμένα ή ανώμαλα και η μελάγχρωση επεκτείνεται στον παρακείμενο ιστό.



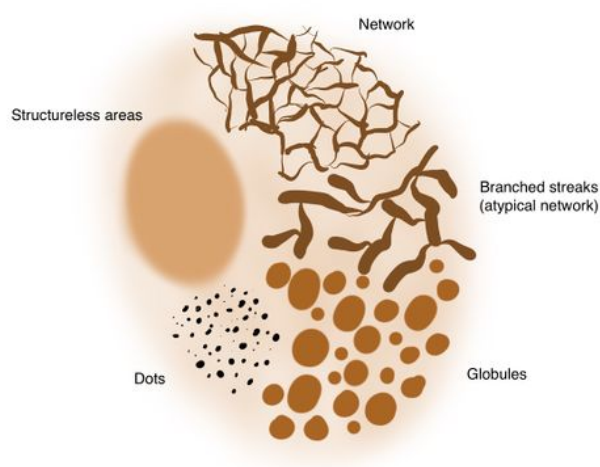
Εικόνα 1.5: Αριστερά διακρίνουμε μία καλοήγη μελαγχρωματική βλάβη ενώ δεξιά μία κακοήγη.

- **C (Colour) – Χρώμα:** Το χρώμα της βλάβης δεν είναι χρωματικά ομοιόμορφο, παρουσιάζει πολυχρωμία. Σημαντικές θεωρούνται οι αποχρώσεις των χρωμάτων: λευκό, κόκκινο, καφέ, γκρι, μπλε και μαύρο.



Εικόνα 1.6: Αριστερά διακρίνουμε μία καλοήγη μελαγχρωματική βλάβη ενώ δεξιά μία κακοήγη.

- **D (Dermoscopic structures) - Δερματοσκοπικές δομές:** Η μελαγχρωματική βλάβη παρουσιάζει ανομοιομορφία ως προς τις δερματοσκοπικές δομές. Συγκεκριμένα εμφανίζει ταυτόχρονα περιοχές χωρίς δομή (Structureless areas), δίκτυα χρωστικής (Pigment networks), διακλαδούμενα δίκτυα (Branched streaks / atypical networks), σφαιρίδια (Globules) και τελείες (Dots) (βλέπε εικόνα 1.7).



Εικόνα 1.7: Πιθανές δερματοσκοπικές δομές στο εσωτερικό μιας μελαγχρωματικής βλάβης [19]

Στη συνέχεια, τα κριτήρια ABCD βαθμολογούνται (βλέπε πίνακα 1.1) και αφού εφαρμοστούν οι αντίστοιχοι συντελεστές βαρύτητας προκύπτει το Total Dermoscopy Score (TDS):  $TDS = 1,3A + 0,1B + 0,5C + 0,5D$ . Η διάγνωση της βλάβης γίνεται με βάση την τιμή του TDS ως εξής:

- Αν  $TDS < 4,75$  η μελαγχρωματική βλάβη διαγιγνώσκεται ως καλοήθης.
- Αν  $4,75 \leq TDS \leq 5,45$  η μελαγχρωματική βλάβη είναι ύποπτη και χρήζει στενής ιατρικής παρακολούθησης.
- Αν  $TDS > 5,45$  κρίνεται απαραίτητη η χειρουργική αφαίρεση και βιοψία της βλάβης.

Κριτήριο	Βαθμός	Συντελεστής Βαρύτητας
Ασυμμετρία	0-2	x1,3
Όρια βλάβης	0-8	x0,1
Χρώμα	1-6	x0,5
Δερματοσκοπικές δομές	1-5	x0,5

Πίνακας 1.1: Βαθμολόγηση ABCD κριτηρίων

### 1.5.3 Δερματοσκόπηση

Η δερματοσκόπηση είναι μια μη επεμβατική μέθοδος διάγνωσης δερματικών παθήσεων. Με αυτήν την ανώδυνη διαγνωστική τεχνική καθίσταται δυνατή η παρατήρηση των μορφολογικών χαρακτηριστικών της επιδερμίδας που δεν φαίνονται δια γυμνού οφθαλμού. Για τη λήψη δερματοσκοπικών εικόνων χρησιμοποιούνται ειδικά ψηφιακά συστήματα. Τα

επιμέρους όργανα των συστημάτων δερματοσκόπησης είναι ένα ειδικό μικροσκόπιο (δερματοσκόπιο) προσαρμοσμένο σε μια ψηφιακή φωτογραφική μηχανή που με την σειρά της είναι συνδεδεμένη σε έναν ηλεκτρονικό υπολογιστή (βλέπε εικόνα 1.8). Τα δερματοσκοπικά ψηφιακά συστήματα δίνουν τη δυνατότητα μεγέθυνσης της δομής του δέρματος κατά τουλάχιστον δέκα φορές ενώ παράλληλα επιτρέπουν τη λήψη εικόνων υψηλής ανάλυσης. Η δερματοσκόπηση πλέον είναι ένα απαραίτητο εργαλείο για την αξιολόγηση μελαγχρωματικών βλαβών. Βοηθάει στην παρατήρηση των ABCD χαρακτηριστικών και εν συνεχεία στην ασφαλέστερη διάγνωση μελανώματος ή καλοήθων βλαβών.



Εικόνα 1.8: Δερματοσκόπηση [20]



**Μέρος I**

**Θεωρητικό Μέρος**

---



## Κεφάλαιο 2

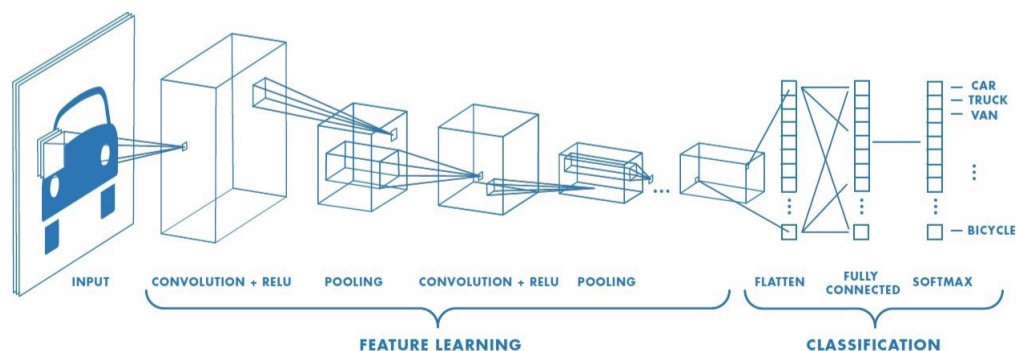
# Συνελικτικά Νευρωνικά Δίκτυα - Convolutional Neural Networks (CNN)

---

Η ιστορία των σύγχρονων συνελικτικών νευρωνικών δικτύων ξεκινάει την δεκαετία του 1990 όταν οι LeCun et al. με το paper “Gradient-Based Learning Applied to Document Recognition” [21] απέδειξαν ότι ένα CNN μπορεί να αξιοποιηθεί για την αναγνώριση χειρόγραφων χαρακτήρων. Συγκεκριμένα, χρησιμοποίησαν την βάση δεδομένων MNIST [22] για την εκπαίδευσή του CNN, ενός πλέον ιδιαίτερα διαδεδομένου dataset που περιέχει εικόνες χειρογράφων ψηφίων. Τα state-of-the-art CNN μοντέλα πλέον επιδεικνύουν σχεδόν τέλεια ακρίβεια στην ταξινόμηση του MNIST. Σε εφαρμογές όρασης υπολογιστών (computer vision) και συγκεκριμένα στην ταξινόμηση εικόνων, τα CNN έρχονται να λύσουν δύο βασικά προβλήματα που αντιμετωπίζουν τα κλασικά νευρωνικά δίκτυα (Artificial Neural Networks-ANN):

Τα ANN για την ταξινόμηση εικόνων μετατρέπουν την είσοδο σε ένα μονοδιάστατο διάνυσμα χάνοντας έτσι τις πολύτιμες χωρικές συσχετίσεις (spatial correlations) που εμπεριέχει μία εικόνα. Απεναντίας, τα CNN, όπως θα δούμε στη συνέχεια, διατηρούν αυτές τις πληροφορίες και εξάγουν χαρακτηριστικά με τη χρήση φίλτρων που ολισθαίνουν στην εικόνα-είσοδο.

Τα ANN αποτελούνται από πλήρως συνδεδεμένα επίπεδα στα οποία κάθε νευρώνας του δικτύου συνδέεται με όλους τους νευρώνες του προηγούμενου επιπέδου. Αυτό έχει ως αποτέλεσμα τα ANN να μην μπορούν να ανταπεξέλθουν σε προβλήματα ταξινόμησης εικόνων λόγω του προβλήματος της κλιμάκωσης. Αν μία έγχρωμη εικόνα ανάλυσης  $64 \times 64 \times 3$  τροφοδοτηθεί σε ένα ANN δίκτυο, μόνο για το πρώτο κρυφό επίπεδο χρειάζονται 12288 βάρη-παράμετροι για κάθε νευρώνα. Σε πραγματικές εφαρμογές η ανάλυση των εικόνων ξεπερνάει κατά πολύ αυτή του παραπάνω παραδείγματος με αποτέλεσμα τα ANN να καταλήγουν με έναν μη διαχειρίσιμο αριθμό παραμέτρων. Το γεγονός αυτό τα καθιστά πολύ αργά στην εκπαίδευση και ευάλωτα στο πρόβλημα υπερπροσαρμογής στα δεδομένα εκπαίδευσης. Αντιθέτως, τα συνελικτικά νευρωνικά δίκτυα αποτελούνται κυρίως από μερικώς συνδεδεμένα επίπεδα στα οποία κάθε νευρώνας συνδέεται με ένα μικρό αριθμό νευρώνων του προηγούμενου επιπέδου. Έτσι, από το πρώτο κιάλας επίπεδο οι διαστάσεις της εικόνας του παραδείγματος μπορούν να μειωθούν, μέσω της συνέλιξης, σε  $1 \times 1 \times 10$  χωρίς να χαθεί σημαντική πληροφορία. Επιπροσθέτως, στα CNN υφίσταται διαμοιρασμός παραμέτρων που όπως θα δούμε στη συνέχεια μειώνει περαιτέρω το πλήθος των παραμέτρων. Στις ενότητες που ακολουθούν θα αναλύσουμε τα βασικά επίπεδα ενός συνελικτικού νευρωνικού δικτύου.



Σχήμα 2.1: Η δομή ενός συνελκτικού νευρωνικού δικτύου [23]

## 2.1 Συνελκτικό επίπεδο (Convolutional layer)

Το συνελκτικό επίπεδο αποτελεί το σημαντικότερο δομικό στοιχείο των CNN. Περιλαμβάνει ένα σύνολο φίλτρων (kernels) τα οποία διατρέχουν την εικόνα - είσοδο του δικτύου και παράγουν ως έξοδο έναν χάρτη χαρακτηριστικών (feature map). Ο βασικός στόχος αυτού του επιπέδου είναι η εξαγωγή χαρακτηριστικών από την εικόνα.

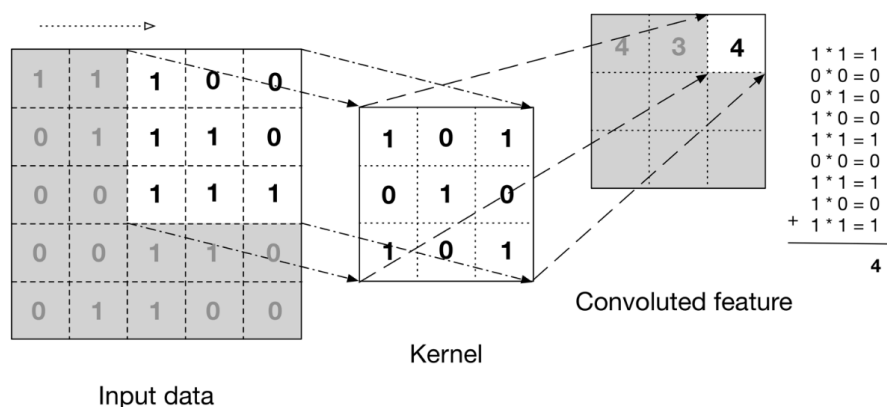
### 2.1.1 Φίλτρα

Κάθε φίλτρο ενός συνελκτικού επιπέδου αποτελεί έναν τετραγωνικό πίνακα με διαστάσεις  $H \times W \times D$ . Οι δύο πρώτες διαστάσεις Height, Width αφορούν τις χωρικές διαστάσεις του φίλτρου ενώ η τελευταία διάσταση Depth αποτελεί το βάθος του φίλτρου και είναι πάντα ίσο με το βάθος της εισόδου που λαμβάνει. Για παράδειγμα το επίπεδο που λαμβάνει ως είσοδο την ίδια την εικόνα διαθέτει φίλτρα βάθους ίσου με τρία. Επισημαίνουμε στο σημείο αυτό ότι μία έγχρωμη εικόνα έχει διαστάσεις  $H \times W \times 3$  δηλαδή κάθε εικονοστοιχείο (pixel) λαμβάνει μία τιμή για κάθε ένα από τα τρία βασικά χρώματα RGB (Red, Green, Blue - Κόκκινο, Πράσινο, Μπλε). Τα βάρη κάθε φίλτρου, δηλαδή οι τιμές του πίνακα, διαμορφώνονται κατά το στάδιο εκπαίδευσης του CNN.

### 2.1.2 Συνέλιξη

Κατά την συνέλιξη του φίλτρου με την είσοδο του συνελκτικού επιπέδου, το φίλτρο ολισθαίνει σε όλο το μήκος και το πλάτος της εισόδου. Σε κάθε ολίσθηση τα στοιχεία του φίλτρου πολλαπλασιάζονται ένα προς ένα με αυτά της εισόδου και στην συνέχεια αθροίζονται με το αποτέλεσμα να αποτελεί στοιχείο της εξόδου, δηλαδή του χάρτη χαρακτηριστικών (feature map) [24]. Η είσοδος της συνέλιξης μπορεί να είναι είτε τα δεδομένα εισόδου, στην περίπτωση του πρώτου συνελκτικού επιπέδου, είτε ο χάρτης χαρακτηριστικών που προήλθε από το προηγούμενο συνελκτικό επίπεδο. Διαισθητικά, η τιμή της εξόδου εκφράζει τον βαθμό στον οποίο το χαρακτηριστικό προς ανίχνευση εντοπίστηκε στην είσοδο.

Ουσιαστικά κάθε τιμή του χάρτη χαρακτηριστικών ισοδυναμεί με ένα νευρώνα του δικτύου ο οποίος συνδέεται μερικώς με τους νευρώνες του προηγούμενου επιπέδου ανάλογα με την διάσταση του φίλτρου (π.χ. στο παραπάνω παράδειγμα συνδέεται με  $3 \times 3 \times 1 = 9$  νευρώνες). Εφόσον όλα τα στοιχεία του χάρτη χαρακτηριστικών έχουν προκύψει με το ίδιο



Εικόνα 2.1: Παράδειγμα συνέλιξης [25]

φίλτρο, τα βάρη των συνδέσεων κάθε νευρώνα με το προηγούμενο επίπεδο θα είναι κοινά. Το φαινόμενο αυτό ονομάζεται διαμοιρασμός παραμέτρου (parameter sharing) και είναι ένα ακόμα προτέρημα των συνελκτικών νευρωνικών δικτύων καθώς συμβάλλει στη μείωση των παραμέτρων και της πολυπλοκότητας του δικτύου.

### 2.1.3 Υπερπαραμέτροι - Διαστάσεις εξόδου

Στο παράδειγμα που παρουσιάστηκε στην Εικόνα 2.1 το φίλτρο ολισθαίνει στην είσοδο με βήμα (stride) 1. Το βήμα αποτελεί υπερπαραμέτρο του δικτύου και δεν παίρνει απαραίτητα την τιμή 1. Η αύξηση του βήματος έχει ως αποτέλεσμα τη μείωση των χωρικών διαστάσεων της εξόδου.

Κατά την ολίσθηση του φίλτρου στην εικόνα, τα εικονοστοιχεία που βρίσκονται στα άκρα της εικόνας (π.χ. το πρώτο στο πάνω αριστερό μέρος της εικόνας), συνεισφέρουν μόλις μία φορά στον υπολογισμό του χάρτη χαρακτηριστικών. Αυτό μπορεί να οδηγήσει στην απώλεια πολύτιμης πληροφορίας και αντιμετωπίζεται με την εισαγωγή padding, δηλαδή με την προσθήκη ενός αριθμού επιπλέον εικονοστοιχείων, που συνήθως λαμβάνουν τιμή 0 (zero padding). Η προσθήκη padding αυξάνει τον αριθμό των ολισθήσεων του φίλτρου και κατά συνέπεια τις διαστάσεις του χάρτη χαρακτηριστικών.

Στην πράξη, κάθε συνελκτικό επίπεδο περιλαμβάνει πολλά φίλτρα τα οποία παρότι αυξάνουν το υπολογιστικό κόστος του δικτύου συμβάλλουν στην εξαγωγή περισσότερων χαρακτηριστικών. Ο αριθμός των φίλτρων είναι υπερπαραμέτρος του δικτύου και καθορίζει το βάθος της εξόδου.

Συναρτήσει όλων των παραπάνω υπερπαραμέτρων, οι διαστάσεις της εξόδου  $W' \times H' \times D'$  ενός συνελκτικού επιπέδου λαμβάνονται από τους παρακάτω τύπους. Για είσοδο διαστάσεων  $W_1 \times H_1 \times D_1$  και εφαρμόζοντας  $K$  φίλτρα διαστάσεων  $W_2 \times H_2 \times D_2$  με stride  $S$  και padding  $P$  λαμβάνουμε:

$$W' = \frac{W_1 - W_2 + 2P}{S} + 1$$

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

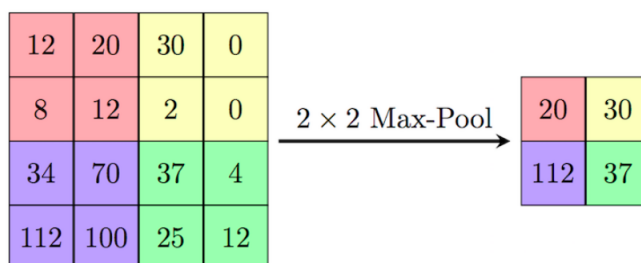
Εικόνα 2.2: Παράδειγμα εφαρμογής Zero Padding με  $P = 1$  [26]

$$H' = \frac{H_1 - H_2 + 2P}{S} + 1$$

$$D' = K$$

## 2.2 Επίπεδο Συγκέντρωσης (Pooling Layer)

Το επίπεδο συγκέντρωσης έπεται του συνελκτικού επιπέδου και εξυπηρετεί στην μείωση των χωρικών διαστάσεων του χάρτη χαρακτηριστικών. Ουσιαστικά, η εφαρμογή συγκέντρωσης (pooling) υποδειγματοληπτει την είσοδο του επιπέδου. Μέσω της διαδικασίας αυτής μειώνονται οι παράμετροι και η πολυπλοκότητα του νευρωνικού δικτύου και έτσι τίθεται υπό έλεγχο το πρόβλημα της υπερπροσαρμογής το οποίο θα αναλύσουμε σε επόμενο κεφάλαιο. Στην εικόνα 2.3 παρουσιάζεται η εφαρμογή max pooling. Όπως παρατηρούμε ένα παράθυρο διαστάσεων  $2 \times 2$  ολισθαίνει με βήμα 2 στον χάρτη χαρακτηριστικών και σε κάθε βήμα τροφοδοτεί την έξοδο με το μεγαλύτερο στοιχείο. Αποτέλεσμα της παραπάνω διαδικασίας είναι ο υποδιπλασιασμός των διαστάσεων του αρχικού πίνακα. Εναλλακτικά, μπορεί να χρησιμοποιηθεί ως συνάρτηση συγκέντρωσης (pooling function) η average που όπως υποδηλώνει και το όνομά της, λαμβάνει τον μέσο όρο των στοιχείων του παραθύρου.



Εικόνα 2.3: Παράδειγμα εφαρμογής Max Pooling [27]

Όπως και στο συνελκτικό επίπεδο, πρέπει να προσδιορίσουμε τις διαστάσεις της περιοχής προς συγκέντρωση καθώς και το βήμα ολίσθησης. Οι διαστάσεις της εξόδου  $W' \times H' \times D'$

ενός επιπέδου συγκέντρωσης λαμβάνονται από τους παρακάτω τύπους. Για είσοδο διαστάσεων  $W_1 \times H_1 \times D_1$  και εφαρμόζοντας pooling με παράθυρο  $W_2 \times H_2 \times D_2$ , με stride  $S$  παίρνουμε:

$$W' = \frac{W_1 - W_2}{S} + 1$$

$$H' = \frac{H_1 - H_2}{S} + 1$$

$$D' = K$$

## 2.3 Πλήρως Συνδεδεμένο Επίπεδο (Fully Connected Layer)

Το πλήρως συνδεδεμένο επίπεδο ενώ αποτελεί ένα τυπικό επίπεδο στα κλασικά νευρωνικά δίκτυα (ANNs), στα συνελκτικά δίκτυα τοποθετείται στα τελευταία επίπεδα της αρχιτεκτονικής. Κάθε νευρώνας του συνδέεται πλήρως με όλους τους νευρώνες του προηγούμενου επιπέδου. Ο στόχος του πλήρως συνδεδεμένου επιπέδου είναι η ταξινόμηση των δειγμάτων στην κατάλληλη κλάση με βάση τα χαρακτηριστικά που έχουν εξάγει τα προηγούμενα επίπεδα.





## Κεφάλαιο **3**

# Εκπαίδευση CNN

---

### 3.1 Συναρτήσεις Κόστους (Cost Functions)

Όπως είδαμε στην προηγούμενη ενότητα το πλήρως συνδεδεμένο επίπεδο λειτουργεί ως ταξινομητής των δειγμάτων στην κατάλληλη κλάση. Το επίπεδο αυτό χρησιμοποιεί μια συνάρτηση κόστους προκειμένου να εκτιμήσει την ποιότητα των προβλέψεων του δικτύου στα δεδομένα εκπαίδευσης. Πρακτικά, η συνάρτηση κόστους ποσοτικοποιεί τη διαφορά μεταξύ της εξόδου που προέβλεψε το δίκτυο και της επιθυμητής εξόδου. Στο στάδιο εκπαίδευσης του CNN επιχειρούμε την ελαχιστοποίηση της. Η επιλογή της κατάλληλης συνάρτησης κόστους είναι καθοριστική για την βελτιστοποίηση του δικτύου. Στην συνέχεια θα εξετάσουμε μερικές από τις δημοφιλέστερες συναρτήσεις κόστους συνελκτικών νευρωνικών δικτύων.

#### 3.1.1 Μέσο τετραγωνικό σφάλμα (Mean squared error)

Η MSE χρησιμοποιείται κατά κύριο λόγο σε regression προβλήματα. Υπολογίζει τον μέσο όρο των τετραγώνων της διαφοράς μεταξύ της πρόβλεψης του δικτύου και της επιθυμητής εξόδου.

$$J(w, b) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

Όπου:

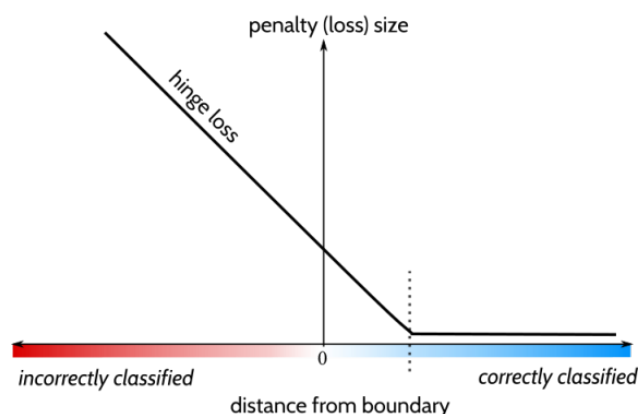
- $J$  η συνάρτηση κόστους
- $w$  (weights) τα βάρη του νευρωνικού δικτύου
- $b$  (biases) οι πολώσεις των νευρώνων
- $n$  ο συνολικός αριθμός δειγμάτων
- $\hat{y}$  η έξοδος του νευρωνικού δικτύου
- $i$  το  $i$ -οστό δείγμα

#### 3.1.2 Hinge Loss

Η συνάρτηση κόστους hinge loss χρησιμοποιείται σε προβλήματα κατηγοριοποίησης. Η hinge loss ορίζει μια τιμή  $m > 0$  ως κατώφλι και εφαρμόζει ποινές ως εξής: Αν το δίκτυο

ταξινομήσει λάθος ένα δείγμα ή το ταξινομήσει σωστά με τιμή μικρότερη του  $m$  θα λάβει ανάλογη ποινή (βλέπε σχήμα 3.1). Διαισθητικά, η hinge loss δεν αρκείται στις σωστές προβλέψεις αλλά απαιτεί από το δίκτυο και ένα επίπεδο "βεβαιότητας".

$$J(w, b) = \sum_{i=1}^n (0, m - (2y_i - 1)\hat{y}_i)$$



Σχήμα 3.1: Η γραφική παράσταση της Hinge Loss [28]. Το όνομα της συνάρτησης προήλθε από την ομοιότητα της γραφική της παράστασης με "μεντεσέ" (hinge)

### 3.1.3 Cross entropy loss

Η συνάρτηση cross entropy loss υπολογίζει την ποιότητα των προβλέψεων ενός ταξινομητή όταν η έξοδος του αναπαριστά την πιθανότητα ενός δείγματος να ανήκει στην κλάση (για αυτόν το λόγο ονομάζεται και softmax loss). Η συνάρτηση αυξάνεται όσο η προβλεπόμενη πιθανότητα αποκλίνει από την πραγματική τιμή ως εξής:

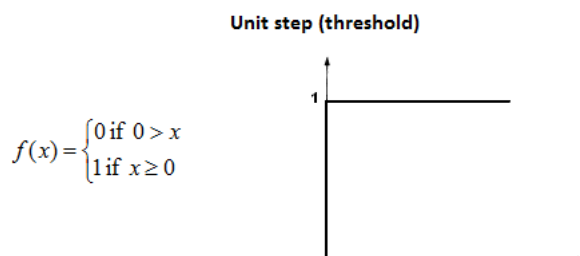
$$J(w, b) = - \sum_{i=1}^n y_i \log(\hat{y}_i)$$

## 3.2 Συναρτήσεις Ενεργοποίησης (Activation Functions)

Οι συναρτήσεις ενεργοποίησης, διαισθητικά, λειτουργούν ως μηχανισμοί επιλογής, καθώς από αυτές εξαρτάται το αν ένας νευρώνας θα "ενεργοποιηθεί". Όπως είδαμε στο μοντέλο του τεχνητού νευρώνα, οι συναρτήσεις ενεργοποίησης κανονικοποιούν το δυναμικό ενεργοποίησης σε μια πεπερασμένη τιμή που αποτελεί και την τελική έξοδο του νευρώνα. Η χρήση μη γραμμικών συναρτήσεων ενεργοποίησης είναι αυτή που επέτρεψε στα νευρωνικά δίκτυα να εισάγουν μη γραμμικές ιδιότητες και εν συνεχεία να εξαγάγουν συμπεράσματα υψηλού επιπέδου. Δίχως την ύπαρξη μη γραμμικών συναρτήσεων ενεργοποίησης τα δίκτυα θα περιορίζονταν στην υλοποίηση γραμμικών ταξινομητών. Στη συνέχεια θα εξετάσουμε τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης.

### 3.2.1 Βηματική (Step function)

Η βηματική συνάρτηση ενεργοποίησης χρησιμοποιεί μία τιμή ως κατώφλι (threshold). Αν η τιμή της εισόδου ξεπερνάει την τιμή κατωφλίου, η έξοδος λαμβάνει την τιμή 1 σηματοδοτώντας την ενεργοποίηση του νευρώνα, διαφορετικά λαμβάνει την τιμή 0.



Σχήμα 3.2: Η γραφική παράσταση της βηματικής συνάρτησης με μηδενική τιμή κατωφλίου.

### 3.2.2 Γραμμική (Linear function)

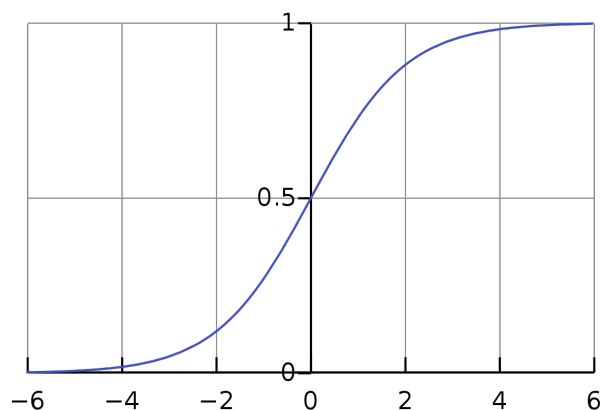
Η γραμμική συνάρτηση είναι της μορφής  $\varphi(x) = ax$  όπου  $a$  σταθερά, συνήθως ίση με τη μονάδα, και έτσι η έξοδος λαμβάνει τιμή ανάλογη της εισόδου. Σε αντίθεση με την βηματική συνάρτηση που επιστρέφει μόνο 0 ή 1, η γραμμική επιτρέπει την έξοδο οποιασδήποτε τιμής.

### 3.2.3 Σιγμοειδής (Sigmoid function)

Η σιγμοειδής συνάρτηση ενεργοποίησης κανονικοποιεί την είσοδο στο εύρος  $[0, 1]$ . Εκφράζεται ως εξής:

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

Το όνομα της προήλθε από την ομοιότητα της γραφικής της παράστασης με το τελικό σίγμα “s”:



Σχήμα 3.3: Η γραφική παράσταση της σιγμοειδούς συνάρτησης.

### 3.2.4 Softmax

Η συνάρτηση ενεργοποίησης softmax πρόκειται για επέκταση της sigmoid με κύρια εφαρμογή τα προβλήματα ταξινόμησης πολλαπλών κλάσεων.

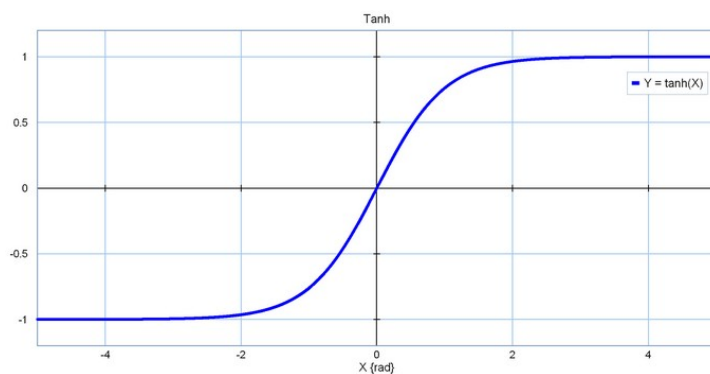
$$\varphi(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Κατά την αντιμετώπιση ενός προβλήματος ταξινόμησης το επίπεδο εξόδου αποτελείται από τόσους νευρώνες όσες και οι κλάσεις του προβλήματος. Για παράδειγμα, σε ένα πρόβλημα με τρεις κλάσεις το επίπεδο εξόδου διαθέτει τρεις νευρώνες. Αν από αυτό το επίπεδο λαμβάναμε τις τιμές  $[1.2, 0.9, 0.75]$  και στη συνέχεια εφαρμόζαμε τη softmax θα παίρναμε το τελικό αποτέλεσμα  $[0.42, 0.31, 0.27]$ . Αυτές οι τιμές εκφράζουν την πιθανότητα του δείγματος-εισόδου να ανήκει σε καθεμία από τις τρεις κλάσεις. Παρατηρούμε ότι το άθροισμά τους ισούται με τη μονάδα.

### 3.2.5 Υπερβολική Εφαπτομένη (Tanh function)

Υλοποιεί την συνάρτηση υπερβολικής εφαπτομένης ώστε να περιορίσει την είσοδο στο εύρος  $[-1, 1]$ . Δίνεται από τη σχέση:

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



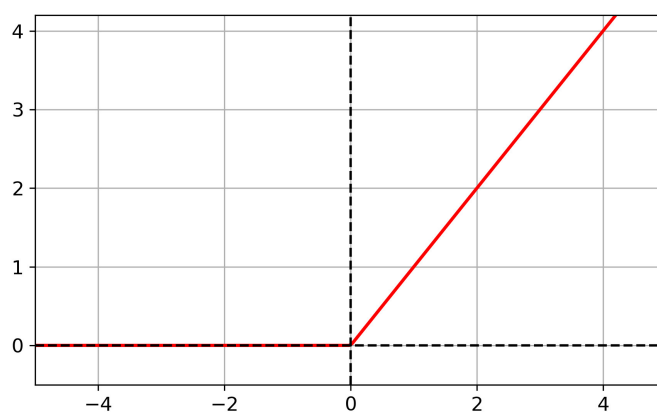
Σχήμα 3.4: Η γραφική παράσταση της υπερβολικής εφαπτομένης

### 3.2.6 Ανορθωμένη Γραμμική Μονάδα (ReLU)

Η έξοδος της ReLU ισούται με το μηδέν όταν  $x < 0$ , διαφορετικά παίρνει την τιμή της εισόδου:

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

Χάρη στην απλότητα και τη μειωμένη υπολογιστική πολυπλοκότητα της, η ReLU είναι ιδιαίτερα διαδεδομένη και μάλιστα καθιερωμένη σε προβλήματα βαθιάς μάθησης (deep learning).

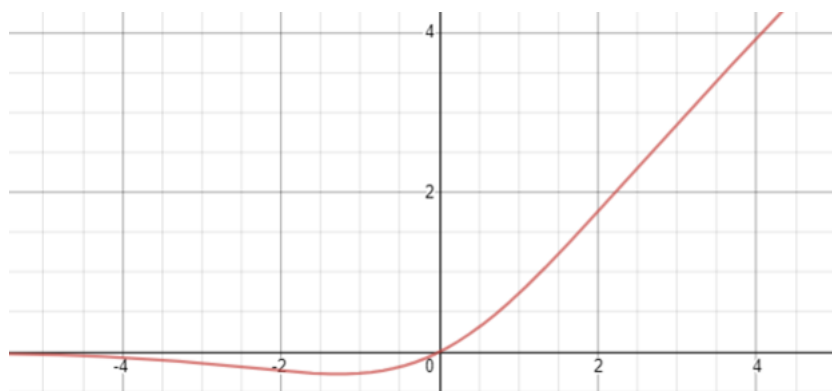


Σχήμα 3.5: Η γραφική παράσταση της ReLU

### 3.2.7 Swish

Μία εξίσου υπολογιστικά αποδοτική συνάρτηση ενεργοποίησης είναι η Swish που ουσιαστικά πρόκειται για παραλλαγή της ReLU και επιτυγχάνει ακόμα καλύτερη απόδοση σε βαθιά νευρωνικά δίκτυα. Η Swish ορίζεται ως εξής:

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



Σχήμα 3.6: Η γραφική παράσταση της Swish

### 3.3 Κανονικοποίηση (Regularization)

Ο μεγάλος αριθμός παραμέτρων των βαθιών νευρωνικών δικτύων τα καθιστά ευάλωτα στο πρόβλημα υπερπροσαρμογής στα δεδομένα εκπαίδευσης (overfitting). Με τον όρο υπερπροσαρμογή αναφερόμαστε στην διαδικασία κατά την οποία ένα μοντέλο ενώ αποδίδει καλά στα δεδομένα εκπαίδευσης αποτυγχάνει στη γενίκευση, δηλαδή στο να αποδώσει εξίσου καλά σε νέα δεδομένα. Η αντιμετώπιση αυτού του φαινομένου γίνεται με μεθόδους κανονικοποίησης μερικές εκ των οποίων θα εξετάσουμε στη συνέχεια:

#### 3.3.1 Προσαύξηση δεδομένων (Data augmentation)

Η προσαύξηση δεδομένων είναι μια από τις πιο απλές μεθόδους βελτίωσης της ικανότητας γενίκευσης των CNN. Η μέθοδος αυτή χρησιμοποιεί ως βάση μια εικόνα του συνόλου δεδομένων και με διάφορες τεχνικές όπως η περιστροφή, η περικοπή κ.ά. συνθέτει νέες εικόνες [29]. Η προσθήκη νέων δειγμάτων στο σύνολο δεδομένων εκπαίδευσης, ειδικά όταν αυτό είναι περιορισμένο, συμβάλλει στην αποφυγή της υπερπροσαρμογής του μοντέλου.

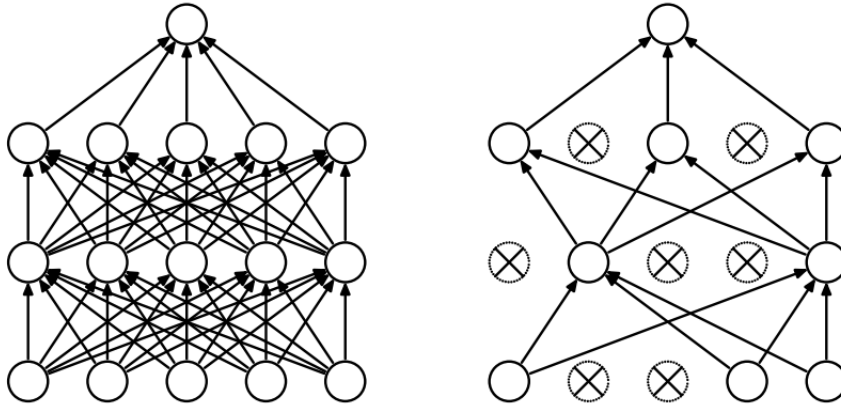
#### 3.3.2 Μέσος όρος μοντέλων (Ensemble model averaging)

Ο μέσος όρος μοντέλων αποτελεί μια ιδιαίτερα αποτελεσματική μέθοδο κανονικοποίησης στην οποία επιστρατεύονται περισσότερα του ενός μοντέλα. Κάθε μοντέλο έχει διαφορετικές παραμέτρους είτε λόγω της διαφορετικής αρχιτεκτονικής του, είτε των διαφορετικών υπερπαραμέτρων του. Η έξοδος του συστήματος λαμβάνεται ως συνδιασμός των επιμέρους εξόδων των μοντέλων [30]. Συγκεκριμένα μπορεί να προκύψει με την χρήση συντελεστών βαρύτητας, με την λήψη του μέσου όρου των εξόδων κ.ά.. Έτσι το τελικό μοντέλο είναι λιγότερο επιρρεπές στο πρόβλημα της υπερπροσαρμογής και πετυχαίνει καλύτερη απόδοση σε σύγκριση με κάθε μοντέλο ξεχωριστά.

#### 3.3.3 Dropout

Το dropout αποτελεί έναν πολύ απλό και ταυτόχρονα εξαιρετικά αποτελεσματικό μηχανισμό κανονικοποίησης. Ενεργεί σε επίπεδα του δικτύου κατά την διάρκεια της εκπαίδευσης απενεργοποιώντας νευρώνες και όλες τις εισερχόμενες-εξερχόμενες συνδέσεις τους [31]. Οι

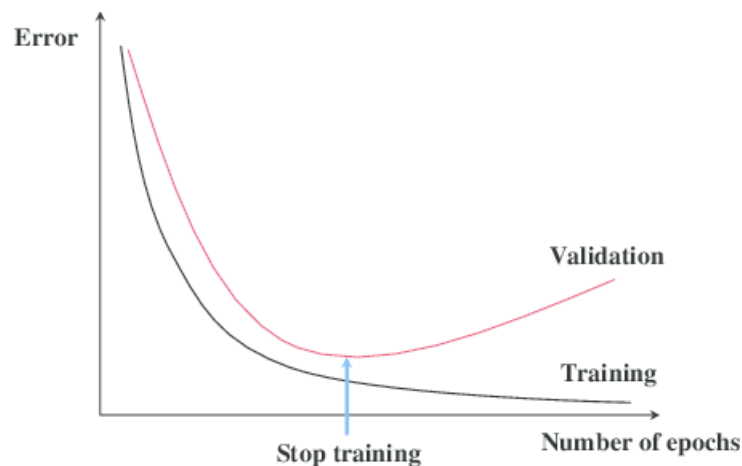
νευρώνες προς απενεργοποίηση επιλέγονται τυχαία με πιθανότητα  $1 - p$ , με το  $p$  να αποτελεί υπερπαράμετρο του δικτύου. Διαισθητικά, με την χρήση της μεθόδου dropout το μοντέλο απλοποιείται στο στάδιο της εκπαίδευσης καθώς χρησιμοποιείται ένα υποδίκτυο με λιγότερους νευρώνες. Έτσι το σύστημα παύει να προσαρμόζει πλήρως τις παραμέτρους στα δεδομένα εκπαίδευσης και γενικεύει καλύτερα.



Σχήμα 3.7: Αριστερά απεικονίζεται το δίκτυο στην αρχική του μορφή ενώ δεξιά απεικονίζεται κατόπιν της εφαρμογής Dropout [31]

### 3.3.4 Πρόωρη διακοπή εκπαίδευσης (Early stopping)

Η μέθοδος κανονικοποίησης early stopping εφαρμόζεται για την αντιμετώπιση της υπερπροσαρμογής σε αλγόριθμους βελτιστοποίησης που βασίζονται στον gradient descent. Πρακτικά ρυθμίζει την διάρκεια της εκπαίδευσης με κριτήριο την αποφυγή της υπερπροσαρμογής. Αν η μείωση του σφάλματος στο train set συνοδεύεται από την μείωση του σφάλματος στο validation set το μοντέλο μπορεί να συνεχίσει να εκπαιδεύεται. Αν όμως παρατηρηθεί μείωση της ικανότητας γενίκευσης, δηλαδή αύξηση του σφάλματος στο validation set, η εκπαίδευση του δικτύου σταματάει πρόωρα [32].

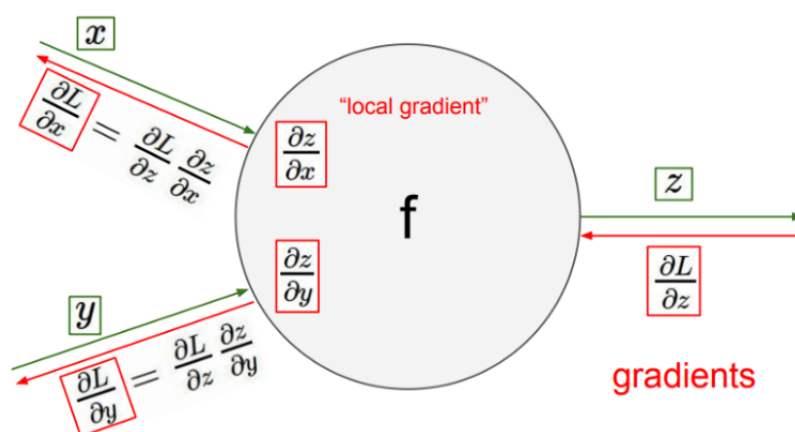


Σχήμα 3.8: Early stopping [33]

## 3.4 Αλγόριθμοι Βελτιστοποίησης (Optimizers)

### 3.4.1 Backpropagation

Ένα κοινό χαρακτηριστικό όλων των αλγορίθμων βελτιστοποίησης είναι ο υπολογισμός των πρώτων παραγωγών της συνάρτησης κόστους ως προς τα βάρη των νευρώνων του επιπέδου εξόδου. Στην συνέχεια με την χρήση του κανόνα της αλυσίδας οι παράγωγοι αυτοί διαδίδονται στα προηγούμενα επίπεδα του δικτύου προκειμένου να επέλθει η προσαρμογή των βαρών και εν συνεχεία η μείωση της συνάρτησης κόστους  $J$ . Υπεύθυνος για αυτήν την προς τα πίσω διάδοση της παραγωγού είναι ο αλγόριθμος backpropagation. Ο τρόπος με τον οποίο η παράγωγος αυτή θα αξιοποιηθεί για την προσαρμογή των βαρών του δικτύου είναι ευθύνη του αλγόριθμου βελτιστοποίησης, ο οποίος δεν πρέπει να συγχέεται με τον αλγόριθμο backpropagation.



Εικόνα 3.1: Backpropagation [34]

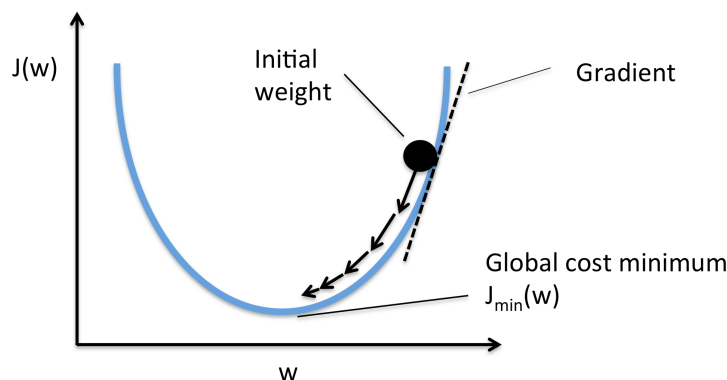
Στην Εικόνα 3.1 απεικονίζεται ο αλγόριθμος backpropagation από την οπτική ενός νευρώνα. Εδώ,  $L$  είναι η συνάρτηση κόστους που προέκυψε από το forward propagation. Η κλήση (gradient)  $\frac{\partial L}{\partial z}$  λαμβάνεται από ένα νευρώνα που βρίσκεται σε επόμενο επίπεδο. Οι τοπικές κλήσεις (local gradients)  $\frac{\partial z}{\partial x}$ ,  $\frac{\partial z}{\partial y}$  η έξοδος  $z$  και οι εισοδοί  $x$ ,  $y$  κάθε νευρώνα πρέπει να αποθηκευτούν για τον υπολογισμό των κλήσεων που θα παραλάβουν οι νευρώνες των προηγούμενων επιπέδων.

### 3.4.2 Gradient Descent

Το ζητούμενο κατά την εκπαίδευση των νευρωνικών δικτύων είναι η εύρεση των κατάλληλων βαρών και πολώσεων για την ελαχιστοποίηση της συνάρτησης κόστους  $J(w, b)$ . Πρόκειται δηλαδή για ένα πρόβλημα εύρεσης ελάχιστου. Ξεκινώντας από ένα σημείο της  $J$ , η κλήση (gradient) της συνάρτησης στο σημείο μας κατευθύνει προς μεγαλύτερες τιμές της  $J$ . Αν λοιπόν εμείς κάνουμε ένα βήμα προς την αντίθετη κατεύθυνση θα οδηγηθούμε σε μικρότερες τιμές της συνάρτησης  $J$ , βλέπε σχήμα 3.9. Το μέγεθος του βήματος καθορίζεται από την υπερπαραμέτρο "ρυθμός εκπαίδευσης"-η.

Ο αλγόριθμος gradient descent επαναλαμβάνει την παραπάνω διαδικασία μέχρι να εντοπίσει ελάχιστο. Η ανανέωση των παραμέτρων σε κάθε επανάληψη εκφράζεται μαθηματικά





Σχήμα 3.9: Gradient Descent [35]

από την παρακάτω εξίσωση :

$$w_t = w_{t-1} - \eta \nabla_w J(w_t)$$

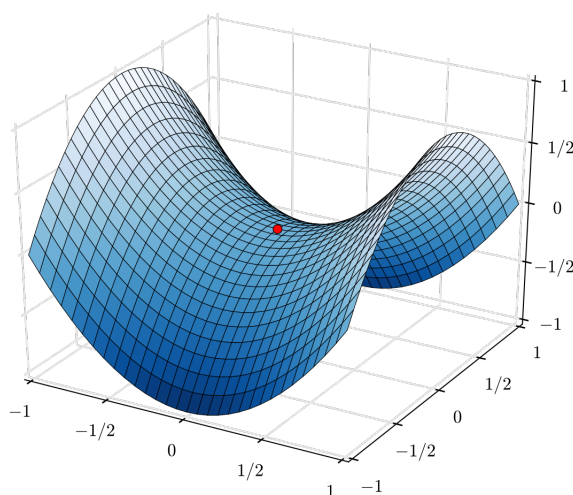
Υπάρχουν τρεις παραλλαγές του αλγόριθμου gradient descent με ειδοποιός διαφορά τον αριθμό των δειγμάτων που χρειάζονται για τον υπολογισμό των μερικών παραγωγών :

- Batch Gradient Descent:** Η πρώτη υλοποίηση του gradient descent, που ονομάστηκε Batch Gradient Descent, υπολογίζει τις μερικές παραγωγούς της συνάρτησης κόστους  $J$ , ως προς τις παραμέτρους του δικτύου, για ολόκληρο το training set. Η ανάγκη υπολογισμού των παραγωγών σε όλο το σύνολο των δεδομένων εκπαίδευσης για την λήψη ενός μόνο βήματος καθιστά τον αλγόριθμο αργό και δύσχρηστο. Στην περίπτωση μάλιστα που το σύνολο δεδομένων εκπαίδευσης υπερβαίνει σε χωρητικότητα την διαθέσιμη μνήμη, ο αλγόριθμος δεν μπορεί να "τρέξει". Όταν η επιφάνεια της συνάρτησης κόστους  $J$  είναι κυρτή (convex function) ο Batch Gradient Descent θα φτάσει αποδεδειγμένα σε ολικό ελάχιστο ωστόσο σε προβλήματα που εμφανίζουν τοπικά ελάχιστα είναι ευάλωτος στο να εγκλωβιστεί.
- Stochastic Gradient Descent:** Ο αλγόριθμος Stochastic Gradient Descent οδηγείται στο άλλο άκρο καθώς ενημερώνει τις παραμέτρους του δικτύου, δηλαδή λαμβάνει ένα βήμα, για κάθε δείγμα (sample) του training set. Αυτή η προσέγγιση έρχεται να λύσει το πρόβλημα των περιττών υπολογισμών καθώς παρακάμπτει τον υπολογισμό "παρόμοιων" παραγωγών πετυχαίνοντας έτσι την αύξηση της χρονικής αποδοτικότητας του δικτύου. Το γεγονός αυτό καθιστά τον αλγόριθμο καταλληλότερο για συστήματα που καλούνται να διαχειριστούν μεγάλο όγκο δεδομένων. Οι συνεχείς ενημερώσεις προκαλούν μεγαλύτερη διακύμανση στις τιμές της συνάρτησης κόστους ενώ μπορεί και να οδηγήσουν σε αστάθεια τη διαδικασία εκπαίδευσης μέσω ακραίων δεδομένων εισόδου (outliers). Παραδόξως, αυτό το φαινόμενο αποτελεί πλεονέκτημα του αλγόριθμου καθώς του επιτρέπει να απεγκλωβιστεί από τα τοπικά ελάχιστα μιας μη κυρτής επιφάνειας και εν συνεχεία να βελτιώσει την απόδοση του δικτύου σε σύγκριση με τον Batch Gradient Descent.

- **Mini-batch Gradient Descent:** Ο Mini-batch Gradient Descent αποτελεί τη χρυσή τομή μεταξύ των δύο παραπάνω προσεγγίσεων καθώς χωρίζει τα δεδομένα εκπαίδευσης σε δέσμες (batches) και ενημερώνει τις παραμέτρους του δικτύου για κάθε δέση ξεχωριστά. Αυτή η προσέγγιση ουσιαστικά συνδυάζει τα πλεονεκτήματα των προαναφερθέντων αλγορίθμων γιατί περιορίζει αισθητά την ανεπιθύμητη διακύμανση των ενημερώσεων διατηρώντας την ικανότητα εύρεσης καλύτερων τοπικών ελαχίστων σε μη κυρτές επιφάνειες.

### 3.4.3 Momentum

Ένα βασικό πρόβλημα του αλγόριθμου Stochastic Gradient Descent είναι οι περιοχές που η επιφάνεια παρουσιάζει τοπικό ελάχιστο σε μία διάσταση και παράλληλα τοπικό μέγιστο σε μία άλλη διάσταση (saddle points ή minimax points).



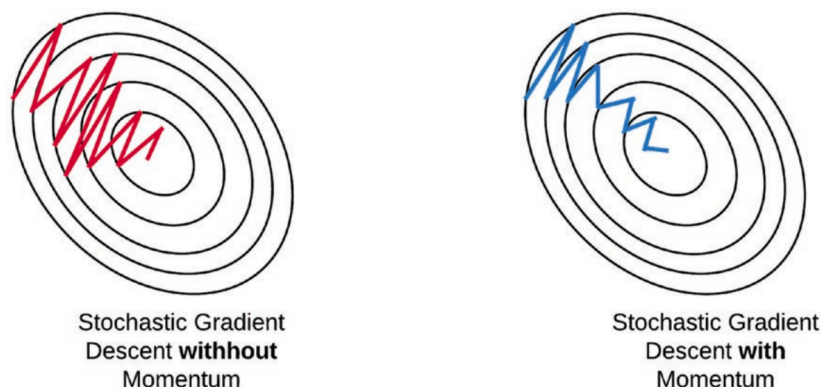
Σχήμα 3.10: *Saddle Point* [36]

Σε αυτές τις περιπτώσεις ο SGD ταλαντώνεται προς τη λάθος διάσταση με αποτέλεσμα να έχει πολύ αργή σύγκλιση. Η ορμή (momentum) αποτελεί μία μέθοδο επιτάχυνσης του αλγόριθμου SGD προς τη σωστή κατεύθυνση, μειώνοντας τις ταλαντώσεις, όπως φαίνεται στην εικόνα 3.2. Αυτό επιτυγχάνεται με την πρόσθεση της κλήσης της προηγούμενης επανάληψης ( $u_{t-1}$ ) σταθμισμένης με έναν παράγοντα  $\beta$  στην εξίσωση ανανέωσης παραμέτρων:

$$u_t = \eta \nabla_w J(w_t) + \beta u_{t-1}$$

$$w_t = w_{t-1} - u_t$$

Διαισθητικά, για την καλύτερη κατανόηση του αλγόριθμου Momentum, χρησιμοποιείται η εξής αναλογία: Όταν αφήνουμε μία μπάλα να κυλήσει από ένα λόφο, η ταχύτητα της, όπως και η ορμή της, συνεχώς αυξάνεται. Το ίδιο συμβαίνει και με τις ανανεώσεις των παραμέτρων του δικτύου: η ορμή αυξάνεται στις διαστάσεις των οποίων οι κλήσεις έχουν την ίδια κατεύθυνση ενώ μειώνεται στις διαστάσεις των οποίων οι κλήσεις αλλάζουν κατεύθυνση. Έτσι, το σύστημα συγκλίνει ταχύτερα μειώνοντας τις ανεπιθύμητες ταλαντώσεις.



Εικόνα 3.2: Momentum [37]

### 3.4.4 AdaGrad

Όπως προαναφέρθηκε, η υπερπαράμετρος “ρυθμός εκπαίδευσης” καθορίζει το μέγεθος του βήματος των αλγορίθμων gradient descent. Οι αλγόριθμοι βελτιστοποίησης που συναντήσαμε μέχρι στιγμής διατηρούν σταθερή την υπερπαράμετρο  $\eta$  για όλες τις παραμέτρους του δικτύου. Ο AdaGrad ανατρέπει αυτήν την προσέγγιση καθώς τροποποιεί το  $\eta$  για κάθε παράμετρο ξεχωριστά ανάλογα με τη συχνότητα εμφάνισής της. Σε “σπάνιες” παραμέτρους ο ρυθμός εκπαίδευσης αυξάνεται εφαρμόζοντας μεγαλύτερες ανανεώσεις ενώ σε “συχνές” παραμέτρους μειώνεται εφαρμόζοντας μικρότερες ανανεώσεις. Αυτό υλοποιείται με την διαίρεση του  $\eta$  κάθε παραμέτρου με το άθροισμα των τετραγώνων των παραγωγών των παραμέτρων μέχρι τη χρονική στιγμή  $t$ :

$$\delta_t = \nabla_w J(w_t)$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{\sum_{t=1}^t \delta_t^2 + \epsilon}} \delta_t$$

Το  $\epsilon$  είναι ένας παράγοντας εξομάλυνσης που συνήθως λαμβάνει τη τιμή  $1e - 8$  και αποτρέπει τη διαίρεση με το μηδέν.

Δεδομένου ότι κάθε όρος που προστίθεται στον παρονομαστή είναι θετικός, κατά την διάρκεια της εκπαίδευσης το άθροισμα αυξάνεται συνεχώς. Έτσι ο ρυθμός εκπαίδευσης οδηγείτε σε πολύ μικρά μεγέθη αποτρέποντας την περαιτέρω βελτίωση του συστήματος. Στη συνέχεια θα εξετάσουμε μερικούς αλγόριθμους που επιλύουν το παραπάνω πρόβλημα.

### 3.4.5 RMSprop

Όπως και ο AdaGrad, ο αλγόριθμος RMSprop προσαρμόζει το ρυθμό εκπαίδευσης κάθε παραμέτρου. Ωστόσο, σε αντίθεση με τον AdaGrad που αποθηκεύει τους παραγωγούς από την αρχή της εκπαίδευσης, ο RMSprop διατηρεί μόνο τους τελευταίους  $n$ . Αυτό επιτυγχάνεται μέσω ενός κυλιόμενου μέσου όρου (running average)  $E[\delta^2]_t$ :

$$E[\delta^2]_t = \gamma E[\delta^2]_{t-1} + (1 - \gamma) \delta_t^2$$

Ο παράγοντας  $\gamma$  συνήθως λαμβάνει την τιμή 0.9. Η παραπάνω συνάρτηση υλοποιεί τον εκθετικά φθίνοντα μέσο όρο των τετραγώνων των παραγώγων κάθε παραμέτρου. Η εξίσωση ανανέωσης παραμέτρων διαμορφώνεται ως εξής:

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[\delta^2]_t + \epsilon}} \delta_t$$

### 3.4.6 Adam

Ο Adam ακολουθεί την ίδια προσέγγιση με τον RMSprop συνδυάζοντας την με την χρήση του αλγόριθμου Momentum. Συγκεκριμένα εκτός από το να αποθηκεύει τον εκθετικά φθίνοντα μέσο όρο των τετραγώνων των παραγώγων  $E[\delta^2]_t$ , αποθηκεύει και τον εκθετικά φθίνοντα μέσο όρο των παραγώγων  $E[\delta]_t$ .

$$E[\delta]_t = \gamma E[\delta]_{t-1} + (1 - \gamma) \delta_t$$

$$E[\delta^2]_t = \gamma E[\delta^2]_{t-1} + (1 - \gamma) \delta_t^2$$

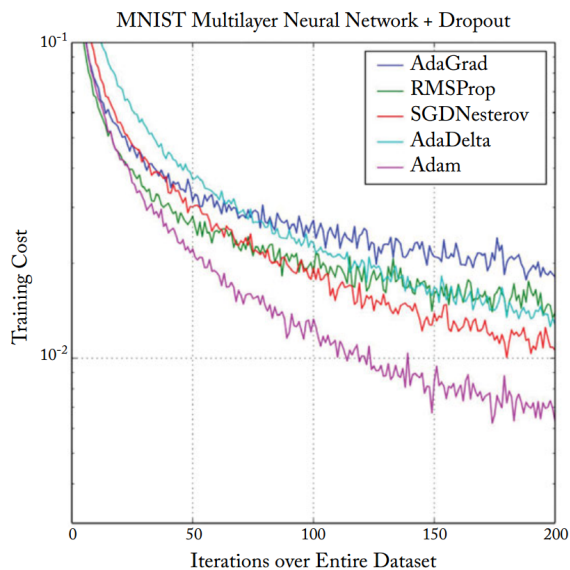
Δεδομένου ότι αρχικοποιούνται με μηδενικές τιμές μπορεί να παραμείνουν κοντά στο μηδέν ακόμα και μετά την πάροδο πολλών επαναλήψεων. Προκειμένου να αντιμετωπιστεί αυτό το πρόβλημα οι εκτιμήσεις των  $E[\delta]_t$ ,  $E[\delta^2]_t$  λαμβάνονται ως εξής:

$$\hat{E}[\delta]_t = \frac{E[\delta]_t}{1 - (\gamma_1)^t}$$

$$\hat{E}[\delta^2]_t = \frac{E[\delta^2]_t}{1 - (\gamma_2)^t}$$

Έτσι, καταλήγουμε με την παρακάτω εξίσωση ανανέωσης παραμέτρων:

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{\hat{E}[\delta^2]_t + \epsilon}} \hat{E}[\delta]_t$$



Σχήμα 3.11: Σύγκριση απόδοσης αλγορίθμων βελτιστοποίησης στο MNIST. [38]

## 3.5 Μετρικές (Metrics)

### 3.5.1 Confusion Matrix

Πρόκειται για έναν πίνακα που απεικονίζει την απόδοση ενός ταξινομητή (classifier). Κάθε γραμμή του confusion matrix εκφράζει τις προβλέψεις του νευρωνικού δικτύου ενώ κάθε στήλη εκφράζει τις πραγματικές τιμές των δειγμάτων (ή και αντίστροφα). Στην περίπτωση ενός δυαδικού ταξινομητή (binary classifier) ο πίνακας διαμορφώνεται ως εξής:

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	<b>TP</b> True Positive	<b>FP</b> False Positive
	negatives	<b>FN</b> False Negative	<b>TN</b> True Negative

Εικόνα 3.3: Confusion Matrix [39]

Όπου:

- **True Positive** (Αληθώς θετικά): ο αριθμός των δειγμάτων που ταξινομήθηκαν από το δίκτυο ως θετικά και πράγματι είναι θετικά.
- **False Positive** (Ψευδώς θετικά): ο αριθμός των δειγμάτων που ταξινομήθηκαν ως θετικά ενώ στην πραγματικότητα είναι αρνητικά. (Γνωστό και ως False Alarm)

- **False Negative** (Ψευδώς αρνητικά): ο αριθμός των δειγμάτων που ταξινομήθηκαν ως αρνητικά ενώ στην πραγματικότητα είναι θετικά.
- **True Negative** (Αληθώς αρνητικά): ο αριθμός των δειγμάτων που ταξινομήθηκαν ως αρνητικά και πράγματι είναι αρνητικά.

Συναρτήσσει των παραπάνω ορισμών θα αναλύσουμε τις μετρικές:

- **Accuracy:** Αποτελεί την πιο διαδεδομένη μετρική. Ισούται με το λόγο των ορθώς ταξινομημένων δειγμάτων προς τον συνολικό αριθμό δειγμάτων.

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

- **Recall / Sensitivity:** Ισούται με το λόγο των δειγμάτων που ορθώς ταξινομήθηκαν ως θετικά προς τον συνολικό αριθμό θετικών δειγμάτων.

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

- **Specificity:** Είναι ο λόγος των δειγμάτων που ορθώς ταξινομήθηκαν ως αρνητικά προς τον συνολικό αριθμό αρνητικών δειγμάτων.

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

- **Precision:** Ισούται με το λόγο των δειγμάτων που ορθώς ταξινομήθηκαν ως θετικά προς τον συνολικό αριθμό των δειγμάτων που ταξινομήθηκαν ως θετικά.

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

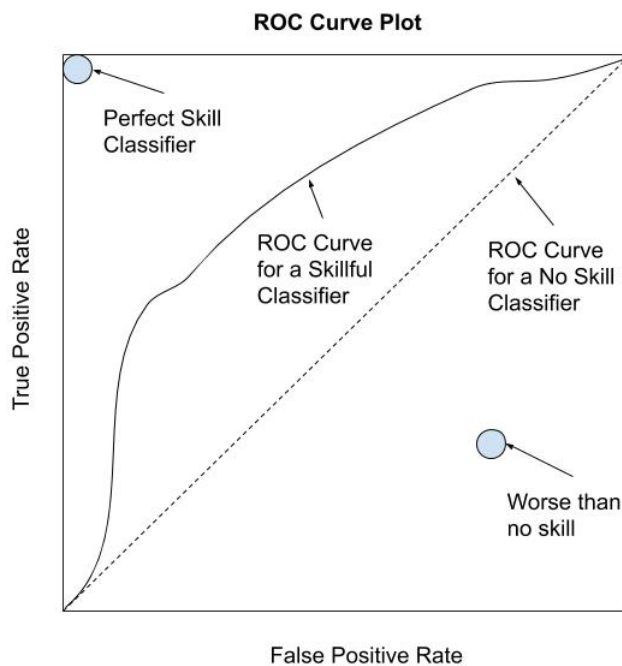
### 3.5.2 Καμπύλη ROC - AUC

Αν για λόγους απλότητας θεωρήσουμε ότι ένας δυαδικός ταξινομητής παράγει ως έξοδο την πιθανότητα ένα δείγμα να είναι θετικό  $p$  και συμπληρωματικά την πιθανότητα να είναι αρνητικό  $1 - p$ , τότε η κλάση στην οποία θα ταξινομηθεί το δείγμα καθορίζεται από την τιμή ενός κατωφλίου  $\tau$ :

- Αν  $p > \tau$  το δείγμα θα ταξινομηθεί ως θετικό.
- Αν  $p < \tau$  το δείγμα θα ταξινομηθεί ως αρνητικό.

Η καμπύλη ROC (Receiver Operating Characteristic) απεικονίζει την απόδοση του δυαδικού ταξινομητή καθώς η τιμή του κατωφλίου μεταβάλλεται:

Στον οριζόντιο άξονα  $x$  έχουμε το συμπλήρωμα του *Specificity* (False Positive Rate) ενώ στον κάθετο άξονα  $y$  έχουμε το *Sensitivity* (True Positive Rate). Κάθε σημείο της καμπύλης αναπαριστά το λόγο *Sensitivity* προς *Specificity* για μια συγκεκριμένη τιμή του κατωφλίου διαχωρισμού. Ένας ταξινομητής αποδίδει καλά όταν για μικρές τιμές του  $1 - Specificity$ , το *Sensitivity* προσεγγίζει την μέγιστη τιμή 1. Για την ποσοτικοποίηση αυτής της ιδιότητας



Σχήμα 3.12: Ερμηνεία της καμπύλης ROC-AUC [40]

υπολογίζουμε το εμβαδόν "κάτω από την καμπύλη" (area under the curve, AUC). Προφανώς όσο αυξάνεται η τιμή του AUC τόσο καλύτερα προβλέπει το μοντέλο.

### 3.5.3 Average Precision

Το average precision συνοψίζει την καμπύλη precision-recall ως τον σταθμισμένο μέσο όρο των precisions καθώς η τιμή του κατωφλίου  $\tau$  μεταβάλλεται [41] :

$$AP = \sum_n a_n P_n$$

όπου :

$$a_n = R_n - R_{n-1}$$

Δηλαδή, το βάρος  $a_n$ , με το οποίο σταθμίζεται το  $R_n$ , ισούται με την αύξηση του Recall από την προηγούμενη τιμή κατωφλίου. Τα  $P_n, R_n$  συμβολίζουν τις μετρικές Precision και Recall αντίστοιχα για τη  $n$ -οστή τιμή κατωφλίου. Η παραπάνω υλοποίηση διαφέρει από τον υπολογισμό της precision-recall auc με χρήση του τραπεζοειδούς κανόνα προσεγγιστικής ολοκλήρωσης, ο οποίος εφαρμόζει γραμμική παρεμβολή και συχνά οδηγεί σε αισιόδοξες εκτιμήσεις.





## Κεφάλαιο **4**

# CNN Αρχιτεκτονικές

---

Αφού αναλύσαμε τα δομικά στοιχεία-επίπεδα ενός συνελκτικού νευρωνικού δικτύου, είμαστε πλέον σε θέση να εξετάσουμε μερικές από τις σημαντικότερες αρχιτεκτονικές CNN.

### 4.1 LeNet

Μία από τις πρώτες σημαντικές αρχιτεκτονικές συνελκτικών νευρωνικών δικτύων προτάθηκε από τους LeCun et al. [21] το 1998 για την αναγνώριση χειρόγραφων χαρακτήρων. Η αρχιτεκτονική που σχεδίασαν ονομάστηκε LeNet-5 με τον αριθμό 5 να εκφράζει τον αριθμό των παραμετρικών επιπέδων του δικτύου. Συγκεκριμένα, το LeNet-5, απαρτίζεται από μόλις δύο ζεύγη convolutional - pooling επιπέδων ακολουθούμενων από ένα flattening convolutional επίπεδο που καταλήγει σε δύο fully connected επίπεδα.

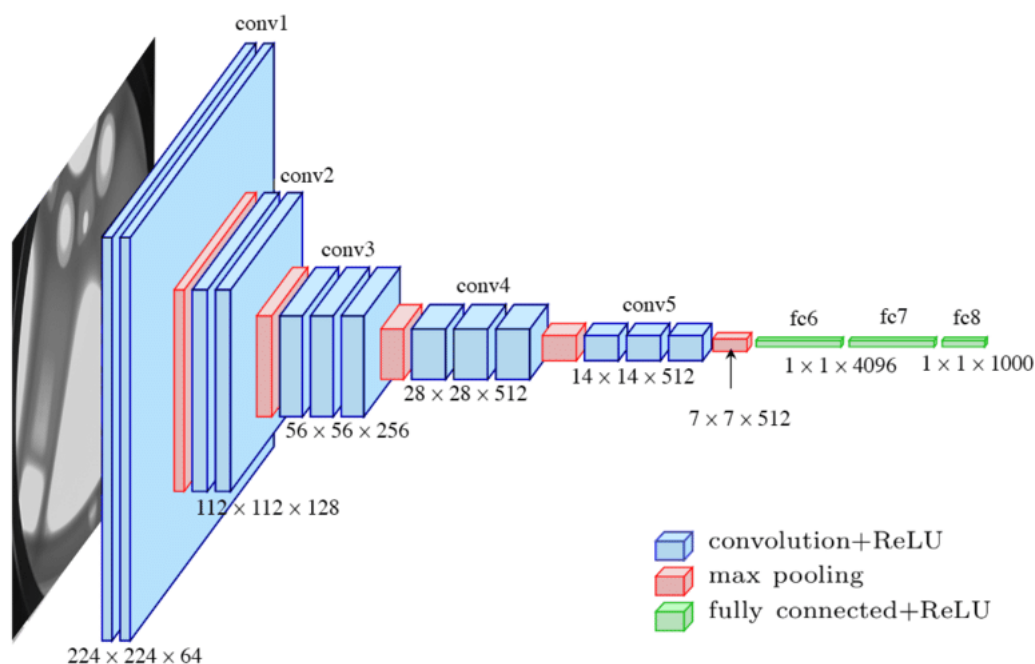
### 4.2 AlexNet

Η αρχιτεκτονική του δικτύου AlexNet [42] σχεδιάστηκε από τον Alex Krizhevsky το 2012 και καθιέρωσε τα συνελκτικά νευρωνικά δίκτυα στο πεδίο της όρασης υπολογιστών (computer vision). Το AlexNet έλαβε μέρος στον διαγωνισμό ImageNet Large Scale Visual Recognition Challenge 2012 [43] (εν συντομία ILSVRC) και κατέκτησε την πρώτη θέση με top-5 error 15.3% και μάλιστα με τεράστια διαφορά από το δεύτερο (26.1%). Η αρχιτεκτονική του Krizhevsky μοιράζεται πολλά κοινά χαρακτηριστικά με αυτήν του LeNet-5. Η καινοτομία που εισήγαγε ήταν η αύξηση του βάθους, δηλαδή του αριθμού των επιπέδων του δικτύου, καθώς και η χρήση μεθόδων κανονικοποίησης όπως Dropout και Data Augmentation. Το AlexNet αποτελείται από οκτώ παραμετρικά επίπεδα εκ των οποίων τα πρώτα πέντε είναι συνελκτικά ενώ τα τρία τελευταία είναι fully connected.

### 4.3 VGG

Οι Simonyan & Zisserman εισήγαγαν την αρχιτεκτονική του δικτύου VGG (Visual Geometric Group) το 2014 στο paper “Very Deep Convolutional Networks for Large Scale Image Recognition” [44]. Παρότι έλαβε την δεύτερη θέση στο διαγωνισμό ILSVRC 2014 - Image Classification με top-5 error 7.3%, το VGG16 κατάφερε να διακριθεί ως ένα καινοτόμο μοντέλο. Αποτέλεσε βάση για τη δημιουργία πολλών σύγχρονων αρχιτεκτονικών ενώ μέχρι και

σήμερα είναι ιδιαίτερα δημοφιλές. Το VGG16 διαθέτει δεκαέξι παραμετρικά επίπεδα καταμεμημένα σε blocks. Τα δυο πρώτα blocks αποτελούνται από δύο συνελκτικά επίπεδα με kernels διαστάσεων  $3 \times 3$  ακολουθούμενα από ένα επίπεδο συγκέντρωσης max pooling  $2 \times 2$ . Τα τρία τελευταία blocks αποτελούνται από τρία συνελκτικά επίπεδα με kernels διαστάσεων  $3 \times 3$  ακολουθούμενα από ένα επίπεδο συγκέντρωσης max pooling  $2 \times 2$ . Το δίκτυο καταλήγει σε δύο fully connected επίπεδα. Το VGG16 κατέστησε σαφές ότι το βάθος ενός συνελκτικού νευρωνικού δικτύου παίζει ιδιαίτερα σημαντικό ρόλο στην επίδοσή του.



Σχήμα 4.1: Η αρχιτεκτονική του VGG16 [45]

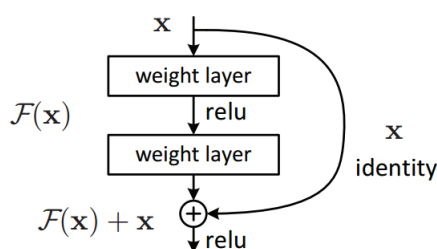
## 4.4 ResNet

Το Residual Neural Network ή ResNet αναπτύχθηκε από τους Kaiming He et al. [46] και κατέλαβε την πρώτη θέση στον διαγωνισμό ILSVRC 2015 μειώνοντας το top-5 error στο 3,6%. Το πρόβλημα που αντιμετώπιζαν τα συμβατικά CNN ήταν ότι η συνεχής αύξηση του βάθους δεν συνοδευόταν και από αύξηση των επιδόσεων. Οι σχεδιαστές των ResNets μέσω της εισαγωγής των Residual Blocks κατέστησαν εφικτή την σημαντική αύξηση του βάθους των δικτύων πετυχαίνοντας ταυτόχρονα το καλύτερο σκορ που είχε καταγραφεί μέχρι εκείνη τη στιγμή στο διαγωνισμό.

### 4.4.1 Residual Block

Το residual block ανατρέπει τη συμβατική ροή επεξεργασίας των CNN κατά την οποία το κάθε επίπεδο λαμβάνει είσοδο  $x$  και παράγει έξοδο  $F(x)$ . Συγκεκριμένα, κάθε residual block διαθέτει μια σύνδεση που παρακάμπτει τα παραμετρικά επίπεδα που εφαρμόζουν την

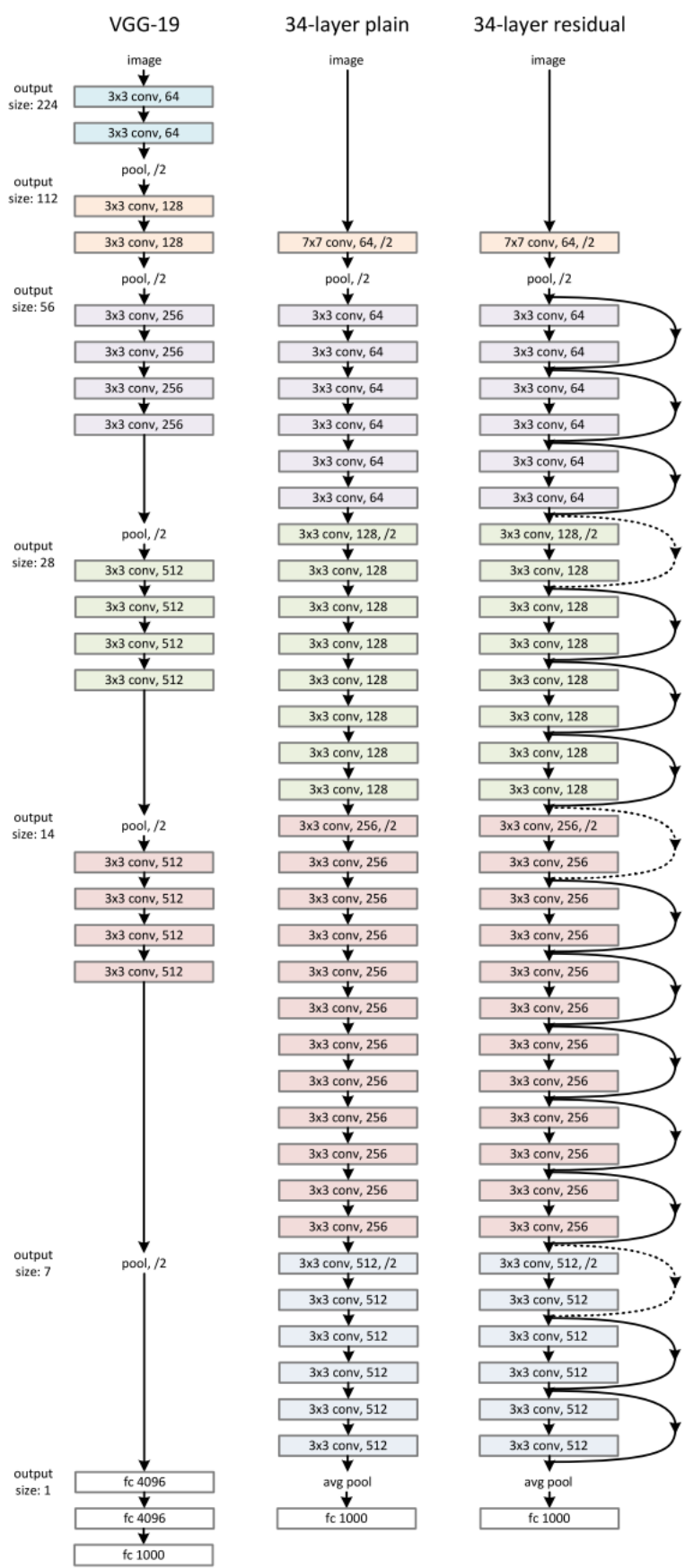
συνάρτηση  $F$  στην είσοδο  $x$ . Η σύνδεση αυτή ονομάζεται "skip identity connection". Έτσι σε κάθε residual block λαμβάνουμε έξοδο  $F(x) + x$ .



Σχήμα 4.2: Η δομή ενός Residual Block [46]

Υπάρχουν πολλές εκδοχές των ResNets με ειδοποιός διαφορά τον αριθμό των επιπέδων τους (ResNet50, ResNet101, ResNet152 κ.ά.). Στην παρούσα εργασία χρησιμοποιείται το ResNet50, δηλαδή το ResNet των πενήντα επιπέδων, καθώς οι διαθέσιμοι υπολογιστικοί πόροι καθιστούν δύσκολα τα μοντέλα μεγαλύτερου βάθους.

Η Αρχιτεκτονική του ResNet50 αρχικά διαθέτει ένα συνελκτικό και ένα επίπεδο συγκεντρώσεις ενώ στη συνέχεια μπορεί να διακριθεί σε τέσσερα στάδια. Το πρώτο στάδιο διαθέτει τρία Residual Blocks, το δεύτερο στάδιο τέσσερα, το τρίτο στάδιο έξι και το τέταρτο στάδιο τρία. Κάθε Residual Block απαρτίζεται από τρία συνελκτικά επίπεδα. Καθώς μεταβαίνουμε από το ένα στάδιο στο επόμενο, το βάθος της εισόδου, δηλαδή ο αριθμός των καναλιών, διπλασιάζεται ενώ οι χωρικές διαστάσεις της υποδιπλασιάζονται. Το δίκτυο καταλήγει σε ένα επίπεδο συγκέντρωσης average pooling ακολουθούμενο από ένα fully connected επίπεδο.



Σχήμα 4.3: Η αρχιτεκτονική του ResNet [46]

## 4.5 EfficientNet

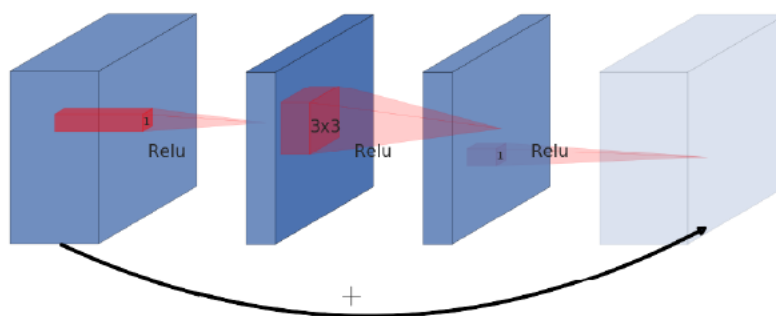
Το Μάιο του 2019, οι μηχανικοί Mingxing Tan και Quoc V. Le της Google Brain Team, δημοσίευσαν το άρθρο “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks” [47]. Βασικός στόχος της δημοσίευσης ήταν η εισαγωγή μιας αποτελεσματικής στρατηγικής κλιμάκωσης νευρωνικών δικτύων, παράλληλα όμως εισήγαγαν μία νέα οικογένεια νευρωνικών δικτύων, τα EfficientNets.

### 4.5.1 EfficientNet-B0

Η αρχιτεκτονική EfficientNet-B0 που αποτελεί το μοντέλο-βάση για την ανάπτυξη των υπολοίπων EfficientNets, σχεδιάστηκε μέσω της τεχνικής “Neural Architecture Search” (NAS) [48]. Η NAS αυτοματοποιεί το σχεδιασμό νευρωνικών δικτύων με στόχο την βελτιστοποίηση της απόδοσης τους και την ελαχιστοποίηση της πολυπλοκότητάς τους. Η αρχιτεκτονική που προέκυψε περιέχει έντεκα εκατομμύρια παραμέτρους προς εκπαίδευση και έχει ως δομικά στοιχεία τη συνάρτηση ενεργοποίησης Swiss, τα Inverted Residual Blocks και τα Squeeze & Excitation Blocks που θα εξετάσουμε στη συνέχεια.

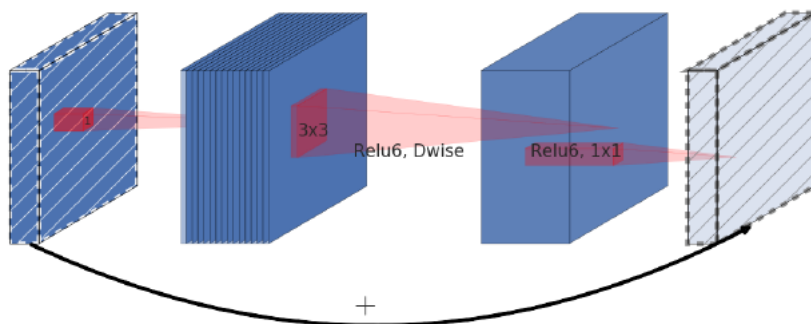
### 4.5.2 Inverted Residual Block

Το inverted residual block προτάθηκε στην αρχιτεκτονική MobileNet [49] η οποία χρησιμοποιεί depthwise separable convolution εντός του block. Στα συμβατικά residual blocks τα skip connections χρησιμοποιούνται για την σύνδεση επιπέδων με μεγάλο βάθος, δηλαδή μεγάλο αριθμό καναλιών, ενώ εντός του block τα επίπεδα διαθέτουν μικρότερο βάθος:



Σχήμα 4.4: Residual block [49]

Τα inverted residual blocks, αντιθέτως, συνδέουν επίπεδα μικρού βάθους (bottlenecks):



Σχήμα 4.5: *Inverted residual block* [49]

Μέσω αυτής της προσέγγισης μειώνονται σημαντικά οι παράμετροι προς εκπαίδευση.

### 4.5.3 Squeeze & Excitation Block (SE block)

Όταν ένα συνελκτικό επίπεδο παράγει ως έξοδο ένα χάρτη χαρακτηριστικών, τα κανάλια της εξόδου είναι ισοβαρή. Το squeeze & excitation block ανατρέπει αυτήν την προσέγγιση εφαρμόζοντας διαφορετικούς συντελεστές σε κάθε κανάλι.

Η αρχιτεκτονική του EfficientNet-B0 αρχικά διαθέτει ένα συνελκτικό επίπεδο και στη συνέχεια επτά Inverted Residual Blocks. Κάθε ένα από αυτά τα Blocks διαθέτει διαφορετικά χαρακτηριστικά. Μετά τα Blocks ακολουθεί ένα συνελκτικό επίπεδο, ένα επίπεδο συγκέντρωσης και τέλος ένα πλήρως συνδεδεμένο επίπεδο.

## Κεφάλαιο **5**

# Transfer Learning

---

Το transfer learning αποτελεί μία ιδιαίτερα δημοφιλή μέθοδο στη μηχανική μάθηση που αποσκοπεί στην αξιοποίηση της γνώσης που έχει προέλθει από την επίλυση ενός προβλήματος A για την επίλυση ενός διαφορετικού προβλήματος B. Συγκεκριμένα το μοντέλο που έχει εκπαιδευτεί για την επίλυση του A θα αποτελέσει την βάση για την επίλυση του B. Αυτή η προσέγγιση συνήθως είναι πιο γρήγορη και αποδοτική από την εκπαίδευση ενός μοντέλου από την αρχή και έχει καθιερωθεί σε εφαρμογές όρασης υπολογιστών όπως είναι η ταξινόμηση εικόνων. Η αιτία καθιέρωσης του transfer learning είναι το γεγονός ότι τα state-of-the-art μοντέλα αποτελούνται από πάρα πολλούς παραμέτρους με αποτέλεσμα να απαιτούνται εξίσου πολλά δείγματα-εικόνες για την εκπαίδευση τους. Στην πράξη, η εύρεση ενός ικανοποιητικού αριθμού δεδομένων δεν είναι εύκολη υπόθεση και έτσι οδηγούμαστε στην χρήση μοντέλων προεκπαιδευμένων σε ένα πολύ μεγάλο σύνολο δεδομένων (π.χ. ImageNet) ως βάση για την επίλυση του προβλήματος μας.

Η υλοποίηση της μεθόδου Transfer Learning ποικίλει ανάλογα με τον αριθμό των δειγμάτων του dataset αλλά και την συνάφεια του προβλήματος μας B, με το πρόβλημα A στο οποίο έχει εκπαιδευτεί το μοντέλο-βάση:

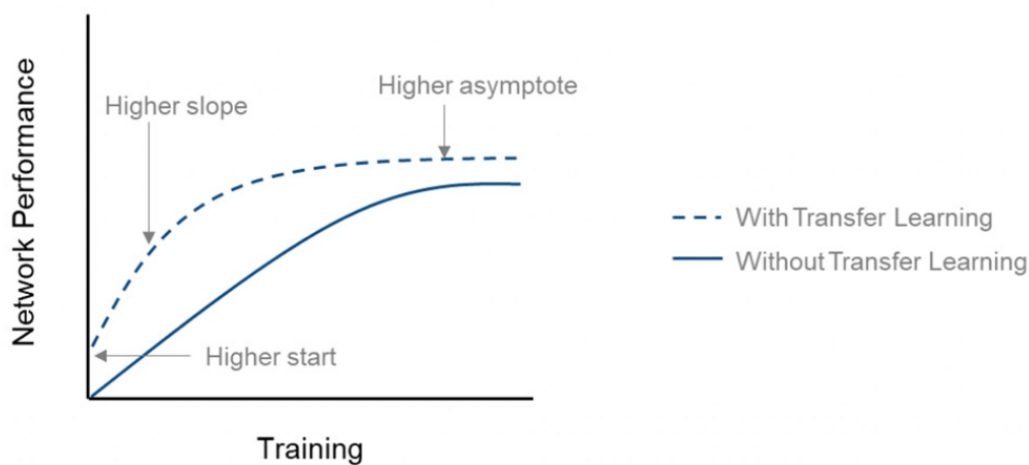
- Στην περίπτωση που το πλήθος των δειγμάτων είναι μικρό και ταυτόχρονα η συνάφεια των προβλημάτων είναι μεγάλη “παγώνουμε” τα επίπεδα του δικτύου που λειτουργούν ως εξαγωγείς χαρακτηριστικών και αντικαθιστούμε τα τελευταία fully connected layers προκειμένου να ανταποκρίνονται στο πρόβλημά μας. Έτσι, στο στάδιο της εκπαίδευσης τροποποιούνται μόνο τα βάρη των τελευταίων επιπέδων που όπως εξετάσαμε σε προηγούμενο κεφάλαιο είναι υπεύθυνα για την ταξινόμηση των δειγμάτων με βάση τα χαρακτηριστικά που έχουν εξάγει τα προηγούμενα επίπεδα. Αυτή η προσέγγιση επιλέγεται προκειμένου να ελαχιστοποιηθεί ο αριθμός των παραμέτρων και εν συνεχεία να αποφευχθεί το overfitting και η πιθανή αλλοίωση της κεκτημένης γνώσης.
- Στην περίπτωση που το πλήθος των δειγμάτων και η συνάφεια των προβλημάτων είναι μικρή πρέπει να αυξήσουμε το βάθος επανεκπαίδευσης. Αυτό σημαίνει ότι πρέπει να παγώσουμε όσο το δυνατόν λιγότερα επίπεδα ώστε να εκπαιδευτούν και συνελκτικά επίπεδα για την καλύτερη εξαγωγή χαρακτηριστικών.
- Όταν ο όγκος των διαθέσιμων δεδομένων είναι μεγάλος και η συνάφεια των προβλημάτων μικρή θεωρητικά θα ήταν καλύτερο να εκπαιδεύσουμε το δίκτυό μας από την

αρχή. Στην πράξη όμως παρατηρείται ότι η αρχικοποίηση του μοντέλου με τις παραμέτρους του προεκπαιδευμένου μοντέλου οδηγεί σε καλύτερα αποτελέσματα. Φυσικά, σε αυτή την περίπτωση η γνώση που αντλήθηκε από το προεκπαιδευμένο μοντέλο χρησιμοποιείται μόνο για την αρχικοποίηση του μοντέλου μας αφού στη συνέχεια θα εκπαιδευτεί στα νέα δεδομένα.

- Τέλος, στην ιδανική περίπτωση που το πλήθος των δειγμάτων και η συνάφεια των προβλημάτων είναι μεγάλη αρχικοποιούμε το μοντέλο μας με τις παραμέτρους του προεκπαιδευμένου και στη συνέχεια το εκπαιδύουμε στα νέα δεδομένα προκειμένου να "εξειδικευτεί" στο πρόβλημά μας.

Η Lisa Torrey αναφέρει ως τρία πιθανά οφέλη της μεθόδου transfer learning τα εξής [50]:

- Το μοντέλο αποδίδει καλύτερα κατά την έναρξη της εκπαίδευσης με αποτέλεσμα να συγκλίνει ταχύτερα (higher start).
- Επιταχύνεται η σύγκλιση του μοντέλου (higher slope).
- Και φυσικά μετά την ολοκλήρωση της εκπαίδευσης το μοντέλο αποδίδει καλύτερα (higher asymptote).



Σχήμα 5.1: Τα οφέλη της μεθόδου Transfer Learning [51]



**Μέρος **

**Πρακτικό Μέρος**

---



## Κεφάλαιο **6**

# Εργαλεία και Βιβλιοθήκες

---

### 6.1 Python Libraries

Αρχικά θα αναφερθούμε σε ορισμένες βιβλιοθήκες της γλώσσας προγραμματισμού Python που χρησιμοποιήθηκαν στο πειραματικό μέρος της διπλωματικής εργασίας

#### 6.1.1 NumPy

Η NumPy [52] αποτελεί μια βιβλιοθήκη για την Python που μεταξύ άλλων παρέχει την δομή δεδομένων ndarray. Η δομή ndarray με τη σειρά της παρέχει υποστήριξη σε μεγάλους πολυδιάστατους πίνακες, ενώ επιτρέπει και την αναπαράσταση εικόνων ως πίνακες. Αυτή η δομή δεδομένων χρησιμοποιείται κατά κόρον σε προβλήματα όρασης υπολογιστών καθώς οι περισσότερες βιβλιοθήκες επεξεργασίας εικόνας και μηχανικής μάθησης την επιλέγουν για την αναπαράσταση εικόνων. Τέλος, η NumPy διαθέτει μία τεράστια συλλογή συναρτήσεων που επίσης παίζει σημαντικό ρόλο στην υλοποίηση μας.

#### 6.1.2 Matplotlib

Η Matplotlib [53] είναι μια ιδιαίτερα ευέλικτη και εύχρηστη βιβλιοθήκη απεικόνισης (plotting library) για τη γλώσσα προγραμματισμού Python, η οποία διαθέτει συναρτήσεις απεικόνισης γραφημάτων αλλά και εικόνων.

#### 6.1.3 Scikit-learn

Η scikit-learn [54] πρόκειται για μια βιβλιοθήκη μηχανικής μάθησης για την Python. Μεταξύ άλλων, διαθέτει αλγορίθμους ταξινόμησης (classification), παλινδρόμησης (regression) και ομαδοποίησης (clustering) συμπεριλαμβάνοντας τους support vector machines - SVM, random forests, gradient boosting, k-means κ.ά.. Στα πλαίσια της παρούσας εργασίας η scikit-learn χρησιμοποιήθηκε κυρίως κατά το στάδιο της αξιολόγησης των συστημάτων.

### 6.2 Keras

Για την υλοποίηση των νευρωνικών δικτύων της διπλωματικής εργασίας χρησιμοποιήθηκε η διεπαφή προγραμματιστικών εφαρμογών (API) Keras [55]. Το Keras είναι ανοιχτού κώδικα, γραμμένο στη γλώσσα προγραμματισμού Python. Δεν αποτελεί αυτόνομη βιβλιοθήκη αλλά

high-level διεπαφή ή οποία τρέχει πάνω από low-level βιβλιοθήκες (TensorFlow, Theano, Microsoft Cognitive Toolkit). Το Keras σχεδιάστηκε ώστε να επιταχύνει και να διευκολύνει την ανάπτυξη βαθιών νευρωνικών δικτύων (deep neural networks). Παράλληλα, εστιάζει στο να είναι φιλική προς το χρήστη (user friendly), αρθρωτή (modular) και επεκτάσιμη (extensible).

### 6.3 Google Colaboratory

Τα πειράματα υλοποιήθηκαν εξ ολοκλήρου στο Google Colaboratory, εν συντομία Colab [56]. Το Colab πρόκειται για ένα cloud-based Jupyter notebook περιβάλλον που επιτρέπει την ανάπτυξη και την εκτέλεση κώδικα Python εντός του browser. Επιπλέον, μέσω του Google Drive, δίνεται η δυνατότητα εισαγωγής εξωτερικών δεδομένων για την υλοποίηση εφαρμογών μηχανικής μάθησης (machine learning), ανάλυσης δεδομένων (data analysis) κ.ά.. Το γεγονός ότι ο κώδικας των notebooks εκτελείται στους servers της Google δίνει την δυνατότητα στον χρήστη να εργαστεί με GPUs.

## Κεφάλαιο **7**

# Υλοποίηση μοντέλων

---

Η υλοποίηση των μοντέλων μηχανικής μάθησης της παρούσας διπλωματικής εργασίας μπορεί να διακριθεί στα ακόλουθα πέντε βήματα :

- Συλλογή Δεδομένων (Data Collection)
- Προεπεξεργασία Δεδομένων (Preprocessing Data)
- Επιλογή Αρχιτεκτονικής Δικτύου (Building Network)
- Εκπαίδευση Δικτύου (Training Network)
- Αξιολόγηση Συστήματος (Evaluation)

### 7.1 Συλλογή Δεδομένων

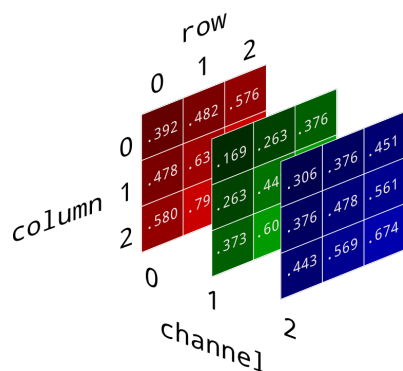
Τα δεδομένα λήφθηκαν από την ISIC στα πλαίσια του διαγωνισμού "ISIC 2016 Challenge: Lesion Classification" [57]. Συγκεκριμένα, για τον διαγωνισμό παρέχονται 900 δερματοσκοπικές εικόνες σπύλων για την εκπαίδευση του δικτύου (training data) και 379 εικόνες για την αξιολόγηση του (test data). Αυτοί οι αριθμοί, ωστόσο, είναι πολύ μικροί για την εκπαίδευση deep learning μοντέλων. Δεδομένου λοιπόν ότι η χρήση εξωτερικών δεδομένων επιτρέπεται από τους κανόνες του διαγωνισμού, στραφήκαμε στο αρχείο της ISIC [58] η οποία μεταξύ άλλων διαθέτει μία συλλογή 21.586 δερματοσκοπικών εικόνων εκ των οποίων οι 19.372 απεικονίζουν καλοήθεις (Benign) μελαγχρωματικές βλάβες ενώ οι υπόλοιπες 2.214 απεικονίζουν κακοήθεις (Malignant) μελαγχρωματικές βλάβες. Αφού η συλλογή λήφθηκε στο σύνολο της, αφαιρέθηκαν οι εικόνες που υπήρχαν και στο test set του διαγωνισμού προκειμένου να αποφευχθεί το πρόβλημα διαρροής πληροφορίας μεταξύ training και test set (Data Leakage). Μετά από αρκετές δοκιμές διαπιστώσαμε ότι οι διαθέσιμοι πόροι του Colab δεν μπορούσαν να διαχειριστούν ένα τόσο μεγάλο αριθμό εικόνων. Συνεπώς, περιοριστήκαμε στις 10.840 εικόνες με 8.626 Benign δείγματα και 2.214 Malignant. Παρατηρούμε ότι οι δύο κλάσεις είναι εξαιρετικά μη ισορροπημένες. Αυτό είναι ένα πρόβλημα που μας απασχόλησε ιδιαίτερα στα πλαίσια της εργασίας και για την αντιμετώπισή του χρησιμοποιήθηκαν τεχνικές που θα αναλυθούν στη συνέχεια. Σε πρώτη φάση παρατηρούμε ότι από την αρχική συλλογή αφαιρέθηκαν αποκλειστικά Benign δείγματα μειώνοντας έτσι τη δυσαναλογία μεταξύ των κλάσεων.

Στη συνέχεια, αφού ανακατέψαμε τις εικόνες, τις χωρίσαμε σε training και validation sets με 8.672 (80%) και 2.168 (20%) δείγματα αντίστοιχα σεβόμενοι την αναλογία των κλάσεων. Το test set του διαγωνισμού, παρότι αποτελείται από έναν πολύ μικρό αριθμό εικόνων, διατηρήθηκε अपαράλλαχτο προκειμένου η απόδοση των μοντέλων μας να συγκριθεί με τα αποτελέσματα των διαγωνιζόμενων.

## 7.2 Προεπεξεργασία Δεδομένων

Η ανάλυση των εικόνων ποικίλει ξεκινώντας από 722x542 pixels. Για λόγους απλότητας αλλά και συμβατότητας με κάποια από τα μοντέλα που χρησιμοποιήσαμε όλες οι εικόνες αναδιαμορφώθηκαν στα 224x224 pixels. Η εν λόγω μετατροπή υλοποιήθηκε με τον αλγόριθμο Bicubic interpolation, ωστόσο σύμφωνα με τους Fabrizio Nunnari et al. [59], η επιλογή του αλγόριθμου στη συγκεκριμένη περίπτωση δεν επηρεάζει την απόδοση του συστήματος. Φυσικά η μείωση της ανάλυσης συνιστά τόσο την απώλεια πληροφορίας όσο και την μερική παραμόρφωση κάποιων εικόνων, γεγονός που όπως αποδεικνύεται και πειραματικά [59] μειώνει την απόδοση του συστήματος. Ωστόσο, είναι ένας περιορισμός που πρέπει να ληφθεί δεδομένου των διαθέσιμων πόρων. Στη συνέχεια εφαρμόσαμε μία σειρά κανονικοποιήσεων στις εικόνες τις οποίες και θα παρουσιάσουμε συνοπτικά παρακάτω.

Προτού προχωρήσουμε θα εξετάσουμε πώς αναπαρίσταται μία εικόνα ως πίνακας ndarray με την χρήση του RGB μοντέλου. Μία έγχρωμη εικόνα με ανάλυση HxW (Height, Width) μπορεί να αναπαρασταθεί ως ένας τρισδιάστατος πίνακας διαστάσεων HxWx3 τον οποίο μπορούμε να φανταστούμε ως τρεις δισδιάστατους HxW πίνακες, έναν για το κόκκινο κανάλι (R), έναν για το πράσινο (G), και έναν για το μπλε (B). Κάθε στοιχείο αυτών των δισδιάστατων πινάκων λαμβάνει μία τιμή στο εύρος [0, 255] που εκφράζει την ένταση του συγκεκριμένου χρώματος για το συγκεκριμένο εικονοστοιχείο.



Εικόνα 7.1: Αναπαράσταση έγχρωμης εικόνας [60]

### 7.2.1 Min-Max Normalization

Αρχικά εφαρμόσαμε την απλούστερη μορφή κανονικοποίησης γνωστή ως Min-Max Normalization η οποία ουσιαστικά προσαρμόζει τις τιμές των pixel από το εύρος [0, 255] στο [0, 1].

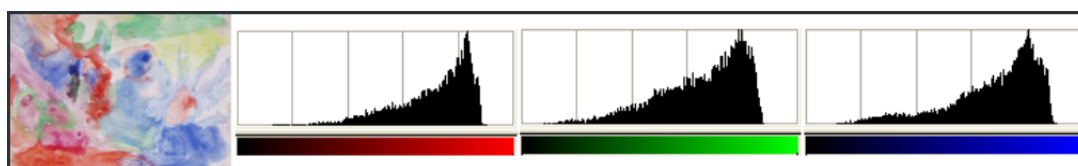
### 7.2.2 Standardization

Στη συνέχεια δοκιμάσαμε την εφαρμογή Standardization [61] σε κάθε εικόνα ξεχωριστά (sample-wise) η οποία για κάθε κανάλι της εικόνας θεωρεί ότι η κατανομή των τιμών είναι γκαουσιανή και τις προσαρμόζει ώστε να έχουν μηδενική μέση τιμή και τυπική απόκλιση ίση με τη μονάδα.

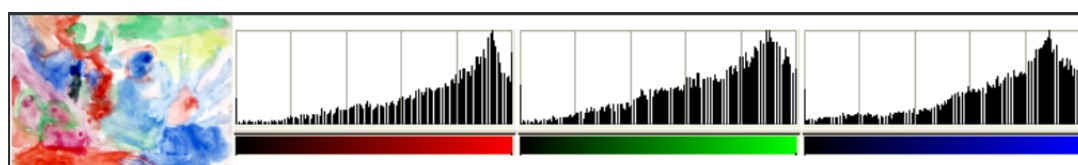
### 7.2.3 Histogram Stretching

Μετά από την ανάλυση της βιβλιογραφίας και των υλοποιήσεων των νικητών του διαγωνισμού ξεχώρισε μία εναλλακτική προσέγγιση για την προεπεξεργασία των εικόνων, η εξισορρόπηση του λευκού (White Balance). Για την εφαρμογή White balancing χρησιμοποιήθηκαν δύο διαφορετικές τεχνικές:

Η πρώτη προσπάθεια υλοποιήθηκε με τη χρήση του αλγόριθμου Histogram Stretching [62] ο οποίος προσαρμόζει τα χρώματα της εικόνας διαμορφώνοντας κάθε κανάλι (R-G-B) ξεχωριστά. Αρχικά στο ιστογράμμο κάθε καναλιού "κόβει" τις αποχρώσεις που χρησιμοποιούνται μόνο από το 0.05% των pixels και στη συνέχεια "τεντώνει" (stretching) το εναπομείναν εύρος τιμών όπως απεικονίζεται στα σχήματα 7.1, 7.2.



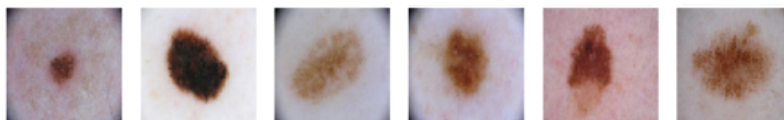
Σχήμα 7.1: Η πρωτότυπη εικόνα συνοδευόμενη από τα ιστογράμματα κάθε καναλιού [62]



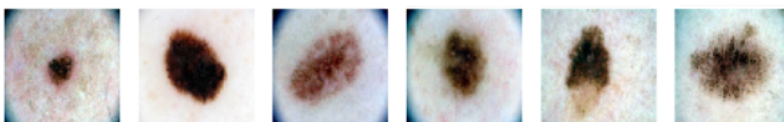
Σχήμα 7.2: Η εικόνα μετά την εφαρμογή Histogram Stretching και τα προσαρμοσμένα ιστογράμματα [62]

### 7.2.4 Color Constancy - Gray World

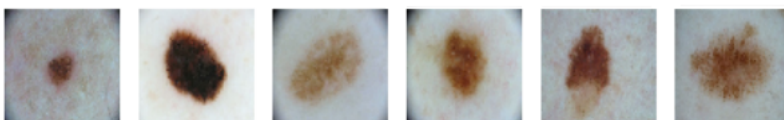
Η δεύτερη τεχνική υλοποιήθηκε με τον Color Constancy αλγόριθμο Gray World [63]. Στόχος των Color Constancy αλγορίθμων είναι η μετατροπή των χρωμάτων μιας εικόνας, που λήφθηκε υπό άγνωστη πηγή φωτός, ώστε να προσομοιώνει την λήψη της υπο μία συγκεκριμένη πηγή φωτός, συνήθως το τέλειο λευκό φως. Αυτή η μετατροπή επιτυγχάνεται σε δύο βήματα. Αρχικά πρέπει να γίνει μία εκτίμηση της απόχρωσης της πραγματικής πηγής φωτός και στη συνέχεια με βάση αυτήν την εκτίμηση η εικόνα αναπροσαρμόζεται για την επίτευξη του επιθυμητού αποτελέσματος. Για την εκτίμηση της απόχρωσης της πηγής φωτός χρησιμοποιήθηκε ο αλγόριθμος Gray World.



Εικόνα 7.2: Εικόνες στην αρχική τους μορφή



Εικόνα 7.3: Εικόνες μετά την εφαρμογή Histogram Stretching



Εικόνα 7.4: Εικόνες μετά την εφαρμογή White Balance μέσω του αλγόριθμου Color Constancy - Gray World

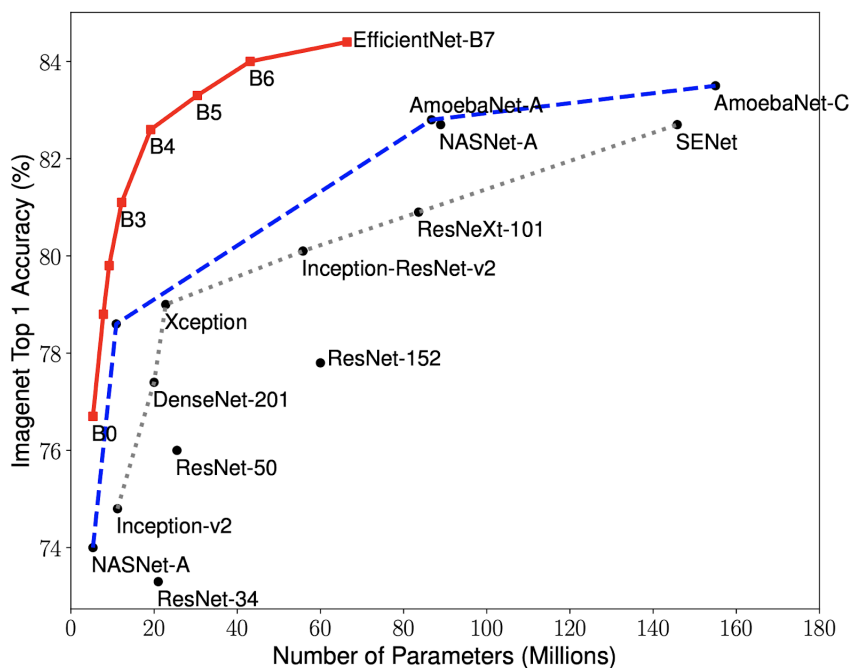
Από όλες τις παραπάνω τεχνικές προκρίθηκε το White balancing και συγκεκριμένα ο αλγόριθμος Gray Word καθώς πέτυχε την καλύτερη απόδοση.

### 7.3 Επιλογή Αρχιτεκτονικής Δικτύου

Στο πειραματικό μέρος της διπλωματικής εργασίας έγιναν δοκιμές σε πολλά μοντέλα, αυτά όμως που εν τέλει επιλέξαμε να παρουσιάσουμε είναι τα εξής:

- **ConvNet**: Αποτέλεσε την αφετηρία της πειραματικής διαδικασίας. Πρόκειται για ένα μοντέλο με απλή αρχιτεκτονική που υλοποιήθηκε για την εξοικείωση μας με τα επίπεδα και την τεχνολογίες των συνελκτικών νευρωνικών δικτύων.
- **VGG16**: Αποτέλεσε το αμέσως επόμενο βήμα. Αύξησε σημαντικά την πολυπλοκότητα αλλά και την απόδοση του συστήματος.
- **ResNet50**: Δεν θα μπορούσε να παραληφθεί καθώς αποτέλεσε την κυρίαρχη επιλογή των διαγωνιζόμενων του ISIC Challenge 2016.
- **EfficientNet-B0**: Πρόκειται για το πιο σύγχρονο μοντέλο το οποίο με μικρότερο αριθμό παραμέτρων πέτυχε την καλύτερη απόδοση.





Σχήμα 7.3: Η επικράτηση του EfficientNet στο Imagenet

## 7.4 Εκπαίδευση Δικτύου

Σε αυτή την ενότητα θα αναφερθούμε σε ορισμένες τεχνικές λεπτομέρειες που αφορούν την εκπαίδευση των συστημάτων μας:

### 7.4.1 Η callback κλάση ModelCheckpoint

Πρόκειται για μία κλάση που κατά την διάρκεια της εκπαίδευσης του μοντέλου λαμβάνει checkpoints, δηλαδή αποθηκεύει τα βάρη του δικτύου, ανάλογα με το κριτήριο που έχει οριστεί [64]. Στη δική μας περίπτωση τα βάρη του δικτύου αποθηκεύονται σε κάθε νέο peak του validation accuracy. Τα οφέλη αυτής της προσέγγισης είναι πολλαπλά καθώς μετά την ολοκλήρωση της εκπαίδευσης μπορούμε να δοκιμάσουμε την απόδοση του μοντέλου στο testing set σε διαφορετικά στάδια. Επιπλέον, μπορούμε να συνεχίσουμε την εκπαίδευση του μοντέλου χρησιμοποιώντας τα αποθηκευμένα βάρη ως βάση. Σε αντίθεση με το early stopping δεν χρειάζεται να τερματιστεί η εκπαίδευση του δικτύου για την αποφυγή overfitting και έτσι μπορούμε να δούμε σε ποιο σημείο θα φτάσει το variance για να αξιολογήσουμε τις μεθόδους κανονικοποίησης του συστήματος.

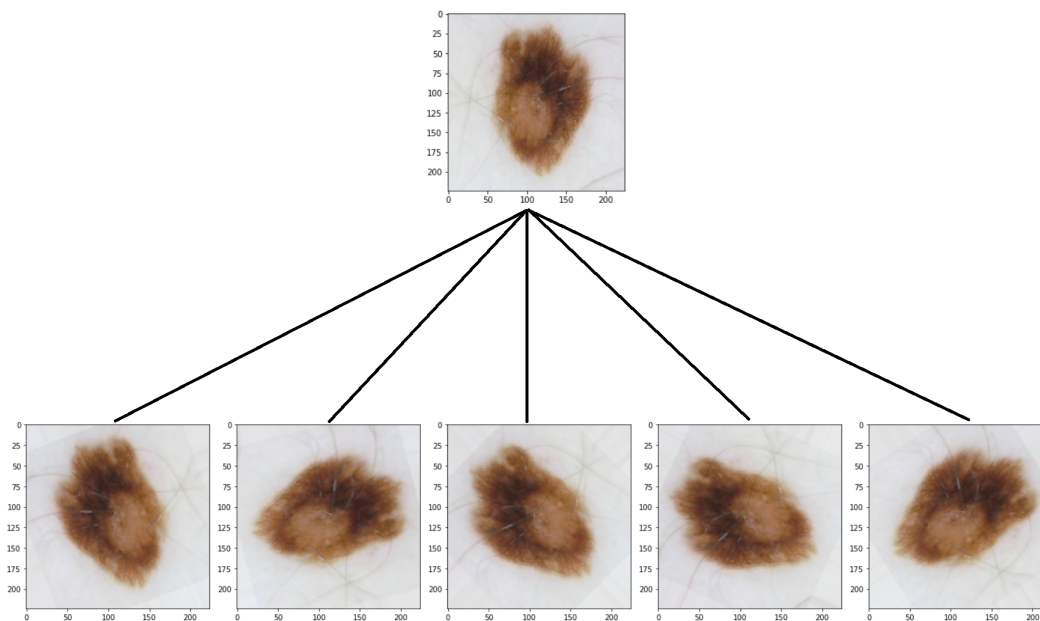
### 7.4.2 Η callback κλάση ReduceLRonPlateau

Η κλάση ReduceLRonPlateau [65] χρησιμοποιείται για την μείωση του ρυθμού εκπαίδευσης - η όταν η μετρική που έχει καθοριστεί δεν παρουσιάζει βελτίωση. Αν μετά από ένα συγκεκριμένο αριθμό εποχών (patience) η ποσότητα που παρατηρείται (monitored quantity) δεν βελτιωθεί, ο ρυθμός εκμάθησης μειώνεται κατά έναν παράγοντα (factor). Αυτή η πρακτική εφαρμόζεται σε όλα τα μοντέλα μας ακόμα και σε αυτά που έχουν adaptive learning rate

optimizers καθώς παρατηρήθηκε βελτίωση στην απόδοσή τους. Συγκεκριμένα, ο ρυθμός εκμάθησης υποδιπλασιάζεται ( $\text{factor} = 0.5$ ) μετά από 5 εποχές ( $\text{patience}=5$ ) στασιμότητας του validation accuracy ( $\text{monitor} = \text{"val\_acc"}$ ).

### 7.4.3 Η κλάση ImageDataGenerator

Η ImageDataGenerator [61] παρέχεται από το Keras και αποτελεί ένα πολύ χρήσιμο εργαλείο κατά την εκπαίδευση CNNs. Στα πλαίσια της εργασίας χρησιμοποιήθηκε για την "in-place" επαύξηση των δεδομένων εκπαίδευσης (data augmentation). Όπως αναλύσαμε στο θεωρητικό μέρος της εργασίας, η μέθοδος data augmentation αποτελεί μέτρο αντιμετώπισης του overfitting. Η ImageDataGenerator λαμβάνει δεσμίδες δεδομένων εκπαίδευσης και αφού εφαρμόσει μία σειρά μετατροπών (transformations) τροφοδοτεί τις παραγόμενες εικόνες στο δίκτυο. Οι μετατροπές που θα εφαρμοστούν αποτελούν παραμέτρους της ImageDataGenerator ( $\text{zoom\_range}$ ,  $\text{rotation\_range}$ ,  $\text{horizontal\_flip}$ ,  $\text{vertical\_flip}$  κ.ά.). Έτσι το δίκτυο σε κάθε εποχή λαμβάνει ως είσοδο καινούργιες, ελαφρώς τροποποιημένες εικόνες, και εν συνεχεία εξάγει πιο εύρωστα (robust) χαρακτηριστικά.



Εικόνα 7.5: Παράδειγμα εφαρμογής data augmentation

### 7.4.4 Class Weights

Όπως αναλύσαμε παραπάνω, τα δεδομένα εκπαίδευσης δεν είναι ισάριθμα για τις δύο κλάσεις, μάλιστα οι εικόνες κακοήθων μελαγχρωματικών βλαβών αποτελούν μόλις το 20% του συνολικού αριθμού δειγμάτων εκπαίδευσης. Για την αντιμετώπιση του παραπάνω προβλήματος το Keras παρέχει μία παράμετρο  $\text{class\_weight}$  μέσω της οποίας αυξάνεται το κόστος αποτυχίας πρόβλεψης των κακοήθων δειγμάτων. Στην πράξη αυτό επιτυγχάνεται σταθμίζοντας τη συνάρτηση κόστους με την παράμετρο  $\text{class\_weight}$ . Η αύξηση της ποινής αναγκάζει το μοντέλο να δώσει μεγαλύτερη προσοχή στην κλάση που υποεκπροσωπείται.

Η παράμετρος `class_weight` μπορεί να υπολογιστεί αυτόματα από τη την ομώνυμη συνάρτηση της βιβλιοθήκης `scikit-learn`. Για τα δεδομένα της παρούσας εργασίας `class_w = [0.62827498 2.44893809]`, όπου 0,628 ο συντελεστής για τα καλοήθη δείγματα και 2,448 ο συντελεστής για τα κακοήθη.

## 7.5 Αξιολόγηση Συστήματος

Η επιλογή της κατάλληλης μετρικής για την αξιολόγηση των συστημάτων αποτελεί πρόκληση ιδιαίτερα σε προβλήματα ταξινόμησης με `imbalanced dataset`. Συγκεκριμένα, σε προβλήματα διάγνωσης καρκίνου συνηθίζεται τα καλοήθη δείγματα να υπερέχουν αριθμητικά έναντι των κακοήθων δειγμάτων. Αν για παράδειγμα σε ένα τέτοιο πρόβλημα επιλέξουμε ως μετρική την ακρίβεια (`accuracy`) και η αναλογία καλοήθων κακοήθων δειγμάτων είναι 10 προς 1, μπορεί το μοντέλο μας να πετύχει ικανοποιητικό σκορ (90%) ταξινομώντας όλα τα δείγματα ως καλοήθη. Στην πράξη όμως, αυτός ο ταξινομητής δεν έχει καμία χρησιμότητα, αντιθέτως, θα μπορούσε να χαρακτηριστεί ακόμα και επικίνδυνος, καθώς αποτυγχάνει στον εντοπισμό της παθογένειας. Γίνεται λοιπόν αντιληπτό ότι η επιλογή μιας ακατάλληλης μετρικής για την αξιολόγηση των συστημάτων μπορεί να λειτουργήσει παραπλανητικά. Για την αντιμετώπιση του παραπάνω προβλήματος ο διαγωνισμός προτείνει μία σειρά μετρικών που δεν θεωρούν ισοκατανεμημένα τα δείγματα στις κλάσεις. Για την τελική αξιολόγηση και κατάταξη των διαγωνιζόμενων επιλέγεται η μετρική `average precision` και συγκεκριμένα η υλοποίηση της `scikit-learn` [57].

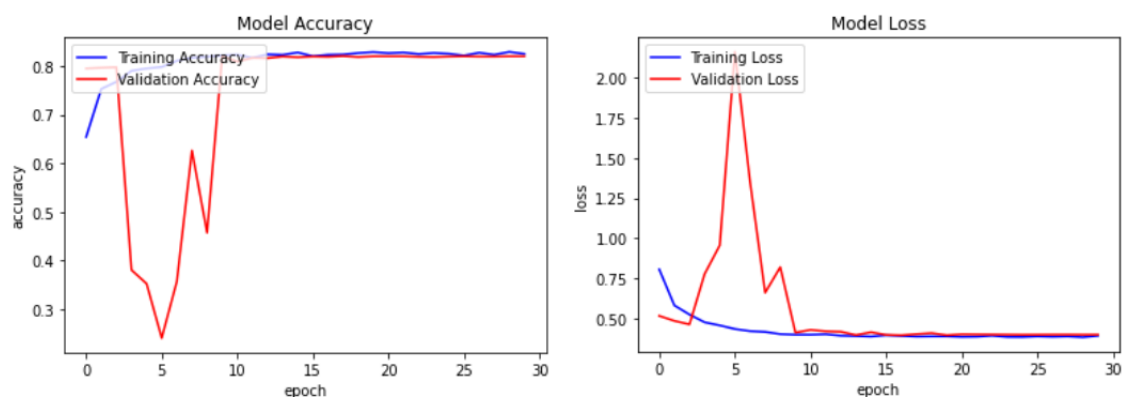
### 7.5.1 ConvNet

Το ConvNet που αναπτύξαμε απαρτίζεται από τρία ζεύγη `convolutional - pooling` επιπέδων που καταλήγουν σε δύο `fully connected` επίπεδα. Ο συνολικός αριθμός παραμέτρων του δικτύου, ανέρχεται περίπου στα 11.000.000.

Ακολουθούν οι τιμές που έλαβαν οι υπερπαραμέτροι του δικτύου:

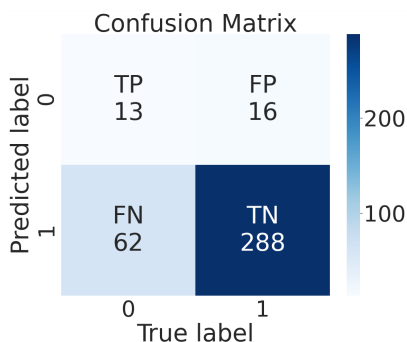
- Epochs = 30
- Batch size = 64
- Optimizer = Adam
  - Learning rate = 1e-3 (default)
- Learning rate reduction = True
  - Monitor = 'val\_accuracy'
  - Patience = 2
  - Factor = 0.5
  - Minimum learning rate = 1e-7
- Data augmentation = True

- Rotation range = 360
- Horizontal flip = True
- Vertical flip = True
- Width shift range = 0.2
- Height shift range=0.2



Σχήμα 7.4: Οι γραφικές παραστάσεις των Accuracy (αριστερά) και Loss (δεξιά) κατά την εκπαίδευση του ConvNet

Παρατηρούμε ότι τις πρώτες εποχές οι καμπύλες εκπαίδευσης παρουσιάζουν υψηλή διακύμανση ενώ στη συνέχεια σταθεροποιούνται. Το δίκτυο εμφανίζει υψηλό bias αλλά ελάχιστο variance.



Εικόνα 7.6: ConvNet Confusion Matrix

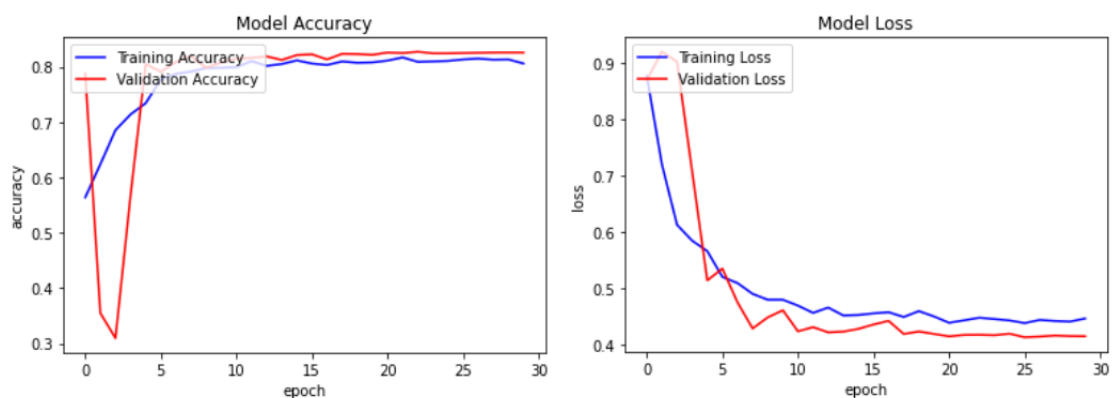
Accuracy score: 0.7941952506596306
Recall score: 0.17333333333333334
Precision score: 0.4482758620689655
<b>Average precision: 0.40764343731598407</b>
Area under ROC curve: 0.7081140350877194

Πίνακας 7.1: Οι τιμές που λήφθηκαν για μία σειρά μετρικών στο στάδιο αξιολόγησης του ConvNet

### 7.5.2 VGG

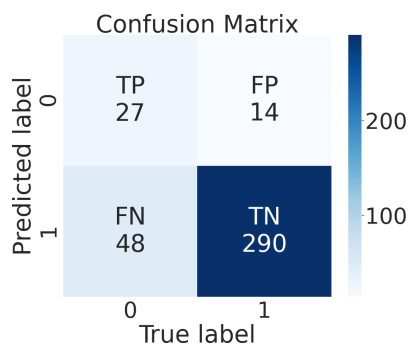
Ακολουθούν οι τιμές που έλαβαν οι υπερπαράμετροι του δικτύου για το VGG:

- Epochs = 30
- Batch size = 64
- Optimizer = SGD
  - Learning rate = 1e-4
  - Momentum = 0.9
- Learning rate reduction = True
  - Monitor = 'val\_accuracy'
  - Patience = 2
  - Factor = 0.5
  - Minimum learning rate = 1e-7
- Data augmentation = True
  - Rotation range = 360
  - Horizontal flip = True
  - Vertical flip = True
- Transfer Learning = True (με επανεκπαίδευση ολόκληρου του δικτύου)



Σχήμα 7.5: Οι γραφικές παραστάσεις των Accuracy (αριστερά) και Loss (δεξιά) κατά την εκπαίδευση του VGG

Το VGG παρουσιάζει παρόμοια συμπεριφορά με το ConvNet καθώς μετά από μερικές εποχές υψηλής διακύμανσης, οι καμπύλες εκπαίδευσης συγκλίνουν. Παρατηρούμε ότι το δίκτυο παρουσιάζει υψηλό bias αλλά μηδαμινό variance.



Εικόνα 7.7: VGG Confusion Matrix

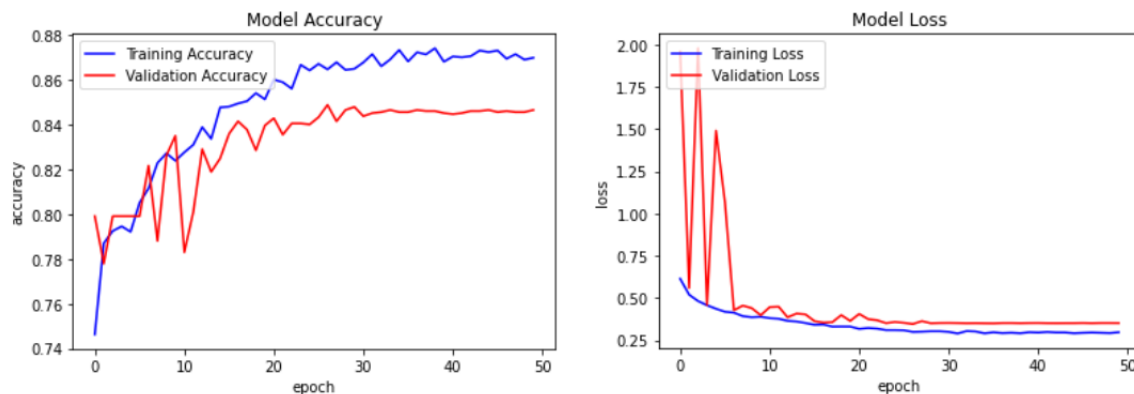
Accuracy score: 0.8364116094986808
Recall score: 0.36
Precision score: 0.6585365853658537
<b>Average precision: 0.6012474984205596</b>
Area under ROC curve: 0.8022368421052632

Πίνακας 7.2: Οι τιμές που λήφθηκαν για μία σειρά μετρικών στο στάδιο αξιολόγησης του VGG

### 7.5.3 ResNet

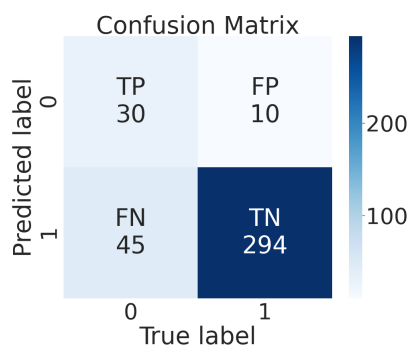
Ακολουθούν οι τιμές που έλαβαν οι υπερπαραμέτροι του δικτύου για το ResNet:

- Epochs = 50
- Batch size = 64
- Optimizer = Adam
  - Learning rate = 1e-3 (default)
- Learning rate reduction = True
  - Monitor = 'val\_accuracy'
  - Patience = 2
  - Factor = 0.5
  - Minimum learning rate = 1e-7
- Data augmentation = True
  - Rotation range = 360
  - Horizontal flip = True
  - Vertical flip = True
- Transfer Learning = True (με επανεκπαίδευση ολόκληρου του δικτύου)



Σχήμα 7.6: Οι γραφικές παραστάσεις των Accuracy (αριστερά) και Loss (δεξιά) κατά την εκπαίδευση του ResNet

Σε αντίθεση με τα δύο προηγούμενα μοντέλα εδώ η συμπεριφορά του δικτύου αντιστρέφεται καθώς βλέπουμε ότι αρχίζει να αποδίδει πολύ καλά στα δεδομένα εκπαίδευσης, δηλαδή να μειώνεται σημαντικά το bias, αλλά ταυτόχρονα παρατηρούμε μεγάλη αύξηση του variance. Τα αποτελέσματα που ακολουθούν προέκυψαν από το μοντέλο όπως διαμορφώθηκε μετά από είκοσι έξι εποχές εκπαίδευσης.



Εικόνα 7.8: ResNet Confusion Matrix

Accuracy score: 0.8548812664907651
Recall score: 0.4
Precision score: 0.75
<b>Average precision: 0.6404387047092135</b>
Area under ROC curve: 0.8169298245614035

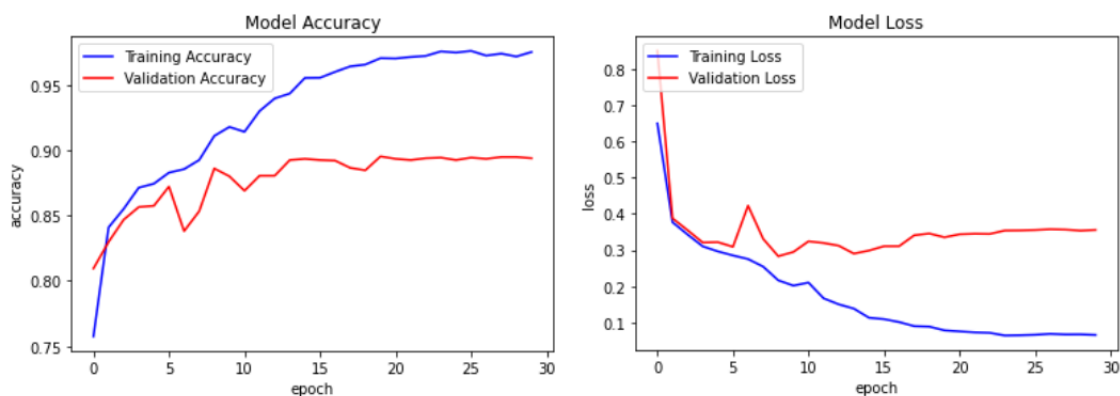
Πίνακας 7.3: Οι τιμές που λήφθηκαν για μία σειρά μετρικών στο στάδιο αξιολόγησης του ResNet

#### 7.5.4 EfficientNet

Ακολουθούν οι τιμές που έλαβαν οι υπερπαραμέτροι του δικτύου για το EfficientNet:

- Epochs = 30

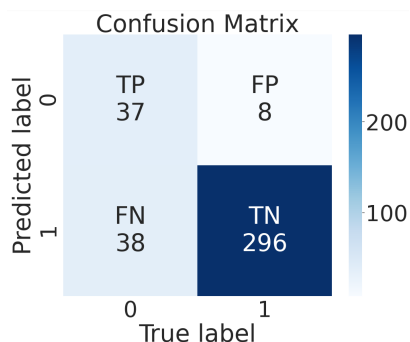
- Batch size = 64
- Optimizer = Adam
  - Learning rate = 1e-3 (default)
- Learning rate reduction = True
  - Monitor = 'val\_accuracy'
  - Patience = 2
  - Factor = 0.5
  - Minimum learning rate = 1e-7
- Data augmentation = True
  - Rotation range = 360
  - Horizontal flip = True
  - Vertical flip = True
- Transfer Learning = True (με επανεκπαίδευση ολόκληρου του δικτύου)



Σχήμα 7.7: Οι γραφικές παραστάσεις των Accuracy (αριστερά) και Loss (δεξιά) κατά την εκπαίδευση του EfficientNet

Όπως και το ResNet, το EfficientNet αποδίδει πολύ καλά στα δεδομένα εκπαίδευσης, μάλιστα μετά από κάποιο σημείο προσαρμόζεται πλήρως σε αυτά καθώς το training accuracy προσεγγίζει τη μονάδα. Τα αποτελέσματα που ακολουθούν προέκυψαν από το μοντέλο όπως διαμορφώθηκε μετά από εννέα εποχές εκπαίδευσης. Όπως παρατηρούμε στις γραφικές παραστάσεις του σχήματος 7.7, ο λόγος που το μοντέλο λήφθηκε στην ένατη εποχή είναι η σημαντική αύξηση του variance, δηλαδή η παρουσία overfitting. Τα παραπάνω επιβεβαιώθηκαν και στην πράξη καθώς μετά από δοκιμές αποδείχθηκε ότι από την ένατη εποχή και έπειτα η μετρική average precision φθίνει.



Εικόνα 7.9: *EfficientNet Confusion Matrix*

Accuracy score: 0.8786279683377308
Recall score: 0.49333333333333335
Precision score: 0.8222222222222222
<b>Average precision: 0.7132463884303015</b>
Area under ROC curve: 0.8725877192982456

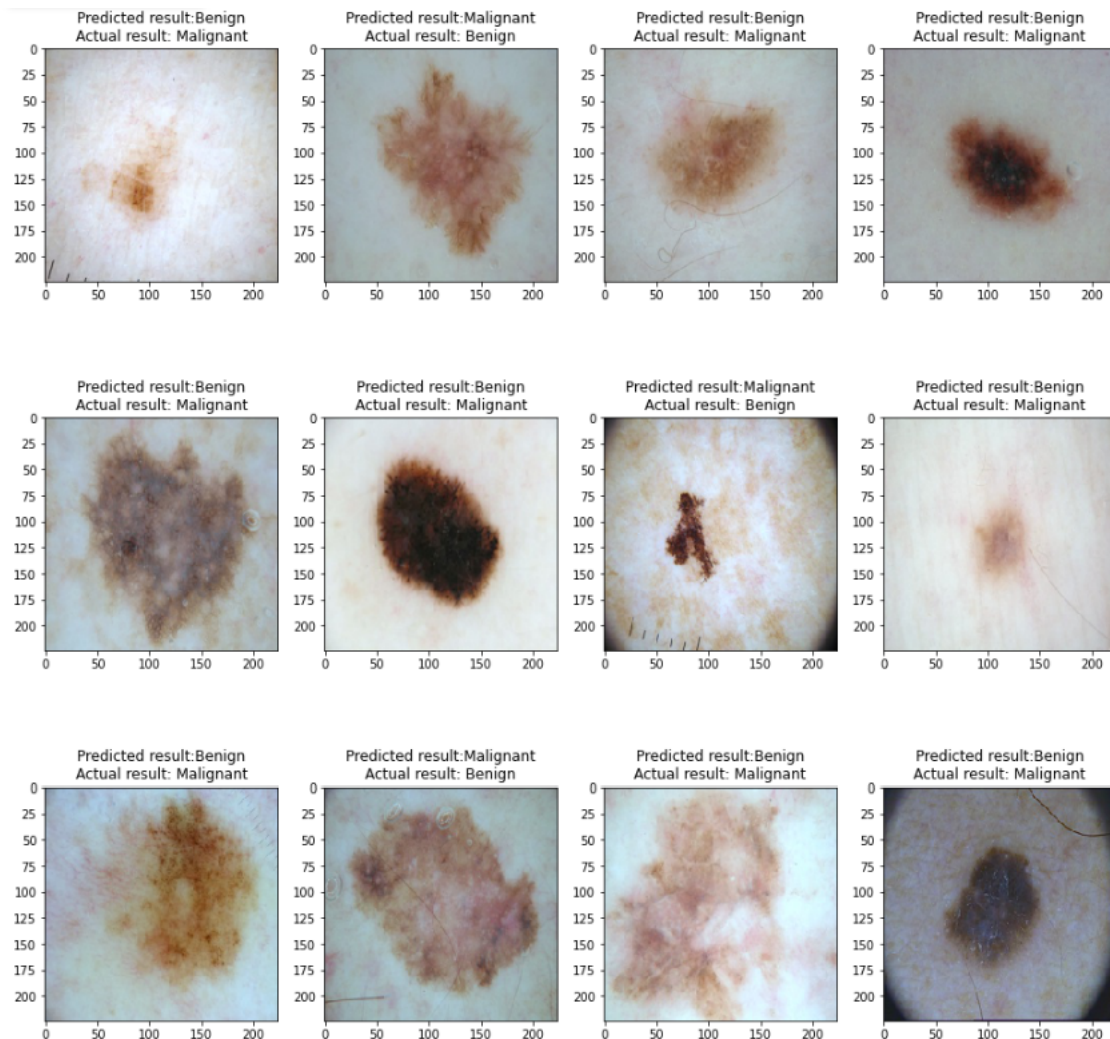
Πίνακας 7.4: Οι τιμές που λήφθηκαν για μία σειρά μετρικών στο στάδιο αξιολόγησης του *EfficientNet*

Όπως ήταν αναμενόμενο τα δίκτυα παρουσιάζουν μία σταδιακή βελτίωση με αποκορύφωμα το τρέχων state-of-the-art *EfficientNet*. Η σημαντική αύξηση του βάθους σε συνδυασμό με την χρήση residual blocks (ανεστραμμένα ή μη), φάνηκε να κάνει τη διαφορά για τα ResNet - *EfficientNet* με τα οποία και καταφέραμε να ξεπεράσουμε τις επιδόσεις των νικητών:

<i>Rank</i>	<i>Score (AP)</i>
<b>EfficientNet</b>	<b>0.713</b>
<b>ResNet</b>	<b>0.640</b>
1	0.637
2	0.619
<b>VGG</b>	<b>0.601</b>
3	0.598
4	0.563
5	0.559
6	0.555
7	0.552
8	0.545
9	0.542
10	0.537
11	0.507
12	0.504
13	0.494
14	0.479
15	0.460
16	0.434
<b>ConvNet</b>	<b>0.407</b>
17	0.390
18	0.355
19	0.347
20	0.339

Πίνακας 7.5: Η απόδοση των μοντέλων μας σε σύγκριση με το Leaderboard του διαγωνισμού [66]

Ακολουθούν ορισμένες εικόνες που το EfficientNet απέτυχε να ταξινομήσει στην σωστή κλάση:



Εικόνα 7.10: Ορισμένες μελαγχρωματικές βλάβες που το EfficientNet απέτυχε να ταξινομήσει σωστά

Παρατηρούμε ότι με βάση τα κριτήρια ABCD που αναλύσαμε στο θεωρητικό μέρος, η ταξινόμηση των παραπάνω μελαγχρωματικών βλαβών σε καλοήθεις ή κακοήθεις είναι ιδιαίτερα δύσκολη. Στο σημείο αυτό, πρέπει να υπογραμμίσουμε, ότι για κάθε βλάβη διαθέτουμε μόνο ένα στιγμιότυπο. Το γεγονός αυτό αποτελεί σημαντικό περιορισμό, καθώς αν μία κακοήθης βλάβη βρίσκεται σε αρχικό στάδιο, πιθανόν να μην έχει αναπτύξει όλα τα στοιχεία που την χαρακτηρίζουν. Η διάγνωση, λοιπόν, των εν λόγω βλαβών καθίσταται εξαιρετικά δύσκολη. Σε αυτή την παραδοχή αποδίδουμε την ύπαρξη αρκετών ψευδώς αρνητικών δειγμάτων ακόμα και από το EfficientNet.



## Κεφάλαιο 8

# Επίλογος

---

### 8.1 Συμπεράσματα

Πρωταρχικός στόχος της παρούσας διπλωματικής εργασίας ήταν η εμβάθυνση στον κλάδο της μηχανικής μάθησης. Συγκεκριμένα, η εξοικείωση μας με τα δομικά στοιχεία ενός νευρωνικού δικτύου, καθώς και με τις μεθόδους εκπαίδευσης, αξιολόγησης και βελτιστοποίησης του. Για την επίτευξη του παραπάνω στόχου αποφασίσαμε να ασχοληθούμε με το πρόβλημα που τέθηκε στον διαγωνισμό “ISIC 2016 Challenge: Lesion Classification” προκειμένου να υπάρχει και ένας απτός στόχος όσον αφορά την επίδοση των μοντέλων μας. Με την ολοκλήρωση της εργασίας λοιπόν, θεωρούμε ότι οι στόχοι μας επιτεύχθηκαν, καθώς εκτός από την εκτενή ενασχόληση μας με το αντικείμενο της μηχανικής μάθησης, καταφέραμε με ελάχιστους πόρους να ξεπεράσουμε τις επιδόσεις των νικητών. Στη συνέχεια θα αναφερθούμε επιγραμματικά σε ορισμένα συμπεράσματα που προέκυψαν κατά την διεξαγωγή της διπλωματικής εργασίας:

- Ο όγκος και η ποιότητα των δεδομένων αποτελούν ένα από τα σημαντικότερα συστατικά για την υλοποίηση ενός συστήματος μηχανικής μάθησης. Η εκτενής μελέτη των δεδομένων κρίνεται μείζονος σημασίας και οι preprocessing τεχνικές πρέπει να προσαρμόζονται στα δεδομένα του προβλήματος.
- Ενώ καμία αρχιτεκτονική μεμονωμένα δεν κατάφερε να ξεπεράσει σε απόδοση την state-of-the-art (EfficientNet), η χρήση διαφορετικών μοντέλων λειτούργησε εποικοδομητικά καθώς μέσω της τριθής με το πρόβλημα προέκυψαν συμπεράσματα που μας βοήθησαν να βελτιώσουμε το τελικό σύστημα.
- Σε ένα μεγάλο αριθμό δοκιμών τα μοντέλα μας ταξινόμησαν όλα τα δείγματα του test set ως καλοήθη, γεγονός που όπως αναλύσαμε οφείλεται στο imbalanced dataset. Είναι λοιπόν προφανές ότι αυτό το πρόβλημα δεν μπορεί να περάσει απαρατήρητο, η αντιμετώπισή του κρίνεται απαραίτητη.
- Ένα ακόμη πολύ σημαντικό πρόβλημα κατά την εκπαίδευση deep learning μοντέλων είναι η υπερπροσαρμογή των παραμέτρων στα δεδομένα εκπαίδευσης και εν συνεχεία η αδυναμία γενίκευσης σε νέα δεδομένα. Η αντιμετώπιση του εν λόγω προβλήματος είναι απαραίτητη και η χρήση data augmentation αναγκαία.

- Όπως αποδείχτηκε και πειραματικά, η εφαρμογή transfer learning σε state-of-the-art αρχιτεκτονικές αποτελεί μονόδρομο σε deep learning προβλήματα, ειδικά όταν ο όγκος των δεδομένων μας είναι περιορισμένος. Ακόμα και όταν το πρόβλημα που καλούμαστε να αντιμετωπίσουμε δεν φέρει μεγάλες ομοιότητες με το πρόβλημα στο οποίο προεκπαιδεύτηκε το μοντέλο, η χρήση transfer learning επιδρά θετικά στο σύστημα.
- Η εφαρμογή manual error analysis, δηλαδή η εμφάνιση και ανάλυση δειγμάτων που το δίκτυο απέτυχε να ταξινομήσει επιτυχώς, μπορεί να αποδειχθεί κομβική, ειδικά σε περιπτώσεις που αδυνατούμε να ερμηνεύσουμε την συμπεριφορά του δικτύου. Αυτή η μέθοδος επιβεβαιώνει την αξία μελέτης των δεδομένων και στην παρούσα διπλωματική εργασία οδήγησε στον εντοπισμό ενός σημαντικού σφάλματος.

## 8.2 Μελλοντικές Επεκτάσεις

Παρά την επίτευξη των στόχων μας όσον αφορά τις επιδόσεις των μοντέλων, υπάρχουν αρκετές κατευθύνσεις για την περαιτέρω βελτίωσή τους:

- Επέκταση του συνόλου δεδομένων. Η συλλογή ενός μεγαλύτερου και πιθανώς καλύτερα ισορροπημένου συνόλου δεδομένων είναι σχεδόν βέβαιο ότι θα οδηγήσει στην αύξηση της απόδοσης των μοντέλων μας.
- Η άρση του περιορισμού πόρων, δηλαδή η χρήση ισχυρών υπολογιστικών συστημάτων, θα επέτρεπε την αύξηση της ανάλυσης των εικόνων και ταυτόχρονα την κλιμάκωση των μοντέλων (π.χ. χρήση EfficientNet B1-B7). Η κατεύθυνση αυτή εγγυάται την σημαντική βελτίωση της τελικής απόδοσης.
- Όπως είναι γνωστό, ο χώρος αναζήτησης των βέλτιστων υπερπαραμέτρων στη βαθιά μηχανική μάθηση είναι πολύ μεγάλος. Συνεπώς η εκτενέστερη μελέτη των βέλτιστων τιμών, ειδικά για το learning rate, θα μπορούσε να βελτιώσει τα αποτελέσματα.
- Η υλοποίηση ενός αλγόριθμου περικοπής των εισόδων (cropping strategy) περιμετρικά των μελαγχρωματικών βλαβών, θα μπορούσε να λειτουργήσει ευεργετικά στην καλύτερη εξαγωγή χαρακτηριστικών κατά την διάρκεια εκπαίδευσης των μοντέλων.
- Το resizing των εικόνων θα μπορούσε να υλοποιηθεί με βάση το μέσο aspect ratio του συνόλου δεδομένων προκειμένου να ελαχιστοποιηθεί η παραμόρφωση που εισάγεται από αυτήν τη διαδικασία.
- Η χρήση ensemble learning είναι επίσης μία πολλά υποσχόμενη τεχνική που θεωρούμε ότι θα μπορούσε να αυξήσει την τελική απόδοση των μοντέλων μας.

## Βιβλιογραφία

---

- [1] Alan M Turing. *Computing machinery and intelligence. Parsing the turing test*, σελίδες 23-65. Springer, 2009.
- [2] *Bombe*. <https://en.wikipedia.org/wiki/Bombe>. Ημερομηνία πρόσβασης: 18-7-2020.
- [3] John McCarthy, Marvin L Minsky, Nathaniel Rochester και Claude E Shannon. *A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955*. *AI magazine*, 27(4):12-12, 2006.
- [4] Joseph Weizenbaum. *ELIZA—a computer program for the study of natural language communication between man and machine*. *Communications of the ACM*, 9(1):36-45, 1966.
- [5] *Tower of Hanoi*. [https://en.wikipedia.org/wiki/Tower\\_of\\_Hanoi](https://en.wikipedia.org/wiki/Tower_of_Hanoi). Ημερομηνία πρόσβασης: 13-7-2020.
- [6] Murray Campbell, A Joseph Hoane Jr και Feng hsiung Hsu. *Deep blue*. *Artificial intelligence*, 134(1-2):57-83, 2002.
- [7] *Kasparov*. <http://www.kasparov.com/>. Ημερομηνία πρόσβασης: 18-7-2020.
- [8] Marvin Minsky και Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.
- [9] Michael Haenlein και Andreas Kaplan. *A brief history of artificial intelligence: On the past, present, and future of artificial intelligence*. *California management review*, 61(4):5-14, 2019.
- [10] Arthur L Samuel. *Some studies in machine learning using the game of checkers*. *IBM Journal of research and development*, 3(3):210-229, 1959.
- [11] T Mitchell. *Machine Learning*. 1997. ISBN: 0070428077.
- [12] Yann LeCun, Yoshua Bengio και Geoffrey Hinton. *Deep learning*. *nature*, 521(7553):436-444, 2015.
- [13] *Artificial Intelligence vs Machine Learning vs Deep Learning*. <https://top6sites.com/2020/05/15/ai-vs-machine-learning-vs-deep-learning/>. Ημερομηνία πρόσβασης: 10-7-2020.
- [14] Warren S McCulloch και Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. *The bulletin of mathematical biophysics*, 5(4):115-133, 1943.

- [15] Frank Rosenblatt. *The perceptron: a probabilistic model for information storage and organization in the brain*. *Psychological review*, 65(6):386, 1958.
- [16] Simon Haykin. *Neural Networks and Learning Machines*, 3/E. Pearson Education India, 2010.
- [17] Facundo Bre, Juan M Gimenez και Victor D Fachinotti. *Prediction of wind pressure coefficients on building surfaces using artificial neural networks*. *Energy and Buildings*, 158:1429-1441, 2018.
- [18] Wilhelm Stolz, U Semmelmayr, K Johow και WH Burgdorf. *Principles of dermatoscopy of pigmented skin lesions*. *Seminars in Cutaneous medicine and Surgery*, τόμος 22, σελίδες 9-20, 2003.
- [19] *Dermoscopic structures*. <http://www.dermnostics.com/>. Ημερομηνία πρόσβασης: 16-7-2020.
- [20] *Dermoscopy*. <https://www.fotofinder.de/en/technology/skin-cancer-diagnostics/medicam>. Ημερομηνία πρόσβασης: 13-7-2020.
- [21] Yann LeCun, Léon Bottou, Yoshua Bengio και Patrick Haffner. *Gradient-based learning applied to document recognition*. *Proceedings of the IEEE*, 86(11):2278-2324, 1998.
- [22] Li Deng. *The mnist database of handwritten digit images for machine learning research [best of the web]*. *IEEE Signal Processing Magazine*, 29(6):141-142, 2012.
- [23] *CNN Layers*. <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>. Ημερομηνία πρόσβασης: 19-7-2020.
- [24] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah και Mohammed Bennamoun. *A guide to convolutional neural networks for computer vision*. *Synthesis Lectures on Computer Vision*, 8(1):1-207, 2018.
- [25] Josh Patterson και Adam Gibson. *Deep learning: A practitioner's approach*. " O'Reilly Media, Inc.", 2017.
- [26] *Zero Padding*. <https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [27] *Max Pooling*. [https://computersciencewiki.org/index.php/Max-pooling/\\_/Pooling](https://computersciencewiki.org/index.php/Max-pooling/_/Pooling). Ημερομηνία πρόσβασης: 19-7-2020.
- [28] *Hinge Loss*. <https://towardsdatascience.com/a-guide-to-neural-network-loss-functions-with-applications-in-keras-3a3baa9f71c5>. Ημερομηνία πρόσβασης: 19-7-2020.



- [29] Agnieszka Mikołajczyk και Michał Grochowski. *Data augmentation for improving deep learning in image classification problem. 2018 international interdisciplinary PhD workshop (IIPhDW)*, σελίδες 117–122. IEEE, 2018.
- [30] Lutz Prechelt. *Early stopping-but when? Neural Networks: Tricks of the trade*, σελίδες 55–69. Springer, 1998.
- [31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever και Ruslan Salakhutdinov. *Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research*, 15(1):1929–1958, 2014.
- [32] H Jabbar και Rafiqul Zaman Khan. *Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). Computer Science, Communication and Instrumentation Devices*, σελίδες 163–172, 2015.
- [33] *Early stopping*. <http://fouryears.eu/2017/12/06/the-mystery-of-early-stopping/comment-page-1/>. Ημερομηνία πρόσβασης: 18-7-2020.
- [34] *Backpropagation*. [http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture4.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture4.pdf). Ημερομηνία πρόσβασης: 18-7-2020.
- [35] *Gradient Descent*. [http://rasbt.github.io/mlxtend/user\\_guide/general\\_concepts/gradient-optimization/](http://rasbt.github.io/mlxtend/user_guide/general_concepts/gradient-optimization/). Ημερομηνία πρόσβασης: 18-7-2020.
- [36] *Saddle Point*. [https://en.wikipedia.org/wiki/Saddle\\_point](https://en.wikipedia.org/wiki/Saddle_point). Ημερομηνία πρόσβασης: 18-7-2020.
- [37] Ekaba Bisong. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer, 2019.
- [38] Sebastian Bock, Josef Goppold και Martin Weiß. *An improvement of the convergence proof of the ADAM-Optimizer. arXiv preprint arXiv:1804.10587*, 2018.
- [39] *Confusion Matrix*. [Codeproject.com](http://codeproject.com). Ημερομηνία πρόσβασης: 18-7-2020.
- [40] *ROC Curve Plot*. <https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [41] *Average Precision*. [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average\\_precision\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html). Ημερομηνία πρόσβασης: 19-7-2020.
- [42] Alex Krizhevsky, Ilya Sutskever και Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems*, σελίδες 1097–1105, 2012.
- [43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein και others. *Imagenet large scale visual recognition challenge. International journal of computer vision*, 115(3):211–252, 2015.

- [44] Karen Simonyan και Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition*. *arXiv preprint arXiv:1409.1556*, 2014.
- [45] Max Ferguson, Ronay Ak, Yung Tsun Tina Lee και Kincho H Law. *Automatic localization of casting defects with convolutional neural networks*. *2017 IEEE international conference on big data (big data)*, σελίδες 1726–1735. IEEE, 2017.
- [46] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. *Deep residual learning for image recognition*. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 770–778, 2016.
- [47] Mingxing Tan και Quoc V Le. *Efficientnet: Rethinking model scaling for convolutional neural networks*. *arXiv preprint arXiv:1905.11946*, 2019.
- [48] Thomas Elsken, Jan Hendrik Metzen και Frank Hutter. *Neural architecture search: A survey*. *arXiv preprint arXiv:1808.05377*, 2018.
- [49] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov και Liang Chieh Chen. *Mobilenetv2: Inverted residuals and linear bottlenecks*. *Proceedings of the IEEE conference on computer vision and pattern recognition*, σελίδες 4510–4520, 2018.
- [50] Emilio Soria Olivas, Jos David Mart Guerrero, Marcelino Martinez-Sober, Jose Rafael Magdalena-Benedito, L Serrano και others. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques*. IGI Global, 2009.
- [51] *Transfer Learning*. <https://kr.mathworks.com/discovery/transfer-learning.html>. Ημερομηνία πρόσβασης: 19-7-2020.
- [52] *Numpy*. <https://numpy.org/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [53] *Matplotlib*. <https://matplotlib.org/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [54] *Scikit-learn*. <https://scikit-learn.org/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [55] *Keras*. <https://keras.io/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [56] *Google Colaboratory*. <https://colab.research.google.com/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [57] *ISIC Challenge 2016*. <https://challenge.isic-archive.com/landing/2016>. Ημερομηνία πρόσβασης: 19-7-2020.
- [58] *ISIC Archive*. <https://isic-archive.com/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [59] Fabrizio Nunnari και Daniel Sonntag. *A CNN toolbox for skin cancer classification*. *arXiv preprint arXiv:1908.08187*, 2019.
- [60] *RGB Image*. [https://brohrer.github.io/convert\\_rgb\\_to\\_grayscale.html](https://brohrer.github.io/convert_rgb_to_grayscale.html). Ημερομηνία πρόσβασης: 19-7-2020.

- 
- [61] *Keras Preprocessing*. <https://keras.io/api/preprocessing/image/>. Ημερομηνία πρόσβασης: 19-7-2020.
- [62] *Histogram Stretching*. <https://docs.gimp.org/2.6/en/gimp-layer-white-balance.html>. Ημερομηνία πρόσβασης: 19-7-2020.
- [63] Catarina Barata, M Emre Celebi και Jorge S Marques. *Improving dermoscopy image classification using color constancy*. *IEEE journal of biomedical and health informatics*, 19(3):1146–1152, 2014.
- [64] *Model Checkpoint*. [https://keras.io/api/callbacks/model\\_checkpoint/](https://keras.io/api/callbacks/model_checkpoint/). Ημερομηνία πρόσβασης: 19-7-2020.
- [65] *Reduce Learning Rate On Plateau*. [https://keras.io/api/callbacks/reduce\\_lr\\_on\\_plateau/](https://keras.io/api/callbacks/reduce_lr_on_plateau/). Ημερομηνία πρόσβασης: 19-7-2020.
- [66] *ISIC Challenge 2016 Leaderboard*. <https://challenge.kitware.com/#phase/5667455bcad3a56fac786791>. Ημερομηνία πρόσβασης: 14-1-2020.