



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΙΩΑΝΝΙΝΩΝ

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.**  
**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΟΠΤΙΚΗΣ ΑΝΑΓΝΩΡΙΣΗΣ**  
**ΧΑΡΑΚΤΗΡΩΝ ΓΙΑ ΑΛΦΑΡΙΘΜΗΤΙΚΑ ΔΕΔΟΜΕΝΑ ΜΕ ΧΡΗΣΗ**  
**ΤΕΧΝΙΚΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ**

Σαρικάκης Ηλίας – Ιωάννης

ΑΜ: 1428

Email: [sarikakisilias@gmail.com](mailto:sarikakisilias@gmail.com)

Επιβλέπων: Σταύρος Αδάμ

Επίκουρος Καθηγητής

Άρτα, Οκτώβριος 2021

**DEVELOPMENT OF AN OPTICAL CHARACTER RECOGNITION  
APPLICATION FOR ALPHANUMERIC DATA USING MACHINE  
LEARNING TECHNIQUES**

**Εγκρίθηκε από τριμελή εξεταστική επιτροπή**

Αρτα, 8/10/2021

## **ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ**

1. Επιβλέπων καθηγητής

Σταύρος Αδάμ

Επίκουρος καθηγητής

2. Μέλος επιτροπής

Νικόλαος Γιαννακέας

Επίκουρος καθηγητής

3. Μέλος επιτροπής

Πέτρος Καρβέλης

Επίκουρος καθηγητής

© Σαρικάκης Ηλίας – Ιωάννης, 2021.

Με επιφύλαξη παντός δικαιώματος, All rights reserved.

## **Δήλωση μη λογοκλοπής**

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφή και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Σαρικάκης Ηλίας – Ιωάννης

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή μου, Κύριο Σταύρο Αδάμ για τις χρήσιμες συμβουλές του κατά τα πρώτα στάδια υλοποίησης της εφαρμογής, καθώς επίσης και την οικογένεια μου για την ιδιαίτερη υπομονή, στήριξη και ελευθερία που μου έδωσαν για να ολοκληρώσω την συγκεκριμένη εργασία. Τέλος θα ήθελα να δώσω ένα ιδιαίτερα μεγάλο ευχαριστώ σε ένα εξαιρετικά σημαντικό μου πρόσωπο με αρχικά I.U. Ένα πρόσωπο που ήταν πάντα πλάι μου, από την πρώτη μέρα εκπόνησης, έως και την τελευταία ημέρα παράδοσης της συγκεκριμένης εργασίας.

## ΠΕΡΙΛΗΨΗ

Η συγκεκριμένη πτυχιακή εργασία αναφέρεται στην τεχνολογία της οπτικής αναγνώρισης χαρακτήρων (OCR). Αρχικά παρουσιάζεται μια σύντομη εξήγηση της σημασίας που έχουν τα δεδομένα και η πληροφορία στον κόσμο της μηχανικής μάθησης και της τεχνητής νοημοσύνης, σχετικά με τον τρόπο με τον οποίο δέχονται επεξεργασία και τον τρόπο με τον οποίο αξιοποιούνται. Στη συνέχεια γίνεται μια απόπειρα ανάλυσης των νευρωνικών δικτύων και του τρόπου με τον οποίο αξιοποιούνται τα δεδομένα που λαμβάνουν ως είσοδο για να παράγουν την απαιτούμενη πληροφορία. Έπειτα ο αναγνώστης γνωρίζεται με την τεχνολογία της OCR, την ιστορία της, καθώς επίσης και τον τρόπο με τον οποίο αξιοποιείται η συγκεκριμένη τεχνολογία. Τέλος υπάρχει μια εκτεταμένη αναφορά στην εφαρμογή που υλοποιήθηκε αποκλειστικά για τον σκοπό της συγκεκριμένης εργασίας, περιλαμβάνοντας μια αναλυτική περιγραφή των χαρακτηριστικών και του τρόπου λειτουργίας της.

**Λέξεις-κλειδιά:** Υλοποίηση εφαρμογής, Οπτική Αναγνώριση Χαρακτήρων (OCR), Νευρωνικά δίκτυα, Python, Keras

## **ABSTRACT**

The current thesis aims to introduce the reader to the world of Optical Character Recognition (OCR). At first there is a brief explanation of the importance that data and information have in the world of machine learning and artificial intelligence regarding the way they are being processed and the way they are being used. Next there is an attempt at analyzing what neural networks are and how they work in harmony with the collected data to produce the needed information. Afterwards the reader gets introduced to the technology of OCR, its history and also the many different applications that use this specific technology. Finally, there is an extensive mention of the software that was developed exclusively for the purpose of the current thesis, including a detailed description of its features and its functionality.

**Keywords:** Software development, Optical Character Recognition (OCR), Neural networks, Python, Keras



# ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	6
ΠΕΡΙΛΗΨΗ.....	7
ABSTRACT .....	8
ΠΕΡΙΕΧΟΜΕΝΑ .....	9
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ.....	11
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	12
ΕΙΣΑΓΩΓΗ.....	13
1. ΔΕΔΟΜΕΝΑ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑ.....	14
1.1 ΟΡΙΣΜΟΣ .....	14
1.2 ΕΙΔΗ ΔΕΔΟΜΕΝΩΝ .....	15
1.3 ΤΡΟΠΟΙ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΜΩΝ.....	16
1.4 ΕΞΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΑΣ .....	17
1.5 ΑΞΙΟΠΟΙΗΣΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ.....	18
2. ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ .....	19
2.1 ΤΙ ΕΙΝΑΙ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ.....	19
2.2 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ.....	19
2.2.1 Κρυμμένες Στρώσεις ή Στρώματα .....	20
2.2.2 Βάρη και Bias .....	21
2.2.3 Συνάρτηση Ενεργοποίησης .....	22
2.3 ΠΩΣ ΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΕΞΑΡΤΩΝΤΑΙ ΑΠΟ ΤΑ ΔΕΔΟΜΕΝΑ.....	26
2.4 ΕΙΔΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ .....	28
2.5 ΤΡΟΠΟΙ ΑΞΙΟΠΟΙΗΣΗΣ .....	32
3. ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ .....	33
3.1 ΙΣΤΟΡΙΑ.....	33
3.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ .....	36

3.3 ΕΦΑΡΜΟΓΕΣ .....	37
3.4 ΚΑΤΗΓΟΡΙΕΣ.....	39
3.5 ΤΕΧΝΙΚΕΣ .....	42
4. ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ .....	45
4.1 Η ΣΥΛΛΗΨΗ ΤΗΣ ΙΔΕΑΣ .....	45
4.2 DATASETS .....	46
4.2.1 Γεννήτρια παραγωγής εικόνων .....	48
4.2.2 Ενίσχυση εικόνων .....	52
4.3 ΚΑΤΑΣΚΕΥΗ ΜΟΝΤΕΛΟΥ.....	55
4.3.1 Το δίκτυο.....	56
4.3.2 Datagen, Ανακλητές, Μεταγλώττιση .....	60
4.3.3 Αξιολόγηση του μοντέλου .....	63
4.3.4 Υποπροσαρμογή και Υπερπροσαρμογή.....	66
4.4 ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ.....	67
4.5 ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ .....	71
4.6 ΑΠΟΤΕΛΕΣΜΑΤΑ.....	73
4.7 ΒΕΛΤΙΩΣΕΙΣ .....	76
ΑΝΑΦΟΡΕΣ .....	77

## ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Βαθύ νευρωνικό δίκτυο. ....	21
Εικόνα 2: Βηματική συνάρτηση.....	23
Εικόνα 3: Σιγμοειδής συνάρτηση.....	24
Εικόνα 4: Συνάρτηση ReLU.....	25
Εικόνα 5: Συναρτήσεις Leaky ReLU και Parametric ReLU.....	26
Εικόνα 6: Ένα σχεδόν ολοκληρωμένο διάγραμμα των νευρωνικών δικτύων.....	29
Εικόνα 7: Perceptron και δίκτυο εμπρόσθιας τροφοδοσίας.....	30
Εικόνα 8: Επαναλαμβανόμενο νευρωνικό δίκτυο. ....	31
Εικόνα 9: Συνελικτικό νευρωνικό δίκτυο. ....	32
Εικόνα 10: Ένα μικρό δείγμα από το MNIST dataset.....	47
Εικόνα 11: Ένα μικρό δείγμα από την γεννήτρια εικόνων. ....	48
Εικόνα 12: Ο χαρακτήρας με το εντοπισμένο περίγραμμα. ....	54
Εικόνα 13: Η ορθογωνιοποίηση του χαρακτήρα. ....	54
Εικόνα 14: Πριν και μετά την συνάρτηση Augment_Image. ....	55
Εικόνα 15: Τρόπος λειτουργίας του Conv2D. ....	58
Εικόνα 16: Γραφική αναπαράσταση της λειτουργίας MaxPooling2D. ....	59
Εικόνα 17: Γραφική αναπαράσταση αποτελεσμάτων Πίνακα 10. ....	65
Εικόνα 18: Υποπροσαρμογή και Υπερπροσαρμογή μοντέλου.....	66
Εικόνα 19: Παράθυρο γραφικής διεπαφής για δημιουργία περιοχών. ....	71
Εικόνα 20: Παράθυρο επιλογής διαδικασίας φόρτωσης ή δημιουργίας μοντέλου. ....	73
Εικόνα 21: Παράθυρο επιλογής dataset για δημιουργία και εκπαίδευση. ....	73
Εικόνα 22: Παράθυρο αποτελεσμάτων.....	75

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Ορισμός γραμματοσειρών.....	49
Πίνακας 2: Συλλογή χαρακτήρων.....	49
Πίνακας 3: Γεννήτρια εικόνων.....	50
Πίνακας 4: Συνάρτηση Augment_Image.....	53
Πίνακας 5: Το σειριακό μοντέλο βαθιάς μάθησης της εφαρμογής.....	57
Πίνακας 6: Datagen, train_set και validation_set.....	61
Πίνακας 7: Οι ανακλητές του προγράμματος.....	62
Πίνακας 8: Μεταγλώττιση του μοντέλου.....	63
Πίνακας 9: Η συνάρτηση εκπαίδευσης.....	63
Πίνακας 10: Αποτελέσματα εκπαίδευσης αριθμητικών χαρακτήρων για 10 εποχές.....	64
Πίνακας 11: Η συνάρτηση sort_contours.....	68
Πίνακας 12: Διαδικασία απομόνωσης χαρακτήρων.....	70
Πίνακας 13: Η συνάρτηση φόρτωσης μοντέλων.....	74
Πίνακας 14: Η συνάρτηση πρόβλεψης.....	75

## ΕΙΣΑΓΩΓΗ

Τεχνητή νοημοσύνη, νευρωνικά δίκτυα, μηχανική και βαθιά μάθηση: όροι ευρέως γνωστοί τόσο εντός, όσο και εκτός του χώρου της πληροφορικής. Μια σειρά όρων που από πίσω τους περιλαμβάνουν μια ολόκληρη επιστήμη, περιτριγυρισμένη από ταλαντούχους ερευνητές και παθιασμένους ανθρώπους που επιδιώκουν να βελτιώσουν και να εξελίξουν ριζικά τον τρόπο ζωής μας μέσα από λογισμικά και εφευρέσεις αυτοματοποίησης χειρωνακτικών διαδικασιών.

Η οπτική αναγνώριση χαρακτήρων (OCR) είναι ένας τομέας που επιδιώκει την υλοποίηση αλγορίθμων που επιτρέπουν την αναγνώριση τυπωμένων ή ακόμα και χειρόγραφων χαρακτήρων μέσα από έγγραφα και εικόνες. Μια επιστήμη που βασίζεται σε δεδομένα και πληροφορίες που μπορούμε να εξάγουμε από αυτά. Αυτός είναι και ο πρώτος σταθμός της παρούσας πτυχιακής εργασίας με τελικό στόχο την κατανόηση του τρόπου με τον οποίο τα δεδομένα και τα νευρωνικά δίκτυα συνδέονται για να αλληλοεπιδρούν με την τεχνολογία της οπτικής αναγνώρισης.

# 1. ΔΕΔΟΜΕΝΑ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑ

Βρισκόμαστε σε μια εποχή που θα μπορούσε να χαρακτηριστεί εύκολα ως η χρυσή εποχή της πληροφορίας. Με 5,27 δισεκατομμύρια ανθρώπους (67,1% του συνολικού πληθυσμού) να κατέχουν ένα κινητό τηλέφωνο, 4,72 δισεκατομμύρια χρήστες του διαδικτύου, και 4,33 δισεκατομμύρια ενεργούς χρήστες στα μέσα κοινωνικής δικτύωσης (datareportal, 2021), θα μπορούσε κανείς να συμπεράνει πως υπάρχει ένας άπειρος όγκος από δεδομένα και πληροφορίες που περιμένουν να συλλεχθούν και να αναλυθούν. Στην τρέχουσα ενότητα γίνεται μια απόπειρα εξήγησης του τρόπου με τον οποίο τα δεδομένα συλλέγονται, αναλύονται και μετατρέπονται σε πληροφορίες.

## 1.1 ΟΡΙΣΜΟΣ

Δεδομένα και πληροφορίες: δύο λέξεις που θα μπορούσαν να χαρακτηριστούν όμοιες ή ακόμα και συνώνυμες, παρόλα αυτά υπάρχει μια λεπτή διαφορά μεταξύ των δύο όρων.

Σύμφωνα με το λεξικό της Οξφόρδης, η λέξη «Δεδομένο-α» ('Data' in Lexico Dictionaries, n.d) ορίζεται ως τα γεγονότα και τα στατιστικά στοιχεία, τα οποία έχουν συλλεχθεί συνολικά για αναφορά ή ανάλυση. Συγκεκριμένα στον τομέα της πληροφορικής, τα δεδομένα αναφέρονται σε ποσότητες, που αναπαριστούνται από αριθμούς, χαρακτήρες ή σύμβολα, τα οποία υφίστανται επεξεργασία από έναν υπολογιστή, ενώ ταυτόχρονα είναι δυνατή η αποθήκευση και η μετάδοση αυτών των δεδομένων με την μορφή ηλεκτρικών σημάτων.

Κατά κύριο λόγο, τα δεδομένα βρίσκονται σε μια ακατέργαστη (raw), και ανοργάνωτη (unorganized) μορφή. Μπορεί να αντιστοιχούν σε απλά γεγονότα ή στοιχεία, τα οποία πιθανώς να θεωρηθούν τυχαία, ή ακόμα και άχρηστα μέχρι να υποστούν μια οργάνωση ή μια επεξεργασία.

Όταν η επεξεργασία των δεδομένων έχει ολοκληρωθεί και τα δεδομένα βρίσκονται σε μια κατηγοριοποιημένη δομή, ή παρουσιάζονται επάνω σε κάποιο συγκεκριμένο θέμα, τότε τα δεδομένα μετατρέπονται σε πληροφορία. (Diffen, n.d)

## 1.2 ΕΙΔΗ ΔΕΔΟΜΕΝΩΝ

Υπάρχουν διάφορα είδη δεδομένων, από γραπτά κείμενα ή αριθμούς σε χαρτί, μέχρι και bits δεδομένων μέσα σε μνήμες ηλεκτρονικών συσκευών. Στα πλαίσια της πληροφορικής επεξεργασίας που μας ενδιαφέρει εδώ, ορισμένες κύριες κατηγορίες δεδομένων θα μπορούσαν να χαρακτηριστούν οι ακόλουθες: (Bridgwater, 2018)

### **Μεγάλα Δεδομένα (Big Data):**

Τα Μεγάλα Δεδομένα ορίζονται ως ένας τεράστιος όγκος δεδομένων, τα οποία συχνά έχουν μέγεθος από μερικά terabytes σε εκατοντάδες zettabytes, ενώ σε πολλές περιπτώσεις η παραγωγή τέτοιου μεγάλου όγκου δεδομένων μπορεί να πραγματοποιηθεί σε πραγματικό χρόνο. Για να μπορέσουμε να κατανοήσουμε το πραγματικό μέγεθος, και την σημασία που έχουν τα μεγάλα δεδομένα, ακολουθούν ορισμένα στατιστικά στοιχεία:

1. Σε χρονικό διάστημα μιας ημέρας πραγματοποιούνται πάνω από 3,5 δισεκατομμύρια αναζητήσεις μέσω της μηχανής αναζήτησης Google. (Internet Live Stats, n.d)
2. Κάθε λεπτό της ημέρας, οι καταναλωτές ξοδεύουν ένα εκατομμύριο δολάρια ηλεκτρονικά, πραγματοποιούν 1,4 εκατομμύρια βιντεοκλήσεις, στέλνουν 150 χιλιάδες μηνύματα στο Facebook και στέλνουν 69 χιλιάδες αιτήσεις εργασίας στο LinkedIn. (Domo, 2020)

Η συγκεκριμένη κατηγορία δεδομένων αποτελεί το «καύσιμο» που τροφοδοτεί την μηχανική μάθηση, ένας τομέας της τεχνητής νοημοσύνης που εστιάζει στη χρήση κυρίως στατιστικών τεχνικών για την ανάλυση των δεδομένων και τη δόμηση γνώσης που εξάγεται από αυτά. Την ίδια στιγμή, οι πολυεθνικές εταιρίες και οι μεγάλες διαδικτυακές υπηρεσίες προσπαθούν να επεξεργαστούν τα μεγάλα αυτά δεδομένα που συλλέγουν από τους χρήστες και πελάτες τους για να μπορέσουν να ανταπεξέλθουν στις ανάγκες και προτιμήσεις των καταναλωτών.

### **Ανοιχτά Δεδομένα (Open Data):**

Τα Ανοιχτά Δεδομένα αποτελούν μια κατηγορία δεδομένων που είναι ελεύθερα σε οποιονδήποτε ενδιαφερόμενο όσον αφορά τα πνευματικά δικαιώματα. Θα μπορούσε δηλαδή ο οποιοσδήποτε να αποκτήσει πρόσβαση στα συγκεκριμένα δεδομένα, να τα

επεξεργαστεί, ή ακόμα και να εξάγει συμπεράσματα έπειτα από ανάλυση επάνω σε αυτά. Πολλές φορές τα Μεγάλα Δεδομένα που προέρχονται από επιστημονικές έρευνες ή ακόμα και από οργανισμούς, μπορούν να χαρακτηριστούν και ως Ανοιχτά Δεδομένα, εφόσον είναι άμεσα και ελεύθερα διαθέσιμα. Στη συγκεκριμένη περίπτωση έχει μεγαλύτερη σημασία η δυνατότητα πρόσβασης στα συγκεκριμένα δεδομένα, παρά το συνολικό τους μέγεθος.

### **Δεδομένα Πραγματικού Χρόνου (Real Time Data):**

Αποτελούν δεδομένα τα οποία δέχονται επεξεργασία και ανάλυση σε πραγματικό χρόνο από υπολογιστικά συστήματα. Κατά κύριο λόγο, τέτοια δεδομένα μπορούν να αξιοποιηθούν στην ανάλυση και στην πρόβλεψη της αύξησης ή πτώσης μετοχών στο χρηματιστήριο, ή ακόμα και στην παρακολούθηση της κίνησης σε κάποιον πολυσύχναστο δρόμο.

### **Επιχειρησιακά Δεδομένα (Operational Data):**

Ίσως τα σημαντικότερα δεδομένα που κατέχει μια επιχείρηση. Συχνά περιλαμβάνουν στατιστικά στοιχεία της εταιρίας, πληροφορίες προμηθευτών, δεδομένα λογιστηρίου, ακόμα και πληροφορίες για τις κινήσεις των ανταγωνιστών τους.

## **1.3 ΤΡΟΠΟΙ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΜΩΝ**

Προτού το διαδίκτυο επικρατήσει για τα καλά τόσο στην καθημερινή ζωή μας, όσο και στην ζωή των επιχειρήσεων, διάφορες εταιρίες συνήθιζαν να αξιοποιούν τεχνικές συλλογής δεδομένων που δύσκολα θα συναντούσε κάποιος στις μέρες μας. Συγκεκριμένα, μια επιχείρηση μπορούσε να δημιουργήσει ερωτηματολόγια ή ακόμα και να πάρει συνεντεύξεις από άτομα που βρίσκονται σε μια συγκεκριμένη ομάδα, για την οποία υπάρχει κάποιο ενδιαφέρον από την πλευρά μιας εταιρίας.

Στις μέρες μας μια επιχείρηση μπορεί κυριολεκτικά να συλλέξει τεράστιους όγκους δεδομένων από οπουδήποτε. Χρησιμοποιώντας το διαδίκτυο και εργαλεία ανάλυσης, μία κινητή συσκευή, μια ιστοσελίδα, ακόμα και η δραστηριότητα σε έναν server θα μπορούσαν να χαρακτηριστούν αξιόπιστες πηγές συνεχούς συλλογής πληροφορίας. (McLaughlin, 2020)



## 1.4 ΕΞΑΓΩΓΗ ΠΛΗΡΟΦΟΡΙΑΣ

Με την ολοκλήρωση της συλλογής δεδομένων από κάποια, ή κάποιες πηγές, τα δεδομένα βρίσκονται σε μια ακατέργαστη μορφή, από την οποία δεν είναι εύκολο να εξάγουμε πληροφορία. Για να επιτευχθεί η εξαγωγή της πληροφορίας από ένα σύνολο δεδομένων, θα πρέπει πρώτα να ολοκληρωθούν μια σειρά από βήματα: (Talend, n.d)

1. **Προετοιμασία Δεδομένων (Data Preparation).** Η προετοιμασία δεδομένων είναι το αμέσως επόμενο στάδιο από την συλλογή δεδομένων. Συχνά αποκαλείται ως το στάδιο προ-επεξεργασίας, και αναφέρεται κυρίως στον καθαρισμό και την οργάνωση που δέχονται τα δεδομένα για να εισέλθουν στο επόμενο βήμα. Συγκεκριμένα, τα δεδομένα ελέγχονται για τυχόν σφάλματα, με σκοπό να αφαιρεθούν αχρείαστα ή ελλιπή δεδομένα και να παραμείνουν μόνο τα δεδομένα υψηλής ποιότητας.
2. **Είσοδος Δεδομένων (Data Input).** Τα καθαρισμένα δεδομένα εισάγονται σε ένα σύστημα Διαχείρισης Πελατειακών Σχέσεων (CRM – Customer Relationship Model), όπου και αρχίζουν να παίρνουν μια πιο ξεκάθαρη μορφή από την οποία μπορεί να ξεκινήσει η εξαγωγή χρήσιμης πληροφορίας.
3. **Επεξεργασία (Processing).** Τα δεδομένα, έχοντας πλέον εισαχθεί σε κάποιο υπολογιστικό σύστημα, βρίσκονται πλέον στο στάδιο της επεξεργασίας. Η επεξεργασία πραγματοποιείται κυρίως χρησιμοποιώντας τεχνικές και μεθόδους μηχανικής μάθησης, αν και η ίδια η επεξεργασία μπορεί να διαφέρει ελαφρώς ανάλογα με το είδος, την πηγή των δεδομένων που υπόκεινται την επεξεργασία, αλλά και από τον τρόπο που επιθυμούμε να αξιοποιήσουμε την εξαγόμενη πληροφορία.
4. **Έξοδος / Διερμηνευση Δεδομένων (Data Output / Interpretation).** Στο τρέχον στάδιο, τα αρχικά δεδομένα βρίσκονται σε μια ιδιαίτερα απλή μορφή, και η πληροφορία αυτών των δεδομένων μπορεί να γίνει εύκολα κατανοητή από άτομα που δεν σχετίζονται με την ανάλυση δεδομένων. Συχνά βρίσκονται σε μορφή γραφήματος, εικόνας, κειμένου, ακόμα και βίντεο.
5. **Αποθήκευση Δεδομένων (Data Storage).** Το τελικό στάδιο της επεξεργασίας των δεδομένων είναι η αποθήκευση. Παρόλο που ένα ποσοστό της εξαγόμενης πληροφορίας μπορεί να αξιοποιηθεί άμεσα, ένα σημαντικό ποσοστό ενδέχεται να χρησιμοποιηθεί μελλοντικά. Η σωστή αποθήκευση της πληροφορίας μπορεί

να παρέχει άμεση και εύκολη πρόσβαση σε μέλη ενός οργανισμού όποτε αυτό χρειαστεί.

## 1.5 ΑΞΙΟΠΟΙΗΣΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Για μια επιχείρηση ή έναν οργανισμό, η πληροφορία ίσως είναι το πολυτιμότερο στοιχείο που θα μπορούσε να κατέχει. Η πληροφορία όταν αξιοποιηθεί κατάλληλα θεωρείται γνώση, και η γνώση με τη σειρά της θεωρείται δύναμη. Κάθε επιχείρηση έχει έναν στόχο, το κέρδος. Σε μια εποχή, όπου υπάρχουν εκατομμύρια μικρές και μεγάλες επιχειρήσεις, ο ανταγωνισμός είναι τεράστιος. Για να μπορέσει μια μικρομεσαία επιχείρηση να ανταπεξέλθει, χρειάζεται τη δύναμη που της παρέχει η πληροφορία.

Οι εταιρίες χρησιμοποιούν την πληροφορία για να λάβουν ένα σημαντικό πλεονέκτημα απέναντι στους ανταγωνιστές τους, ενώ ταυτόχρονα προσπαθούν να γίνουν πιο ελκυστικές προς τους πελάτες τους. Παρόλα αυτά, η πληροφορία είναι εξίσου σημαντική για τη σωστή και αποτελεσματική λειτουργία της ίδιας της εταιρίας, αξιοποιώντας αυτή τη δύναμη μέσω των ακόλουθων ενεργειών: (Williams, 2019)

1. **Εισαγωγές (Inbound Logistics):** Η διαδικασία της εισόδου πρώτων υλών. Εδώ η πληροφορία μπορεί να αυξήσει την αποτελεσματικότητα των εισαγωγών με συστήματα διαχείρισης εφοδιαστικής αλυσίδας (supply-chain management systems), τα οποία εξυπηρετούν τη σωστή διαχείριση του αποθέματος.
2. **Διεργασίες (Operations):** Περιλαμβάνει οποιοδήποτε τμήμα της επιχείρησης που συμμετέχει στη μετατροπή πρώτων υλών σε ένα τελικό προϊόν, ή υπηρεσία. Η πληροφορία μπορεί να παρέχει ενίσχυση στην αποτελεσματικότητα των διεργασιών, καθώς επίσης και στην καινοτομία.
3. **Εξαγωγές (Outbound Logistics):** Όπως και στις εισαγωγές, έτσι και στις εξαγωγές, η πληροφορία είναι σε θέση να βελτιώσει την αποτελεσματικότητα της μεταφοράς του τελικού προϊόντος προς τον πελάτη, χρησιμοποιώντας τεχνικές όπως τον έλεγχο αποθέματος σε πραγματικό χρόνο.
4. **Πωλήσεις / Μάρκετινγκ (Sales / Marketing):** Οι λειτουργίες που θα προσελκύσουν τους καταναλωτές ώστε να αγοράσουν τα προϊόντα μιας επιχείρησης. Η πληροφορία θεωρείται κρίσιμη σε ολόκληρο το τμήμα των πωλήσεων και του μάρκετινγκ. Μέσα από ηλεκτρονικές διαφημίσεις και

δημοσκοπήσεις, η πληροφορία μπορεί να αξιοποιηθεί στον τομέα του σχεδιασμού προϊόντων, και παράλληλα να επιτρέψει στην επιχείρηση να βρίσκεται πιο κοντά στους καταναλωτές τους.

## **2. ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ**

Στον κόσμο της πληροφορικής, τα νευρωνικά δίκτυα χαρακτηρίζονται ως το αποτέλεσμα της σύνδεσης μεταξύ του τρόπου σκέψης, κατανόησης και μάθησης ενός ανθρωπίνου εγκεφάλου με ένα υπολογιστικό σύστημα. Σε αυτή την ενότητα πραγματοποιείται μια απόπειρα εξήγησης του κόσμου των νευρωνικών δικτύων, από τη γενική περιγραφή του τρόπου λειτουργίας τους, μέχρι και ορισμένους από τους τρόπους αξιοποίησής τους.

### **2.1 ΤΙ ΕΙΝΑΙ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ**

Είναι υπολογιστικά συστήματα, τα οποία ορίζονται ως μια σειρά από αλγόριθμους μηχανικής μάθησης, οι οποίοι αποσκοπούν να ανακαλύψουν κρυμμένες σχέσεις σε ένα σετ δεδομένων μέσω μιας διαδικασίας που μιμείται τον τρόπο λειτουργίας του ανθρωπίνου εγκεφάλου. Συγκεκριμένα, τα νευρωνικά δίκτυα αναφέρονται σε συστήματα νευρώνων, τα οποία είναι σε θέση να επιλύσουν προβλήματα, δίχως αυτά να έχουν προγραμματιστεί να ακολουθούν συγκεκριμένες οδηγίες. (Chen, 2020)

### **2.2 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ**

Τα νευρωνικά δίκτυα είναι ένας τρόπος πραγματοποίησης της μηχανικής μάθησης, δηλαδή της διαδικασίας στην οποία ένας αλγόριθμος λαμβάνει δεδομένα εκπαίδευσης, από τα οποία καλείται να μάθει. Αυτά τα δεδομένα πρέπει να περιέχουν μια σωστή «απάντηση», δηλαδή έναν στόχο ή μια κατηγορία που χαρακτηρίζουν πλήρως τα δεδομένα που εισάγονται στον αλγόριθμο. Ο αλγόριθμος μάθησης εντοπίζει μοτίβα και ακολουθίες που ταυτίζουν τα δεδομένα εισόδου σε κάποια κατηγορία, δηλαδή στην απάντηση που προσπαθούμε να προβλέψουμε κατά την ολοκλήρωση της εκπαίδευσης, όταν ο αλγόριθμος εξάγει ένα μοντέλο μηχανικής μάθησης. Αυτό το μοντέλο αποτελεί

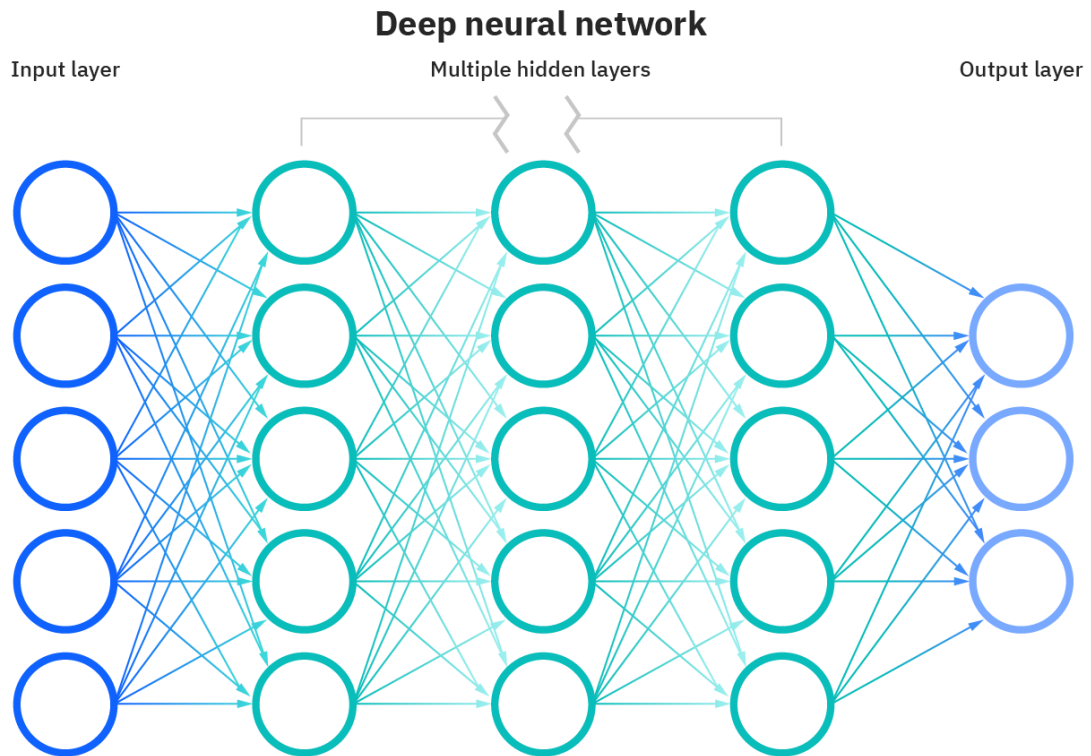
το τελικό αποτέλεσμα της εκπαίδευσης, το οποίο μπορεί να χρησιμοποιηθεί για την πρόβλεψη νέων δεδομένων, από τα οποία δεν είναι γνωστή η κατηγορία τους.

Για παράδειγμα, ένας αλγόριθμος αναγνώρισης αντικειμένων μπορεί να έχει αναλύσει χιλιάδες φωτογραφίες από αυτοκίνητα, σπίτια, έπιπλα, κλπ. με αποτέλεσμα να μπορεί να αναγνωρίσει μοτίβα και ακολουθίες σε αυτές τις εικόνες, και στη συνέχεια να είναι σε θέση να τις συσχετίσει με τις κατηγορίες στις οποίες ανήκουν. Ο αλγόριθμος υλοποιεί ένα εκπαιδευμένο μοντέλο, το οποίο στη συνέχεια χρησιμοποιείται για την συσχέτιση νέων εισόδων σε κάποια κατηγορία από αυτές που γνωρίζει.

Δεδομένου ότι τα νευρωνικά δίκτυα είναι σχεδιασμένα να προσομοιάζουν τη λειτουργία του ανθρώπινου εγκεφάλου, ένα δίκτυο μπορεί να αποτελείται από χιλιάδες, ή ακόμα και εκατομμύρια κόμβους, οι οποίοι είναι πυκνά συνδεδεμένοι μεταξύ τους. Στις μέρες μας, τα περισσότερα νευρωνικά δίκτυα είναι οργανωμένα σε στρώματα (layers) από νευρώνες που ονομάζονται κόμβοι, και χρησιμοποιούν το μοντέλο της εμπρόσθιας τροφοδοσίας (feed-forward) των σημάτων εισόδου. Αυτό σημαίνει πως τα δεδομένα κινούνται προς μια μόνο κατεύθυνση. Δηλαδή, ένας κόμβος μπορεί να είναι συνδεδεμένος με αρκετούς άλλους κόμβους από μια προηγούμενη στρώση κόμβων για να λαμβάνει δεδομένα, και ταυτόχρονα να συνδέεται με άλλους κόμβους σε μια επόμενη στρώση για να στέλνει δεδομένα. (Hardesty, 2017)

### **2.2.1 Κρυμμένες Στρώσεις ή Στρώματα**

Στα νευρωνικά δίκτυα, τα προαναφερόμενα στρώματα ονομάζονται κρυμμένα (Hidden Layers) και βρίσκονται μεταξύ της εισόδου και της εξόδου του δικτύου, στις οποίες νευρωνικό δίκτυο εισάγει βάρη (weights) στις εισόδους των κόμβων, και τις προωθεί σε μια συνάρτηση ενεργοποίησης (activation function) ως έξοδο. Με λίγα λόγια, οι κρυμμένες στρώσεις πραγματοποιούν μετασχηματισμούς (γραμμικούς ή μη γραμμικούς) επάνω στις εισόδους που έχουν εισαχθεί στο δίκτυο. Αυτές οι στρώσεις ποικίλουν ανάλογα με την αρχιτεκτονική του νευρωνικού δικτύου.



Εικόνα 1: Βαθύ νευρωνικό δίκτυο.

Θα μπορούσαμε να χαρακτηρίσουμε τις κρυφές στρώσεις ως μαθηματικές συναρτήσεις, οι οποίες είναι σχεδιασμένες να παράγουν συγκεκριμένες εξόδους ανάλογα με το επιθυμητό αποτέλεσμα. Αυτές οι συναρτήσεις είναι ιδιαίτερα χρήσιμες όταν η επιθυμητή έξοδος του αλγορίθμου είναι πιθανότητα, διότι είναι σε θέση να επεξεργαστούν την είσοδο και να την εξάγουν ως μια τιμή μεταξύ 0 και 1, δηλαδή το εύρος στο οποίο ορίζεται η πιθανότητα.

Κάθε στρώση μπορεί να εξειδικεύεται στην παραγωγή μιας συγκεκριμένης εξόδου. Για παράδειγμα, οι συναρτήσεις μιας στρώσης που χρησιμοποιούνται για να αναγνωρίζουν ανθρώπινα μάτια και αφτιά, μπορούν να χρησιμοποιηθούν σε συνδυασμό με επόμενες στρώσεις για να αναγνωρίσουν πρόσωπα σε εικόνες. (DeepAI a, n.d)

### 2.2.2 Βάρη και Bias

Στα νευρωνικά δίκτυα τα βάρη (weights) λειτουργούν σαν παράμετροι που τροποποιούν τα δεδομένα εισόδου στις κρυφές στρώσεις και στο στρώμα εξόδου του δικτύου. Κάθε κόμβος περιλαμβάνει την είσοδο, το συναπτικό βάρος της εισόδου, και

μια τιμή πόλωσης (bias). Κάθε είσοδος που εισέρχεται στον νευρώνα πολλαπλασιάζεται με την τρέχουσα τιμή του βάρους της εισόδου. Στο άθροισμα όλων των εισόδων προστίθεται το γινόμενο της τιμής bias και του συναπτικού της βάρους, και το τελικό αποτέλεσμα αποτελεί την είσοδο στη συνάρτηση ενεργοποίησης του νευρώνα στην έξοδο του κόμβου. Αυτή η έξοδος μπορεί στη συνέχεια να προχωρήσει στον επόμενο κόμβο του δικτύου. (DeerAI b, n.d)

Το bias λειτουργεί ως μια έξτρα ώθηση επάνω σε κάθε κόμβο. Μπορεί να έχει μια συγκεκριμένη, σταθερή τιμή σε ολόκληρο το δίκτυο, ή μπορεί να μεταβάλλεται κατά τη διάρκεια της εκπαίδευσης, ανάλογα με τον τύπο του δικτύου. Το bias αντιπροσωπεύει πόσο μακριά είναι η πρόβλεψη του κόμβου για μια είσοδο από την επιθυμητή της τιμή.

Τα βάρη μπορούν να χαρακτηριστούν ως η γνώση που έχει ένας κόμβος του δικτύου, αφού είναι σε θέση να αλλάξουν ριζικά την τιμή της εισόδου, και στη συνέχεια να το προωθήσουν προς την έξοδο. Όταν το βάρος είναι χαμηλό, τότε η τιμή της εισόδου δεν θα επηρεαστεί, και θα παραμείνει σχεδόν ίδια κατά την έξοδο. Ταυτόχρονα ένα υψηλό βάρος μπορεί να καθορίσει σε μεγάλο βαθμό την τελική έξοδο του δικτύου.

### **2.2.3 Συνάρτηση Ενεργοποίησης**

Η συνάρτηση ενεργοποίησης (Activation Function) είναι, συνήθως, μια μη-γραμμική μαθηματική συνάρτηση, η οποία λειτουργεί σαν μια «πύλη» που επιτρέπει την ενεργοποίηση, δηλαδή την λειτουργία ενός νευρώνα όταν η τιμή αυτής της εισόδου είναι μεγαλύτερη ενός συγκεκριμένου ορίου. Στη συνέχεια η είσοδος προωθείται στην έξοδο του νευρώνα, και κατά συνέπεια, στην επόμενη στρώση νευρώνων.

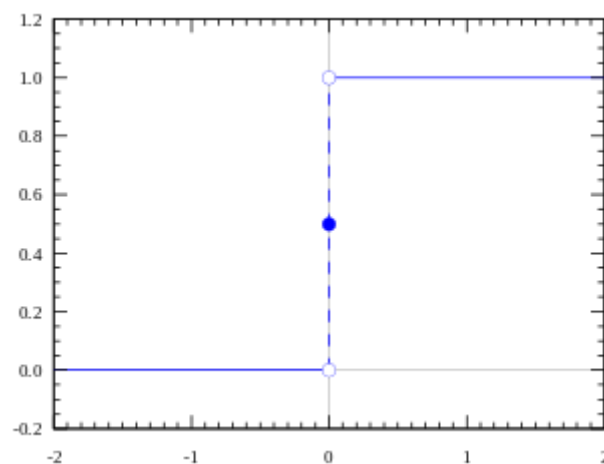
Αυτές οι συναρτήσεις είναι ιδιαίτερα χρήσιμες στα νευρωνικά δίκτυα, μιας και η μη-γραμμικότητά τους, επιτρέπει στο δίκτυο να εκπαιδευτεί επάνω σε ισχυρότερες λειτουργίες. Αν αφαιρούσαμε τις συναρτήσεις ενεργοποίησης από ένα νευρωνικό δίκτυο, τότε το δίκτυο θα μετατρέποταν σε ένα δίκτυο απλής, γραμμικής λειτουργίας. Ως αποτέλεσμα, το δίκτυο δεν θα ήταν πλέον ικανό να πραγματοποιήσει περίπλοκες λειτουργίες, όπως για παράδειγμα η αναγνώριση εικόνων. (Wood, n.d)

Ορισμένες από τις δημοφιλέστερες συναρτήσεις ενεργοποίησης μπορούν να χαρακτηριστούν οι επόμενες, παρόλα μια συνάρτηση μπορεί να φανεί χρησιμότερη από

κάποια άλλη, αναλόγως πάντα με το πρόβλημα που επιχειρούμε να λύσουμε χρησιμοποιώντας ένα νευρωνικό δίκτυο.

### Βηματική Συνάρτηση (Step Function)

Έσος η πιο απλή μορφή συνάρτησης όταν αναφερόμαστε σε κάποιο όριο. Όταν η τιμή της εισόδου είναι μεγαλύτερη από κάποια συγκεκριμένη τιμή ορίου, τότε η συνάρτηση ενεργοποιείται, δηλαδή βρίσκεται σε τιμή 1. Αντιθέτως, όταν η είσοδος είναι μικρότερη του ορίου, η συνάρτηση παραμένει ανενεργή, δηλαδή 0.

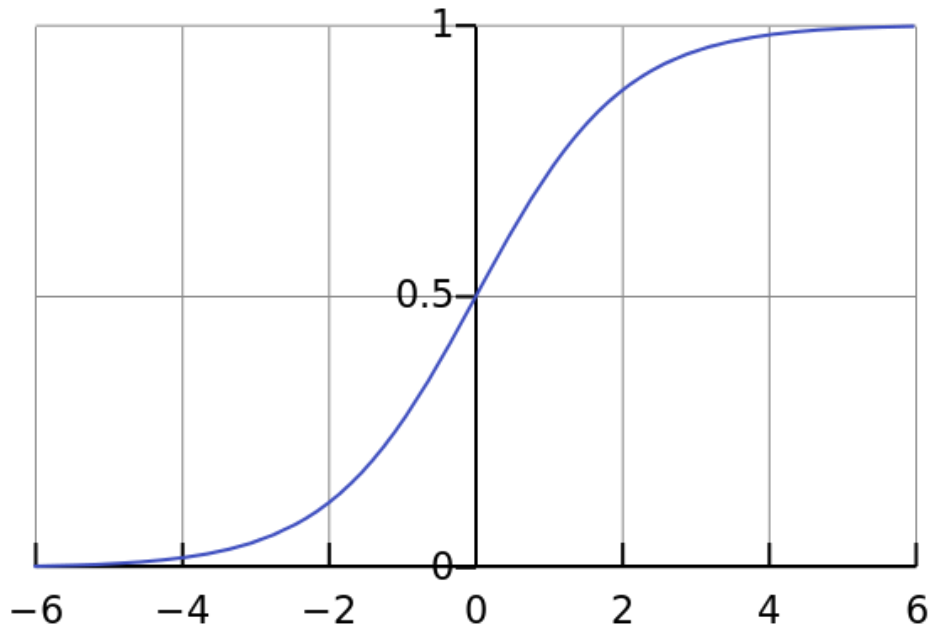


Εικόνα 2: Βηματική συνάρτηση.

Μια τέτοιου είδους συνάρτηση μπορεί να λειτουργήσει εξαιρετικά σε ένα σύστημα δυαδικής κατηγοριοποίησης (binary classification), όπου η έξοδος μπορεί να είναι μόνο μια από δύο πιθανές κατηγορίες (κατηγορία Α ή κατηγορία Β). Σε συστήματα που επιθυμούμε να κατηγοριοποιήσουμε μια είσοδο σε παραπάνω από δύο κατηγορίες (κατηγορία Α, κατηγορία Β, κατηγορία Γ, κλπ.), τότε τα πράγματα περιπλέκονται όσον αφορά στη χρήση της βηματικής συνάρτησης, μιας και υπάρχει η πιθανότητα να ενεργοποιηθούν περισσότεροι από έναν μόνο νευρώνα. Μια είσοδος θα κατηγοριοποιούνταν σε δύο ή ακόμα και περισσότερες κατηγορίες ταυτόχρονα, κάτι που θα έκανε την τελική απόφαση ιδιαίτερα δύσκολη.

### Σιγμοειδής Συνάρτηση (Sigmoid Function)

Για να αντιμετωπιστεί το πρόβλημα που παρουσιάζει η βηματική συνάρτηση, δηλαδή την κατηγοριοποίηση σε περισσότερες από δύο κατηγορίες, μπορούμε να χρησιμοποιήσουμε τη σιγμοειδή συνάρτηση.



Εικόνα 3: Σιγμοειδής συνάρτηση.

Όντας μια από τις πιο διαδεδομένες συναρτήσεις ενεργοποίησης, η σιγμοειδής συνάρτηση περιλαμβάνει ορισμένες ιδιαιτερότητες που την ξεχωρίζουν από άλλες συναρτήσεις. Μεταξύ των σημείων  $X = -2$  και  $X = 2$ , παρατηρούμε πως η συνάρτηση είναι αρκετά απότομη. Αν μια τιμή  $X$  που βρίσκεται μεταξύ αυτών των δύο σημείων, αλλάξει ελαφρώς προς τα δεξιά ή τα αριστερά, θα υπάρξει ιδιαίτερα μεγάλη αλλαγή στο  $Y$  της συνάρτησης. Θα μπορούσαμε να πούμε πως η σιγμοειδής συνάρτηση έχει μια τάση να φέρνει τις  $Y$  τιμές πιο κοντά στα δύο άκρα της καμπύλης, κάτι που μπορεί να ξεκαθαρίσει την πρόβλεψη της κατηγοριοποίησης.

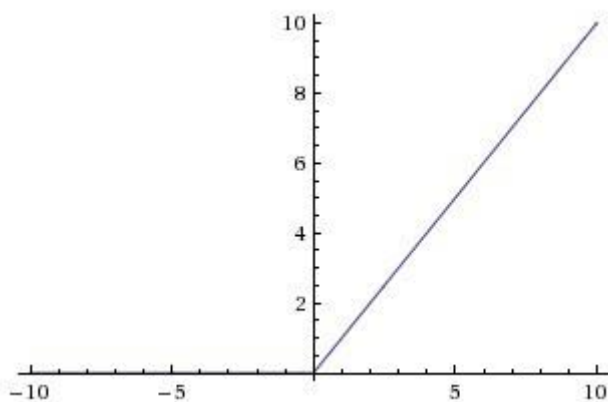
Αν και η σιγμοειδής συνάρτηση είναι αρκετά εύχρηστη, όταν μια τιμή  $Y$  βρίσκεται πολύ κοντά στο 0 ή το 1, τείνει να επηρεάζεται ελάχιστα από κάποια αλλαγή στο  $X$ , ως αποτέλεσμα το νευρωνικό δίκτυο να δυσκολεύεται αρκετά κατά την διάρκεια της εκπαίδευσης.



## Συνάρτηση ReLU (Rectified Linear Unit)

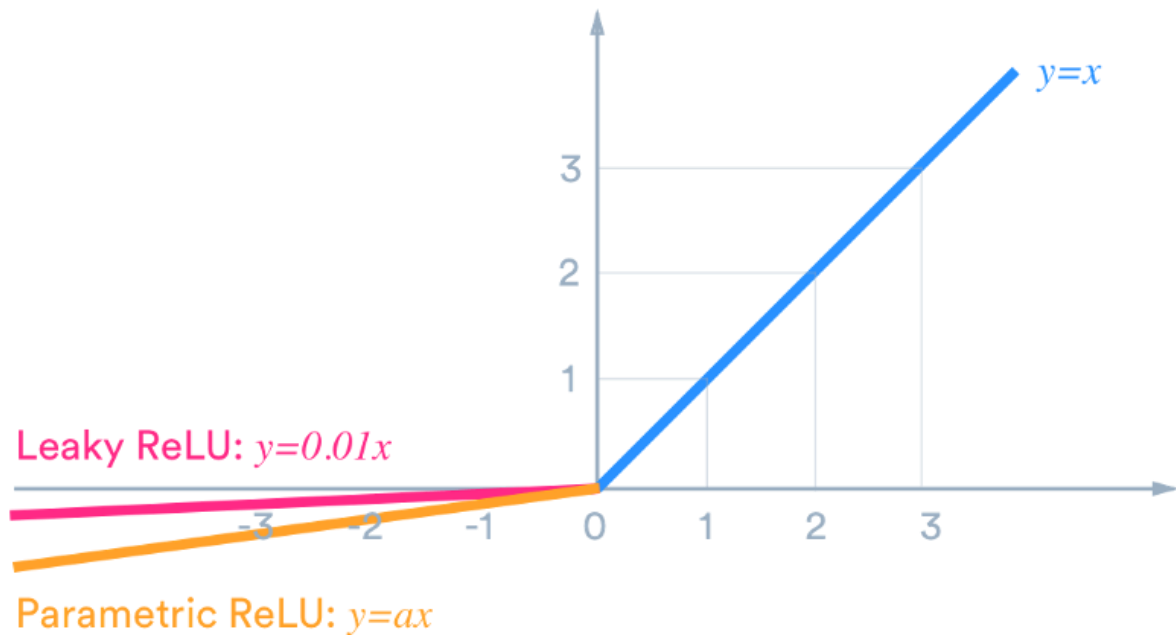
Σε ένα νευρωνικό δίκτυο εκατοντάδων χιλιάδων ή ακόμα και εκατομμυρίων νευρώνων, αν χρησιμοποιούσαμε μια συνάρτηση ενεργοποίησης όπως την σιγμοειδή, λόγω της ιδιαιτερότητας της συγκεκριμένης συνάρτησης να ενεργοποιείται στις θετικές τιμές, θα υπήρχε ενεργοποίηση σχεδόν όλων των νευρώνων. Αυτό σημαίνει πως σχεδόν όλοι οι ενεργοποιημένοι νευρώνες θα δεχόντουσαν επεξεργασία για να υπολογιστεί η έξοδος του δικτύου, κάτι που θα χαρακτήριζε το δίκτυο αρκετά πυκνό.

Η μεγάλη πυκνότητα του δικτύου μπορεί να φέρει αρνητικά αποτελέσματα τόσο στην ταχύτητα αλλά και στην αποδοτικότητα του. Ιδανικά επιθυμούμε ένα μέρος των νευρώνων να μην ενεργοποιούνται. Για να επιτευχθεί αυτό, μπορούν να χρησιμοποιηθούν συναρτήσεις όπως η ReLU.



Εικόνα 4: Συνάρτηση ReLU.

Χρησιμοποιώντας την ReLU, όλοι οι νευρώνες που έχουν αρνητική τιμή δεν θα ενεργοποιηθούν, κάτι που σημαίνει πως το δίκτυο γίνεται ελαφρύτερο. Αυτό ακριβώς είναι και το μεγαλύτερο μειονέκτημα της ReLU. Όταν ένας νευρώνας βρίσκεται στην αρνητική πλευρά της συνάρτησης, θεωρείται εξαιρετικά απίθανη η ανάκαμψη του, κάτι που σημαίνει πως ο νευρώνας θεωρείται «νεκρός». Τέτοιοι νευρώνες δεν βοηθούν στην κατηγοριοποίηση μιας εισόδου, και στην ουσία αχρηστεύονται. Κατά τη διάρκεια της εκπαίδευσης, ένα μεγάλο ποσοστό του δικτύου θεωρείται πλέον ανενεργό. Συνήθως αυτό το φαινόμενο παρατηρείται όταν ο ρυθμός εκπαίδευσης (learning rate) του μοντέλου είναι ιδιαίτερα υψηλός ή υπάρχει ένα υψηλό αρνητικό bias. (Sharma, 2017)



Εικόνα 5: Συναρτήσεις Leaky ReLU και Parametric ReLU.

Για να επιλυθεί το πρόβλημα των «νεκρών» νευρώνων, υπάρχουν ορισμένες τροποποιήσεις στην συνάρτηση της ReLU, οι οποίες προσπαθούν να κρατήσουν τις τιμές των νευρώνων μακριά από την τιμή 0. Για παράδειγμα, οι συναρτήσεις Leaky ReLU και Parametric ReLU μπορούν να θεωρηθούν καλές εναλλακτικές της ReLU. (Liu, 2017)

## 2.3 ΠΩΣ ΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΕΞΑΡΤΩΝΤΑΙ ΑΠΟ ΤΑ ΔΕΔΟΜΕΝΑ

Όπως αναφέραμε στο Κεφάλαιο 1, ο κόσμος μας είναι κυριολεκτικά κυριευμένος από εξωπραγματικούς όγκους δεδομένων, εκ των οποίων μόνο ένα μικρό ποσοστό είναι σωστά δομημένο και κατηγοριοποιημένο.

Οι αλγόριθμοι μηχανικής μάθησης συνήθως εντάσσονται σε δύο κατηγορίες, στους αλγορίθμους μάθησης με επίβλεψη (supervised learning) και μάθησης χωρίς επίβλεψη (unsupervised learning).

Στους αλγορίθμους μάθησης με επίβλεψη, τα δεδομένα που εισάγονται σε ένα νευρωνικό δίκτυο έχουν ελεγχθεί, οργανωθεί, και κατηγοριοποιηθεί εκ των προτέρων,

προτού ξεκινήσει η διαδικασία της εκπαίδευσης. Η κεντρική ιδέα είναι ότι τα υπολογιστικά συστήματα μπορούν να φτάσουν στο σημείο όπου είναι σε θέση να ταξινομήσουν τα δεδομένα εντοπίζοντας κοινά στοιχεία σε μέλη μιας συγκεκριμένης κατηγορίας εικόνων, τα οποία δεν εντοπίζονται σε άλλες κατηγορίες. Ο απώτερος σκοπός αυτού του είδους εκπαίδευσης είναι η χρήση αυτών των στοιχείων που έχει εντοπίσει το δίκτυο, έτσι ώστε να ενισχυθεί σημαντικά η ακρίβεια αναγνώρισης και κατηγοριοποίησης αυτών των δεδομένων που το δίκτυο δεν έχει ξαναδεί. Όσο περισσότερα δεδομένα επεξεργάζεται ένα δίκτυο κατά την εκπαίδευση του, και όσο μεγαλύτερο και ποικιλόμορφο είναι το σύνολο των δεδομένων (dataset) που αξιοποιείται, τόσο καλύτερη ακρίβεια αναγνώρισης μπορεί να αποκτήσει το νευρωνικό δίκτυο.

Αυτή η τεχνική είναι ιδιαίτερα χρήσιμη στην εκπαίδευση συστημάτων για αναγνώριση οπτικών μέσων, όπως εικόνες, βίντεο, γραφικοί χαρακτήρες, κλπ. Επιπλέον, βρισκόμαστε σε μια εποχή, στην οποία ένα δίκτυο είναι σε θέση να αναγνωρίσει εικόνες εξίσου καλά, ή ακόμα και καλύτερα από ανθρώπους.

Από την αντίπερα όχθη, η ιδέα της μάθησης χωρίς επίβλεψη θα μπορούσε να χαρακτηριστεί ως μια πιο «αναρχική» προσέγγιση στον τρόπο με τον οποίο ένα δίκτυο εκπαιδεύεται. Συγκεκριμένα, η ιδέα είναι ότι τα δεδομένα που εισάγονται στο μοντέλο δεν περιέχουν κάποια ένδειξη κατηγορίας. Το νευρωνικό δίκτυο, δηλαδή, αφήνεται ελεύθερο να εξερευνήσει τα δεδομένα που του έχουν δοθεί, με σκοπό να εντοπίσει από μόνο του πρότυπα και συσχετίσεις μεταξύ των δεδομένων, οργανώνοντας τα σε ομάδες (clustering). (Tanz & Cambron, 2017)

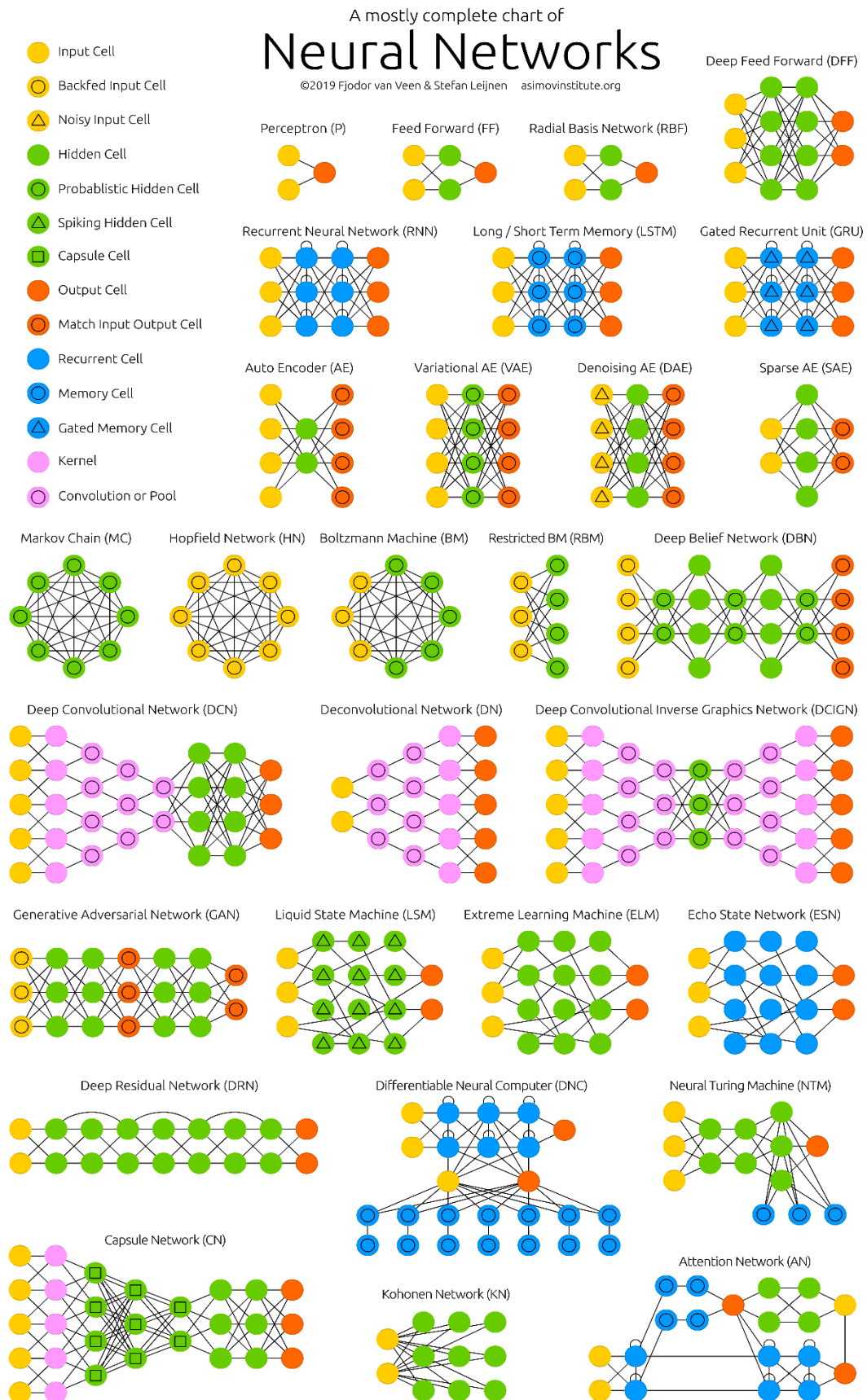
Έχοντας λοιπόν υπόψιν τους δύο κυριότερους τρόπους εκπαίδευσης νευρωνικών δικτύων, θα μπορούσαμε πλέον να συμπεράνουμε πως τα νευρωνικά δίκτυα, και ιδιαίτερα τα δίκτυα βαθιάς μάθησης που χρησιμοποιούν την τεχνική της εκπαίδευσης με επίβλεψη, εξαρτώνται σε τεράστιο βαθμό από σωστά κατηγοριοποιημένα δεδομένα για την σωστή τους εκπαίδευση. Το πρόβλημα όμως βρίσκεται στο γεγονός πως το πλήθος αυτών των συγκεκριμένων δεδομένων είναι αρκετά μικρό σε σύγκριση με το συνολικό όγκο δεδομένων που δημιουργούνται συνεχώς. Ενδέχεται το μέλλον να επιφυλάσσει την αυξημένη χρήση τεχνικών εκπαίδευσης χωρίς επίβλεψη. Ως αποτέλεσμα, πιθανώς να υπάρξει μια δραματική αύξηση στο ποσοστό αξιοποίησης των συνολικά παραγόμενων δεδομένων, μιας και τα συγκεκριμένα δίκτυα θα μπορούν να

λαμβάνουν τα μη κατηγοριοποιημένα ή μερικώς ομαδοποιημένα δεδομένα απευθείας από την πηγή τους.

## **2.4 ΕΙΔΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ**

Στις μέρες μας υπάρχουν πολλά, και διαφορετικά είδη νευρωνικών δικτύων, τα οποία μπορούν να κατηγοριοποιηθούν ανάλογα με την δομή τους, τη ροή των δεδομένων, το πλήθος και την πυκνότητα των νευρώνων που χρησιμοποιούνται, το συνολικό αριθμό κρυμμένων στρώσεων, κλπ.

Φυσικά, κάθε είδος ή ακόμα και κατηγορία νευρωνικών δικτύων χρησιμοποιείται σε συγκεκριμένες περιπτώσεις, ανάλογα πάντα με το πρόβλημα που επιθυμούμε να επιλύσουμε. Λαμβάνοντας τόσο τα θετικά, όσο και τα αρνητικά στοιχεία κάθε τύπου δικτύου, μπορούμε να πάρουμε την κατάλληλη απόφαση όσον αφορά στο κατάλληλο είδος δικτύου που θα καλύψει τις ανάγκες μας.



Εικόνα 6: Ένα σχεδόν ολοκληρωμένο διάγραμμα των νευρωνικών δικτύων.

Ορισμένα χαρακτηριστικά είδη νευρωνικών δικτύων θα μπορούσαν να θεωρηθούν τα επόμενα: (Great Learning Team, 2020)

### **Perceptron και δίκτυα εμπρόσθιας τροφοδότησης (feed-forward networks)**

Το μοντέλο του Perceptron αποτελεί ένα από τα παλαιότερα, και ταυτόχρονα απλούστερα μοντέλα νευρωνικών δικτύων. Το Perceptron μπορεί να χαρακτηριστεί ως ένα απλό είδος δικτύου εμπρόσθιας τροφοδότησης, και συγκεκριμένα εντάσσεται στην κατηγορία των γραμμικών ταξινομητών. Αυτό σημαίνει πως είναι σε θέση να κατηγοριοποιήσει μια είσοδο σε μια εκ των δύο πιθανών κατηγοριών, διαχωρίζοντας αυτές τις δύο κατηγορίες με μια ευθεία γραμμή. Επιπλέον, έχει την δυνατότητα να υλοποιεί λογικές πύλες (Logic Gates), όπως AND, OR ή NAND



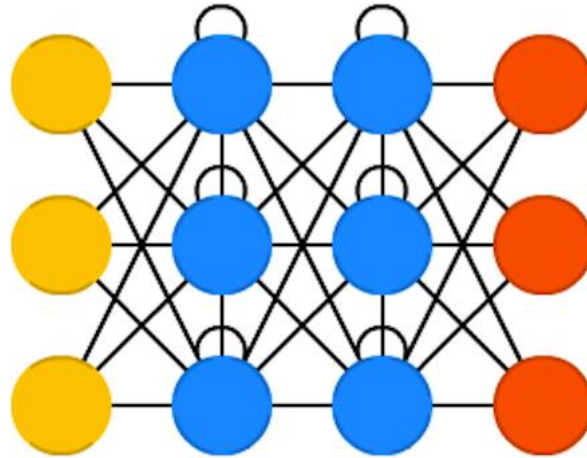
Εικόνα 7: Perceptron και δίκτυο εμπρόσθιας τροφοδοσίας.

Σε ένα δίκτυο εμπρόσθιας τροφοδοσίας, τα δεδομένα εισόδου κινούνται προς μια μόνο κατεύθυνση, περνώντας μέσα από τεχνητούς νευρώνες, και στη συνέχεια εξάγονται από κόμβους εξόδου. Ανάλογα με τον αριθμό των κρυφών στρώσεων, μπορούν να χαρακτηριστούν ως δίκτυα μονής στρώσεως ή πολλαπλών στρώσεων. Είναι σε θέση να πραγματοποιεί απλές κατηγοριοποιήσεις, όπου πολλοί παραδοσιακοί αλγόριθμοι ταξινόμησης αποτυγχάνουν, όπως αναγνώριση προσώπων, ακόμα και αναγνώριση φωνής. Παρόλα αυτά δεν είναι δυνατή η χρήση τους σε περιπτώσεις βαθιάς μάθησης (deep learning), καθώς πρόκειται για μια ρηχή αρχιτεκτονική.

### **Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks)**

Τα αναδρομικά νευρωνικά δίκτυα είναι σχεδιασμένα να αποθηκεύουν την έξοδο του κόμβου, και στην συνέχεια να την ανατροφοδοτούν στην είσοδο για να βοηθήσουν στην σωστή πρόβλεψη της επόμενης εξόδου. Συγκεκριμένα, τέτοιου είδους δίκτυα

αποθηκεύουν πληροφορίες που μπορούν να αξιοποιηθούν μελλοντικά, σε μια επόμενη είσοδο δεδομένων. Αν η πρόβλεψη του δικτύου αποδειχθεί πως είναι λανθασμένη, πραγματοποιούνται μικρές αλλαγές στα βάρη ανάλογα με το ρυθμό εκπαίδευσης (learning rate), οι οποίες στη συνέχεια ωθούν το δίκτυο να πραγματοποιεί σωστές προβλέψεις.



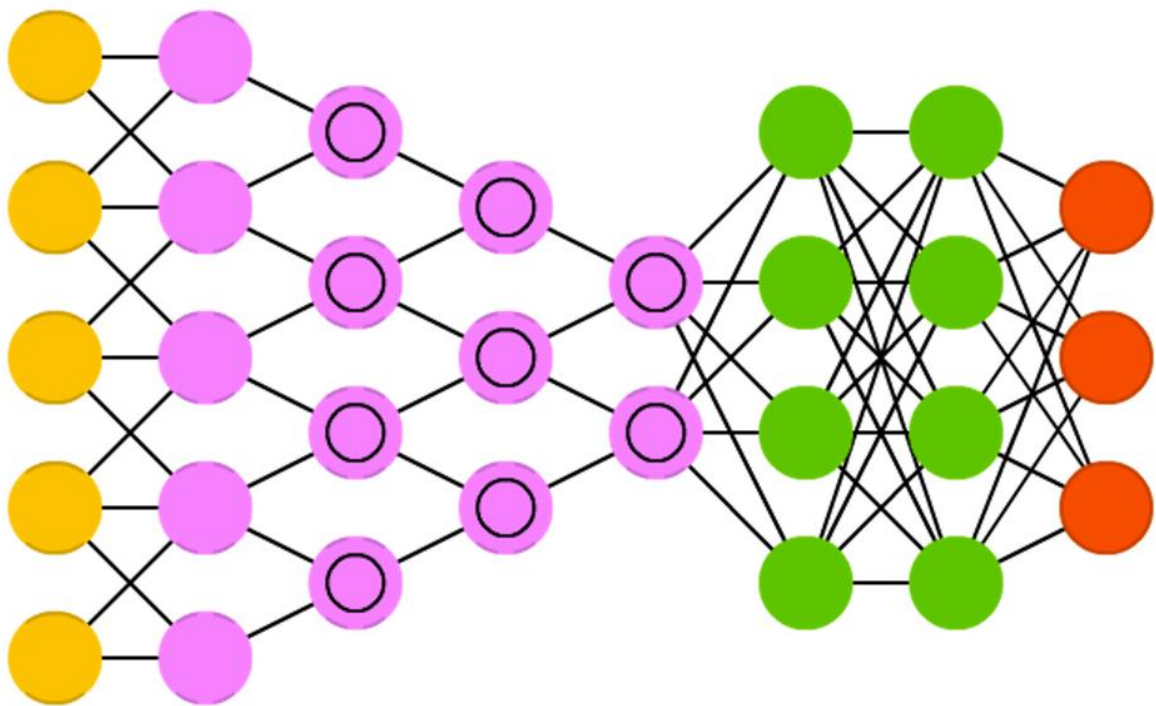
Εικόνα 8: Επαναλαμβανόμενο νευρωνικό δίκτυο.

Τα επαναλαμβανόμενα δίκτυα συχνά χρησιμοποιούνται σε περιπτώσεις όπως η τεχνολογία μετατροπής από κείμενο σε φωνή (text-to-speech), οι μεταφράσεις, και η επεξεργασία κειμένου σε πραγματικό χρόνο, όπως οι γραμματικοί έλεγχοι και οι συμβουλές λέξεων.

### **Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)**

Τα συνελικτικά νευρωνικά δίκτυα είναι αρκετά διαφορετικά σε σύγκριση με δίκτυα άλλων ειδών. Χρησιμοποιούνται κυρίως για αναγνώριση εικόνων, αλλά μπορούν επίσης να χρησιμοποιηθούν για αναγνώριση ήχου. Όταν η είσοδος του δικτύου είναι μια εικόνα, τότε κάθε νευρώνας στο συνελικτικό τμήμα του δικτύου υπολογίζει την είσοδο σε μικρά τμήματα της συνολικής εισόδου.

Η επεξεργασία συνήθως περιλαμβάνει την μετατροπή της αρχικής εικόνας από μορφή RGB ή HSI σε κλίμακα του γκρι (gray-scale), μια κίνηση που βοηθάει στον εντοπισμό των άκρων μέσα στην εικόνα, και κατά συνέπεια στην καλύτερη κατηγοριοποίηση της εισόδου.



Εικόνα 9: Συνελικτικό νευρωνικό δίκτυο.

## 2.5 ΤΡΟΠΟΙ ΑΞΙΟΠΟΙΗΣΗΣ

Όπως αναφέρθηκε στο κεφάλαιο 2.4, υπάρχουν πολλών και διαφορετικών ειδών νευρωνικά δίκτυα. Κάθε ξεχωριστό δίκτυο, όχι μόνο περιέχει τα θετικά και τα αρνητικά του στοιχεία, αλλά μπορεί ταυτόχρονα να αξιοποιηθεί σε διαφορετικές καταστάσεις, ανάλογα με τις ανάγκες και τα προβλήματα που επιθυμούμε να επιλύσουμε.

Στον χώρο του μάρκετινγκ, τα νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για την βελτίωση των ήδη υπάρχοντων εργαλείων, παρέχοντας τη δυνατότητα της πρόβλεψης της συμπεριφοράς των καταναλωτών, την αυτοματοποίηση του χώρου, και την πρόγνωση πωλήσεων. Ιδιαίτερη χρήση αυτών των δικτύων παρατηρείται στον τομέα της αναλυτικής πρόβλεψης (predictive analytics), καθώς τα νευρωνικά δίκτυα μπορούν να βοηθήσουν στην πρόβλεψη των αποτελεσμάτων μιας επιχειρηματικής εκστρατείας εξετάζοντας τα αποτελέσματα προηγούμενων εκστρατειών. (Brenner, 2018)

Παρόμοια χρήση των νευρωνικών δικτύων παρατηρείται και στον οικονομικό και τραπεζικό τομέα. Δηλαδή χρησιμοποιούνται ιστορικά δεδομένα για την πρόβλεψη της αύξησης ή πτώσης της τιμής των χρηματιστηριακών μετοχών. Ιδιαίτερη χρήση αυτών των δικτύων πραγματοποιείται επίσης στην διαδικασία του ελέγχου των αιτήσεων



δανείων, όπου τα νευρωνικά δίκτυα βοηθούν σε έναν εξαιρετικά μεγάλο βαθμό στην λήψη της τελικής απόφασης όσον αφορά την έγκριση ή απόρριψη ενός δανείου. (Seetharaman, 2018)

Άλλοι εξίσου σημαντικοί τρόποι αξιοποίησης αποτελούν την χρήση των νευρωνικών δικτύων στην ιατρική για την ανίχνευση κρίσιμων νευρολογικών παθήσεων, όπως τα εγκεφαλικά, ή ακόμα και τον εντοπισμό καρκινικών κυττάρων (Bresnick, 2018), στην αναγνώριση χαρακτήρων, εικόνων, φωνής, όπως ακόμα και σε συστήματα αυτόματης οδήγησης.

### **3. ΟΠΤΙΚΗ ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ**

Από τις πρώτες μηχανές αναγνώρισης μέχρι και σήμερα έχουν υπάρξει πολλές και διάφορες τεχνολογίες, τεχνικές και αξιοποιήσεις της OCR, κάθε μια με τα δικά της πλεονεκτήματα και μειονεκτήματα, αλλά κάθε μια από αυτές βοήθησε να στρωθεί ένα μονοπάτι για να φτάσει η συγκεκριμένη τεχνολογία στο σημείο όπου βρίσκεται σήμερα. Ως τρίτη ενότητα της παρούσας πτυχιακής εργασίας, πραγματοποιείται μια απόπειρα εξήγησης αυτής της ιδιαίτερα σημαντικής ιστορίας της οπτικής αναγνώρισης με ιδιαίτερη αναφορά στα πλεονεκτήματα και στον τρόπο με τον οποίο αυτή η τεχνολογία αξιοποιεί όλες αυτές τις διάφορες υποκατηγορίες OCR μαζί με ορισμένες εξίσου σημαντικές τεχνικές βελτιστοποίησης.

#### **3.1 ΙΣΤΟΡΙΑ**

Η ιστορία της οπτικής αναγνώρισης χαρακτήρων (OCR) πηγάζει από την περίοδο της τηλεγραφίας, και συγκεκριμένα στην ακμή του πρώτου παγκοσμίου πολέμου. Από το 1850 μέχρι και εκείνη την χρονική περίοδο, υπήρχε ένα ιδιαίτερα μεγάλο ενδιαφέρον για την συγκεκριμένη τεχνολογία, μιας και πολύ εφευρέτες έστελναν σε γραφεία πατεντών σχέδια με εφευρέσεις τους που μπορούσαν να διαβάσουν κείμενα. Παρόλα αυτά, η τεχνολογία της οπτικής αναγνώρισης χαρακτήρων άνθησε ταυτόχρονα με την τεχνολογία των υπολογιστικών συστημάτων, και απογειώθηκε μέσα από την χρήση της στην επεξεργασία δεδομένων.

Το 1914 περίπου, ο φυσικός Emanuel Goldberg κατασκεύασε μια μηχανή, η οποία μπορούσε να διαβάσει χαρακτήρες, και στη συνέχεια τους μετέτρεπε σε κώδικα τηλεγράφου, γνωστός και ως κώδικας μορς, και μετέπειτα μετέδιδε τηλεγραφικά μηνύματα μέσω καλωδίων χωρίς να υπάρχει ανθρώπινη παρέμβαση. Εκείνη την περίοδο, τόσο οι εταιρίες, όσο και οι επιχειρήσεις συνήθιζαν να αποθηκεύουν τα οικονομικά τους στοιχεία σε φιλμ, μια τεχνική που είχε ένα σημαντικό μειονέκτημα στην ανάκτηση αυτών των δεδομένων, καθώς θα έπρεπε πρώτα να γίνει μια χρονοβόρα διαδικασία επάνω στο φιλμ. Για να αντιμετωπίσει αυτό το πρόβλημα, ο Goldberg χρησιμοποίησε φωτοκύτταρα, δηλαδή συσκευές που μετατρέπουν το φως σε ενέργεια, για να πραγματοποιήσει αναγνώριση προτύπων με την βοήθεια ενός προβολέα ταινιών. Βρίσκοντας νέες χρήσεις για ήδη υπάρχοντες τεχνολογίες, ο Goldberg έκανε τα πρώτα βήματα προς την αυτοματοποίηση επιχειρησιακών εγγράφων. (Britton, 2019)

Ταυτόχρονα, ο Dr. Edmund Fournier d'Albe του Πανεπιστημίου του Birmingham δημιούργησε και εκείνος μια συσκευή αναγνώρισης χαρακτήρων, η οποία παρήγαγε ήχους. Η συσκευή αυτή ονομάστηκε Optophone, ένα σκάνερ χειρός, που όταν κινούταν επάνω από μία τυπωμένη σελίδα παρήγαγε συγκεκριμένους ηχητικούς τόνους, οι οποίοι συνδέονταν με ένα συγκεκριμένο γράμμα ή σύμβολο. Κατά συνέπεια, άνθρωποι με προβλήματα όρασης μπορούσαν να κατανοήσουν το περιεχόμενο ενός εγγράφου, μαθαίνοντας πρώτα τους χαρακτηριστικούς ήχους των επί μέρους γραμμάτων.

Το 1951 ο Αμερικανός εφευρέτης David Shepard, μαζί με τον συνάδελφο του Harvey Cook Jr., δημιούργησαν το πρώτο σύστημα οπτικής αναγνώρισης χαρακτήρων, εμπονομαζόμενο ως Gismo, το οποίο αργότερα μετονομάστηκε σε Analyzing Reader. Το Gismo ήταν μια μηχανή, σχεδιασμένη να μετατρέπει τυπωμένα μηνύματα σε γλώσσα μηχανής για μετέπειτα επεξεργασία από υπολογιστικά συστήματα. Μπορούσε να αναγνωρίσει 23 γράμματα του λατινικού αλφάβητου, όπως τυπώνονταν επάνω στο χαρτί από μια απλή γραφομηχανή. Έπειτα από έναν χρόνο αναβαθμίσεων, η μηχανή μπορούσε πλέον να αναγνωρίζει όλους τους 26 λατινικούς χαρακτήρες, και την ίδια περίοδο ο Shepard έκανε αίτηση πατέντας, η οποία και εγκρίθηκε το 1953 με τον αριθμό US 2663758. (Shepard, 1953) Αργότερα, ο Shepard δημιούργησε την δική του επιχείρηση με όνομα Intelligent Machines Research Co. (IMR), όπου και προχώρησε στην εμπορευματοποίηση του Gismo σε εταιρίες όπως η AT&T, First National City

Bank, Reader's Digest, καθώς και σε μεγάλες τράπεζες, και πετρελαϊκές επιχειρήσεις. (History Computer, n.d)

Μια παρόμοια συσκευή με το Optophone δημιουργήθηκε το 1962, όταν ο John Linvill, καθηγητής του Πανεπιστήμιο Stanford, κατασκεύασε μια συσκευή με το όνομα Optacon για την κόρη του που είχε χάσει την όραση της από την ηλικία των τριών. Το Optacon αποτελούνταν από την κύρια συσκευή που περιείχε μια ειδική υποδοχή, στην οποία μπορούσε ο χρήστης να εισάγει τα δάχτυλα του ενός χεριού του. Η συσκευή συνδεόταν με ένα σκάνερ χειρός, με το οποίο μπορούσε ο χρήστης να κινηθεί επάνω από ένα έντυπο. Το σκάνερ αναγνώριζε τους χαρακτήρες από το έντυπο και στη συνέχεια έστελνε παλμούς στις άκρες των δακτύλων του χρήστη, δημιουργώντας μια «εικόνα» επάνω στο δάκτυλο, με την οποία ο χρήστης μπορούσε να κατανοήσει το περιεχόμενο ενός κειμένου. (How We Read, n.d)

Ορισμένες άλλες σημαντικές εφευρέσεις και αξιοποιήσεις της οπτικής αναγνώρισης χαρακτήρων μπορούν να χαρακτηριστούν οι ακόλουθες:

- Το 1974 ο Αμερικανός εφευρέτης Ray Kurzweil δημιουργεί το πρώτο λογισμικό που μπορεί να αναγνωρίσει τυπωμένο κείμενο γραμμένο σε οποιαδήποτε γραμματοσειρά.
- Το 1984 η Caere Corporation κατασκευάζει το πρώτο σκάνερ διαβατηρίων για το Αμερικάνικο State Department (Υπουργείο Εξωτερικών).
- Το 1992 η Ρωσική εταιρεία Okrus δημιουργεί το πρώτο πρόγραμμα αναγνώρισης του κυριλλικού αλφάβητου.
- Το 2011 δημιουργείται το Google Ngram Viewer για την χαρτογράφηση της συχνότητας των λέξεων από οποιαδήποτε τυπωμένη πηγή μεταξύ του 1950 και του 2008.
- Το 2013 δημιουργείται το dataset MNIST για την εκπαίδευση αλγορίθμων μηχανικής μάθησης στην αναγνώριση προτύπων.

Στις μέρες μας, υπάρχει μια ιδιαίτερα ευρεία πρόσβαση σε αλγορίθμους και συστήματα οπτικής αναγνώρισης χαρακτήρων με την μορφή APIs, ή ακόμα και στο διαδίκτυο, τα οποία είναι σε θέση να αναγνωρίσουν τους περισσότερους χαρακτήρες και τις περισσότερες γραμματοσειρές με αρκετά μεγάλο επίπεδο ακρίβειας. Αν και η συγκεκριμένη τεχνολογία εξελίσσεται συνεχώς, το περιθώριο λαθών συνεχίζει να

υπάρχει, ένα γεγονός που κρίνει τόσο τον ανθρώπινο έλεγχο, όσο και την ανθρώπινη παρέμβαση άκρως απαραίτητη για την ομαλή λειτουργία τέτοιων συστημάτων.

### **3.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ**

Οι επιχειρήσεις και οι οργανισμοί μπορούν να χαρακτηριστούν ως οι καλύτεροι πελάτες όσον αφορά τα συστήματα οπτικής αναγνώρισης χαρακτήρων. Με την εξέλιξη της τεχνολογίας των ηλεκτρονικών υπολογιστών, οι εταιρίες ξεκίνησαν να χρησιμοποιούν αυτά τα νέα και προηγμένα συστήματα για να πραγματοποιούν τις καθημερινές τους δραστηριότητες. Σταδιακά, οι επιχειρήσεις ανακάλυψαν τις δυνατότητες που τους παρέχει η ψηφιοποίηση εγγράφων, και ως αποτέλεσμα, ένα σημαντικό ποσοστό των συνολικών δραστηριοτήτων τους ξεκίνησε να πραγματοποιείται με τη χρήση ψηφιοποιημένων εντύπων. Παρόλα αυτά, η χρήση των φυσικών εγγράφων παρέμεινε, έως ότου οι ίδιες οι επιχειρήσεις ανακάλυψαν τις δυνατότητες που τους προσέφερε η τεχνολογία της οπτικής αναγνώρισης χαρακτήρων. Ορισμένες από αυτές τις δυνατότητες μπορούν να χαρακτηριστούν οι ακόλουθες:

#### **Μείωση εξόδων λειτουργίας**

Η χρήση της οπτικής αναγνώρισης χαρακτήρων για εύκολη και άμεση ψηφιοποίηση των εγγράφων επέτρεψε στις επιχειρήσεις να μετατρέψουν ένα ιδιαίτερα σημαντικό μέρος των συνολικών επιχειρησιακών εντύπων τους σε ψηφιακή μορφή. Δίχως αυτήν την τεχνολογία, ο μόνος τρόπος που θα μπορούσε να πραγματοποιηθεί μια τέτοια ενέργεια θα ήταν μέσα από την μίσθωση ατόμων, ενώ ταυτόχρονα η ίδια η ενέργεια θα έπαιρνε ιδιαίτερα μεγάλο και πολύτιμο χρόνο για να ολοκληρωθεί. Για τις επιχειρήσεις, ο χρόνος είναι χρήμα, όσο χρονοβόρα είναι μια ενέργεια, τόσο λιγότερο κέρδος μπορεί να βγάλει μια εταιρία.

#### **Αύξηση αποδοτικότητας και παραγωγικότητας**

Για έναν εργαζόμενο, μια θεωρητικά απλή αναζήτηση ενός εντύπου μπορεί να διαρκέσει από μερικά λεπτά, έως και μερικές ώρες, ανάλογα με την ευκολία του εντοπισμού αυτού του εγγράφου, και την ευκολία πρόσβασης σε αυτό. Αυτός ο χρόνος που δαπανάται άσκοπα για ένα μόνο έγγραφο κοστίζει όχι μόνο στην επιχείρηση από

πλευράς παραγωγικότητας, αλλά και από τον ίδιο τον εργαζόμενο, καθώς του δημιουργεί ψυχολογική κούραση, και συχνά, ακόμα και εκνευρισμό.

Με την ψηφιοποίηση των επιχειρησιακών εγγράφων, τη σωστή αποθήκευση και την αρχειοθέτηση τους, ο ίδιος ο προαναφερόμενος εργαζόμενος μπορεί να αποκτήσει εύκολη, γρήγορη ή και άμεση πρόσβαση στο απαιτούμενο έντυπο μέσα σε λίγα μόνο δευτερόλεπτα, δίχως να υπάρχει η ανάγκη της φυσικής αναζήτησης και εντοπισμού του. Επί πλέον, με τη σωστή ψηφιοποίηση, συχνά δίνεται η δυνατότητα επεξεργασίας του ψηφιοποιημένου αρχείου, ανάλογα με την απαιτούμενη περίσταση. Αν δεν υπήρχε η ψηφιοποίηση, μια τέτοια ενέργεια ίσως απαιτούσε την εξ' ολοκλήρου δημιουργία ενός νέου φυσικού εγγράφου, αποσπώντας σημαντικό χρόνο από τον εργαζόμενο και σημαντικό κέρδος από την επιχείρηση.

### **Μεγαλύτερη ασφάλεια**

Η εύκολη δημιουργία αρχείων ανάκτησης (back up) αποτελεί ένα εξίσου σημαντικό πλεονέκτημα που παρέχεται από την οπτική αναγνώριση χαρακτήρων. Σε περίπτωση που υπάρξει κάποια καταστροφή στο χώρο εργασίας, ένα μεγάλο, ή ακόμα και ολικό, μέρος των φυσικών εγγράφων μπορεί να χαθούν οριστικά, δίχως τη δυνατότητα ανάκτησης μέρους τους. Ευαίσθητα και άκρως σημαντικά έγγραφα για την ορθή λειτουργία της επιχείρησης μπορούν να αποθηκευτούν σε ηλεκτρονικές βάσεις δεδομένων, και σε συστήματα cloud, μακριά δηλαδή από τον φυσικό χώρο εργασίας.

## **3.3 ΕΦΑΡΜΟΓΕΣ**

Είναι προφανές πως από την πρώτη στιγμή που εφευρέθηκαν οι πρώτες μηχανές οπτικής αναγνώρισης χαρακτήρων, εμφανίστηκε μια ιδιαίτερα σημαντική ανάγκη της αξιοποίησης μιας τέτοιας τεχνολογίας από άτομα με προβλήματα όρασης. Με την πάροδο του χρόνου και της εξέλιξης της τεχνολογίας των υπολογιστικών συστημάτων, η αναγνώριση χαρακτήρων βρέθηκε να αξιοποιείται από πολλούς διαφορετικούς τομείς, οι οποίοι φάνταζαν απίστευτοι από τους πρώτους εφευρέτες τέτοιων μηχανών. Ορισμένοι από αυτούς τους τομείς που εφαρμόζουν τα πλεονεκτήματα της οπτικής αναγνώρισης χαρακτήρων είναι οι ακόλουθοι: (Khurana, et al., 2018)

## **Τραπεζικός τομέας**

Τόσο ο τραπεζικός τομέας, καθώς και άλλα οικονομικά τμήματα, όπως ο χώρος των ασφαλίσεων, είναι ιδιαίτερα σημαντικοί καταναλωτές της οπτικής αναγνώρισης χαρακτήρων. Η συνηθέστερη χρήση αυτής της τεχνολογίας συναντάται στον έλεγχο επιταγών. Το περιεχόμενο μιας χειρόγραφης επιταγής μετατρέπεται σε ψηφιακή μορφή έπειτα από το πέρασμα της από ένα σκάνερ, η εγκυρότητα της υπογραφής επαληθεύεται έπειτα από ταυτοποίηση της μέσω μιας βάσης δεδομένων, και στην συνέχεια επικυρώνεται ή ακυρώνεται σε πραγματικό χρόνο.

## **Νομικός τομέας**

Ο κλάδος της νομικής θεωρείται από τους κλάδους που παράγουν τον μεγαλύτερο όγκο εγγράφων παγκοσμίως. Τεράστιες στοίβες από καταθέσεις ενόρκων, αποφάσεις, διαθήκες, και πολλά άλλα είδη έντυπων νομικών εγγράφων μπορούν να ψηφιοποιηθούν, αποθηκευτούν και αρχειοθετηθούν χρησιμοποιώντας συστήματα OCR. Κάτι που επιτρέπει την απλή και ταχύρρυθμη αναζήτηση και πρόσβαση σε εκατοντάδες χιλιάδες έγγραφα που πηγάζουν από δεκάδες ή ακόμα και εκατοντάδες χρόνια πίσω.

## **Ιατρική**

Σε παρόμοια κατάσταση με τον κλάδο της νομικής βρίσκεται και η ιατρική κοινότητα. Ο όγκος εγγράφων είναι ιδιαίτερα μεγάλος, και ταυτόχρονα γεμάτος με άκρως σημαντικά στοιχεία, όπως το ιατρικό ιστορικό ασθενών, αποτελέσματα διαγνώσεων, νοσοκομειακά στοιχεία, καθώς και πολλά άλλα. Η αποθήκευση όλων αυτών των εγγράφων σε μια ψηφιοποιημένη μορφή, όχι μόνο διευκολύνει την αναζήτηση και την πρόσβαση τέτοιων δεδομένων, αλλά ταυτόχρονα βελτιώνει τη λογιστική διαδικασία της λειτουργίας της μονάδας υγείας, όπως οι παραγγελίες φαρμακευτικών ειδών, εξοπλισμού, και άλλων προμηθειών.

## **Διατήρηση ιστορικών εγγράφων**

Παλαιές βιβλιοθήκες, ιστορικά πολιτιστικά κέντρα, ακόμα και μουσεία είναι σημεία όπου αποθηκεύονται χιλιάδες ιστορικά χειρόγραφα έντυπα και απομνημονεύματα. Ντοκουμέντα μεγάλης ηλικίας είναι ιδιαίτερα ευάλωτα σε εξωτερικές συνθήκες, και ταυτόχρονα χρειάζονται ιδιαίτερη προσοχή τόσο στην αποθήκευση, όσο και στην διατήρησή τους. Τεχνικές σάρωσης και ψηφιοποίησης χρησιμοποιούνται ευρέως για τη

μετατροπή αυτών των εύθραυστων εγγράφων σε μια ψηφιακή μορφή, αναλλοίωτη στον χρόνο.

### **Ταυτοποίηση**

Στις μέρες μας η ταυτοποίηση αποτελεί μια άκρως σημαντική διαδικασία που πρέπει να πραγματοποιηθεί σε καταστάσεις όπου απαιτούνται τα προσωπικά δεδομένα ενός ατόμου. Πολλές φορές όμως αυτή η διαδικασία μπορεί να αποβεί ιδιαίτερα χρονοβόρα, ειδικά όταν υπάρχει ένα μεγάλο πλήθος ατόμων που πρέπει να περάσουν από την συγκεκριμένη διαδικασία. Τεχνικές οπτικής αναγνώρισης χαρακτήρων μπορούν να αξιοποιηθούν επάνω σε μηχανήματα σκάνερ για τον έλεγχο ταυτοτήτων, διαβατηρίων, διπλωμάτων οδήγησης, καθώς και άλλων εγγράφων ταυτοποίησης που πρέπει να ελεγχθούν με έναν αρκετά γρήγορο ρυθμό από αστυνομικές διευθύνσεις, αεροδρόμια και λοιπούς οργανισμούς για την ορθότητα και εγκυρότητα των προσωπικών δεδομένων.

### **Άτομα με προβλήματα όρασης**

Ιστορικά, η πρώτη χρήση της οπτικής αναγνώρισης χαρακτήρων πραγματοποιήθηκε για τη βοήθεια στην ανάγνωση και κατανόηση κειμένων σε άτομα με προβλήματα όρασης. Ακόμα και σήμερα, αυτή η τεχνολογία μπορεί να χρησιμοποιηθεί μαζί με ένα σύστημα μετατροπής κειμένου σε ομιλία (text-to-speech) για τον εντοπισμό κειμένου μέσα από έντυπα βιβλία, έγγραφα, κλπ. Εναλλακτικά, η τεχνολογία της OCR θα μπορούσε να αποτελέσει ένα άκρως σημαντικό τμήμα σε ένα σύστημα από εκτυπωτές, σχεδιασμένοι να τυπώνουν βιβλία σε μορφή Braille.

## **3.4 ΚΑΤΗΓΟΡΙΕΣ**

Η τεχνολογία της οπτικής αναγνώρισης αποτελείται από διάφορες κατηγορίες, και κατ' επέκταση από διαφορετικές τεχνολογίες. Εκ πρώτης όψεως μπορεί αυτές οι κατηγορίες να φαίνονται αρκετά πανομοιότυπες, παρόλα αυτά κάθε μια από αυτές περιέχει τα δικά της μοναδικά χαρακτηριστικά, εφαρμόζεται σε συγκεκριμένες περιπτώσεις, και τέλος κατέχει τα δικά της πλεονεκτήματα και μειονεκτήματα. Στο δια ταύτα όμως θα μπορούσαμε να συμπεράνουμε πως όλες αυτές οι κατηγορίες εντάσσονται στην ίδια

τεχνολογική οικογένεια. Ορισμένες από τις σημαντικότερες κατηγορίες μπορούν να χαρακτηριστούν οι εξής.

### **Οπτική αναγνώριση χαρακτήρων (OCR)**

Η οπτική αναγνώριση χαρακτήρων (Optical Character Recognition) χαρακτηρίζεται ως η πιο διαδεδομένη μορφή της οπτικής αναγνώρισης. Είναι μια τεχνολογία που επιτρέπει τη μετατροπή εικόνων ή εγγράφων που περιέχουν κάποια μορφή (συνήθως τυπωμένου) κειμένου σε ηλεκτρονική μορφή, και την αποθήκευση των αποτελεσμάτων σε ένα υπολογιστικό σύστημα. Αυτή η μετατροπή πραγματοποιείται σχεδόν άμεσα, και μπορεί να παρέχει δυνατότητες όπως η εύκολη και γρήγορη επεξεργασία του αναγνωρισμένου κειμένου, η εύκολη αναζήτηση λέξεων εντός του περιεχομένου του κειμένου, καθώς επίσης και η άμεση εισαγωγή δεδομένων σε φόρμες ή ακόμα και βάσεις δεδομένων.

Το νευρωνικό δίκτυο εκπαιδεύεται επάνω σε ξεχωριστές εικόνες ενός μόνο χαρακτήρα ή συμβόλου κάθε φορά, ως αποτέλεσμα, η αναγνώριση μιας λέξης πραγματοποιείται με την αναγνώριση κάθε γράμματος ξεχωριστά. Αυτό αποτελεί και ένα από τα κυριότερα αρνητικά στοιχεία της παρούσας τεχνολογίας, δεδομένου ότι σε περίπτωση που υπάρξει αλλοίωση της εικόνας προς αναγνώριση, και δύο χαρακτήρες εμφανίζονται «πολύ κοντά» ο ένας με τον άλλον, το σύστημα ενδέχεται να αναγνωρίσει αυτούς τους δύο χαρακτήρες ως έναν, με αποτέλεσμα να εξαχθεί εσφαλμένο συμπέρασμα από το σύστημα.

Άρα, η συγκεκριμένη τεχνολογία δεν είναι 100% ακριβής, αφού μπορούν να υπάρξουν λανθασμένα αποτελέσματα λόγω μη ξεκάθαρων γραμματοσειρών, ή αλλοίωσης της εισόδου. Θεωρείται λοιπόν ιδιαίτερα σημαντικός ο έλεγχος των αποτελεσμάτων για τον εντοπισμό τυχόν σφαλμάτων κατά τη διαδικασία της αναγνώρισης, κάτι που μπορεί να αποτελέσει μια ιδιαίτερα χρονοβόρα διαδικασία, ανάλογα με το μέγεθος της εξόδου.

### **Οπτική αναγνώριση λέξεων (OWR)**

Η οπτική αναγνώριση λέξεων (Optical Word Recognition) αποτελεί μια δεύτερη τεχνική αναγνώρισης στην ίδια τεχνολογία της OCR. Στην συγκεκριμένη περίπτωση, η αναγνώριση πραγματοποιείται σε ολόκληρες λέξεις γλωσσών όπου υπάρχει ένα εμφανές κενό ανάμεσα στις λέξεις, έναντι της αναγνώρισης επάνω σε κάθε χαρακτήρα ξεχωριστά. Τις περισσότερες φορές, όταν γίνεται αναφορά σε συστήματα OCR,



θεωρείται πως η αναγνώριση πραγματοποιείται σε ολόκληρες λέξεις, παρόλα αυτά οι δύο κατηγορίες διαφέρουν ελαφρώς μεταξύ τους.

### **Οπτική αναγνώριση σημείων (OMR)**

Σε αντίθεση με τις τεχνολογίες OCR, η οπτική αναγνώριση σημείων (Optical Mark Recognition) βασίζεται κυρίως στον εντοπισμό συγκεκριμένων σημείων ή προτύπων σε ειδικά έντυπα. Η συγκεκριμένη τεχνολογία σχεδιάστηκε κατά κύριο λόγο για χρήση στη διόρθωση απαντήσεων σε τεστ εκπαιδευτικών οργανισμών, όπου και χρησιμοποιείται ακόμα και σήμερα. Επιπλέον η ίδια τεχνολογία μπορεί να συναντηθεί και σε πρακτορεία στοιχηματισμού για την επικύρωση στοιχηματικών δελτίων από ειδικά σχεδιασμένα μηχανήματα, για την άμεση εξαγωγή δεδομένων από φόρμες, ακόμα και για την καταμέτρηση ψήφων σε εκλογικές διαδικασίες.

Οι συγκεκριμένες συσκευές λειτουργούν σαν σαρωτές, εκπέμποντας δηλαδή φως επάνω στο εισερχόμενο έγγραφο, το οποίο βοηθάει στον εντοπισμό των συγκεκριμένων πεδίων που προορίζονται προς αναγνώριση. Τα ίδια τα έγγραφα που χρησιμοποιούνται είναι σχεδιασμένα να περιέχουν μικρούς κύκλους ή τετράγωνα που λειτουργούν ως πεδία, μερικά από τα οποία μπορεί να είναι μαρκαρισμένα ανάλογα με τις επιλογές του κάθε χρήστη. (Ask Any Difference, n.d)

### **Έξυπνη αναγνώριση χαρακτήρων (ICR)**

Η έξυπνη αναγνώριση χαρακτήρων (Intelligent Character Recognition) θα μπορούσε πολύ εύκολα να χαρακτηριστεί ως μια εξελιγμένη έκδοση της τεχνολογίας OCR. Συγκεκριμένα, η ICR είναι σχεδιασμένη να εντοπίζει, να αναγνωρίζει πολλούς διαφορετικούς τρόπους γραφικών χαρακτήρων, και στη συνέχεια να τους μετατρέπει σε ψηφιακούς χαρακτήρες. Αυτή είναι και η κυριότερη διαφορά μεταξύ των τεχνολογιών OCR και ICR. Για την αναγνώριση τυπωμένων εγγράφων λαμβάνουμε καλύτερα αποτελέσματα με την OCR, ενώ για την αναγνώριση σημειώσεων ή χειρόγραφων εντύπων, η τεχνολογία της ICR αποτελεί μονόδρομο.

Η ICR περιέχει ένα νευρωνικό σύστημα, το οποίο επιτρέπει στο λογισμικό να ενημερώνει αυτόματα μια εσωτερική βάση δεδομένων με νέους γραφικούς χαρακτήρες. Ως αποτέλεσμα, ένα πρόγραμμα ICR μπορεί να κατανοήσει οποιονδήποτε χειρόγραφο

χαρακτήρα, και κατά συνέπεια αυξάνεται η ακρίβεια και η αποτελεσματικότητα της αναγνώρισης ανάλογα με τον χρόνο και την εμπειρία που λαμβάνει. (Taylor, n.d)

### **Έξυπνη αναγνώριση λέξεων (IWR)**

Όπως και με την τεχνολογία της OWR, έτσι και η έξυπνη αναγνώριση λέξεων (Intelligent Word Recognition) αποτελεί μια τροποποιημένη έκδοση της ICR, η οποία είναι σχεδιασμένη για την αναγνώριση επάνω σε ολόκληρες λέξεις, έναντι της ICR που η αναγνώριση γίνεται σε κάθε χαρακτήρα ξεχωριστά.

## **3.5 ΤΕΧΝΙΚΕΣ**

Η τεχνολογία της οπτικής αναγνώρισης χαρακτήρων εξαρτάται σε μεγάλο βαθμό από την ποιότητα των δεδομένων εισόδου που εισάγονται σε ένα τέτοιου είδους μοντέλο. Αυτά τα δεδομένα, ή αλλιώς εικόνες εισόδου, μπορεί να βρίσκονται σε μια καλή, ή ακόμα και εξαιρετική ποιότητα, ανάλογα με την ανάλυση, το επίπεδο θορύβου και την λεπτομέρεια που περιέχει κάθε εικόνα, σύμφωνα με τα κριτήρια που έχει θέσει ο χρήστης του συστήματος, κάτι που διευκολύνει τη διαδικασία της αναγνώρισης και κατά συνέπεια παράγει μεγάλα ποσοστά ακρίβειας κατά την ολοκλήρωση της. Αντιθέτως όμως, τα δεδομένα μπορούν επίσης να έχουν μια αρκετά κακή ποιότητα, όπως θόρυβο, κακή ανάλυση της εικόνας, κακός φωτισμός, θόλωση της εικόνας, κλπ. Σε αυτή την περίπτωση, συχνά απαιτείται να πραγματοποιηθούν ορισμένες τεχνικές που εξυπηρετούν στην βελτίωσης της ακρίβειας αναγνώρισης του συστήματος. Ορισμένες κατηγορίες βελτιστοποίησης μπορούν να χαρακτηριστούν οι ακόλουθες: (Karandish, n.d), (Magesh, n.d)

### **Προ-Επεξεργασία (Pre-Processing)**

Το πρώτο βήμα που πραγματοποιείται προτού οι εικόνες εισόδου εισέλθουν εντός του συστήματος οπτικής αναγνώρισης είναι η προ-επεξεργασία αυτών των εικόνων. Για να μπορέσει το δίκτυο να εξάγει τα καλύτερα δυνατά αποτελέσματα, το περιεχόμενο των εικόνων που χρησιμοποιούνται πρέπει να είναι όσο το δυνατόν ξεκάθαρο. Η διαδικασία

αυτή πραγματοποιείται με διάφορους τρόπους, αναλόγως της κατάστασης κάθε εικόνας. Γενικότερα, ορισμένες κύριες μέθοδοι προ-επεξεργασίας θεωρούνται οι εξής:

- **Διόρθωση στρέβλωσης (De-skew):** Η διαδικασία της περιστροφής μιας εικόνας σε τέτοιο βαθμό όπου το περιεχόμενο της εικόνας είναι απόλυτα ευθυγραμμισμένο τόσο οριζόντια όσο και κατακόρυφα. Συχνά η περιστροφή πραγματοποιείται κατά λίγες μοίρες ακολουθώντας την φορά των δεικτών του ρολογιού ή και αντίστροφα. Αυτή η διόρθωση συνήθως συναντάται όταν ένα έγγραφο δεν έχει σαρωθεί σωστά.
- **Διαδική μετατροπή (Binarization):** Η μετατροπή μιας έγχρωμης εικόνας σε ασπρόμαυρη. Θεωρείται ως μια εύκολη και ιδιαίτερα ακριβής τεχνική, καθώς μπορεί να ξεκαθαρίσει τα σημεία στα οποία υπάρχει κείμενο, ή ακόμα και άλλα απαιτούμενα στοιχεία εντός της εικόνας.
- **Ανάλυση διάταξης (Layout analysis):** Αποτελεί τον εντοπισμό διαφόρων τμημάτων σε μια εικόνα ή ένα έγγραφο. Συγκεκριμένα μπορεί να πραγματοποιηθεί ο διαχωρισμός πινάκων, παραγράφων, κ.λπ. σε ξεχωριστά κομμάτια για την αναγνώριση ή ακόμα και την αποφυγή αναγνώρισης επάνω σε αυτά τα συγκεκριμένα τμήματα.
- **Αναγνώριση γραφής (Script recognition):** Σε έγγραφα που περιέχουν γλώσσες πολλών διαφορετικών αλφάβητων και μπορούν να εναλλάσσονται τακτικά εντός του κειμένου, θεωρείται απαραίτητος ο εντοπισμός των σημείων όπου η γλώσσα αλλάζει, έτσι ώστε να μπορεί να αξιοποιηθεί το καλύτερο δυνατό σύστημα για κάθε σημείο του κειμένου.
- **Απομόνωση χαρακτήρων (Character isolation / Segmentation):** Πολλές φορές η κακή ανάλυση μιας εικόνας μπορεί να ενώσει δύο ξεχωριστούς χαρακτήρες σε σημείο όπου εμφανίζονται σαν ένας. Εναλλακτικά υπάρχει και η περίπτωση όπου ένας χαρακτήρας έχει χωριστεί σε δύο ή περισσότερα τμήματα. Η τεχνική της απομόνωσης χρησιμοποιείται σε τέτοιες περιπτώσεις για να διαχωρίσει ή ακόμα και για να ενώσει αυτούς τους χαρακτήρες, και κατά συνέπεια να μειώσει πιθανά λάθη που μπορεί να συμβούν κατά την διάρκεια της αναγνώρισης.

## **Αναγνώριση κειμένου (Text recognition)**

Η διαδικασία της αναγνώρισης κειμένου αποτελεί τον κορμό ολόκληρης της τεχνολογίας OCR, καθώς είναι υπεύθυνο για ολόκληρο το κομμάτι της αναγνώρισης χαρακτήρων ή λέξεων σε μια εικόνα ή ένα έγγραφο. Για να επιτευχθεί η αναγνώριση μπορούν να χρησιμοποιηθούν δύο κύριες τεχνικές, οι οποίες είναι οι ακόλουθες:

- **Συσχέτιση εικόνων (Image correlation):** Η ταύτιση εικόνων θεωρείται η πιο απλή μορφή αναγνώρισης, καθώς το σύστημα οπτικής αναγνώρισης προσπαθεί να αντιστοιχήσει έναν χαρακτήρα από μια εικόνα με έναν από τους χαρακτήρες που γνωρίζει από την διαδικασία της εκπαίδευσης του. Η ταύτιση επιτυγχάνεται απομονώνοντας κάθε χαρακτήρα ξεχωριστά, και στη συνέχεια αναλύοντας κάθε pixel αυτού του χαρακτήρα, με σκοπό την ταυτοποίηση του με κάποιον χαρακτήρα που υπάρχει μέσα στο εκπαιδευμένο μοντέλο. Η συγκεκριμένη τεχνική λειτουργεί καλύτερα σε τυπωμένους χαρακτήρες, αλλά υστερεί σημαντικά όταν καλείται να αναγνωρίσει χαρακτήρες από γραμματοσειρές τις οποίες το σύστημα δεν γνωρίζει.
- **Εξαγωγή χαρακτηριστικών (Feature extraction):** Ο δεύτερος τρόπος αναγνώρισης είναι η εξαγωγή χαρακτηριστικών. Το μοντέλο προσπαθεί να εντοπίσει και να εξάγει πρότυπα που υπάρχουν σε κάθε χαρακτήρα κατά την διαδικασία της εκπαίδευσης, όπως για παράδειγμα γραμμές, κύκλους που δημιουργούνται μέσα σε χαρακτήρες, γραμμές οι οποίες ενώνονται με άλλες, κλπ. Αποτελεί μια ιδιαίτερα αποτελεσματική τεχνική τόσο από πλευράς ακρίβειας, όσο και υπολογιστικής απόδοσης, καθώς δεν περιορίζεται σε συγκεκριμένες γραμματοσειρές όπως στην τεχνική της συσχέτισης εικόνων. Συχνά συναντάται σε συστήματα ICR, καθώς επίσης και στα πιο προηγμένα συστήματα OCR.

## **Μετα-Επεξεργασία (Post-Processing)**

Κατά την ολοκλήρωση της αναγνώρισης, η έξοδος του συστήματος μπορεί επίσης να περάσει από μια διαδικασία μετα-επεξεργασίας για περαιτέρω ενίσχυση της ακρίβειας αναγνώρισης. Μια ιδιαίτερα χρήσιμη τεχνική τέτοιου είδους περιλαμβάνει τη χρήση λεξικών, δηλαδή μια τεράστια λίστα λέξεων που θα μπορούσαν να εντοπιστούν εντός της εικόνας προς αναγνώριση. Το αποτέλεσμα που εξάγεται από το στάδιο της

αναγνώρισης ελέγχεται με το λεξικό για τον εντοπισμό πιθανών λέξεων. Η παρούσα τεχνική μπορεί να εντοπίσει πιθανές λάθος αναγνώρισεις που έχουν πραγματοποιηθεί, και στη συνέχεια να διορθώσει αυτά τα λάθη σύμφωνα με τις ήδη υπάρχουσες λέξεις εντός του λεξικού.

## **4. ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ**

Στην τελευταία ενότητα περιλαμβάνεται η διαδικασία με την οποία υλοποιήθηκε η εφαρμογή που συνοδεύει την τρέχουσα πτυχιακή εργασία, από τον τρόπο με τον οποίο επινοήθηκε η ιδέα ολόκληρης της εργασίας, στην γενική περιγραφή του κώδικα της εφαρμογής, του τρόπου λειτουργίας της, και τέλος στα τελικά αποτελέσματα που μπορεί να λάβει ο χρήστης. Τέλος, πραγματοποιείται μια αναφορά σε πιθανές βελτιώσεις που μπορεί να υλοποιηθούν στην εφαρμογή με σκοπό τον εμπλουτισμό των λειτουργιών της, καθώς επίσης και για την καλύτερη ακρίβεια και απόδοση της.

Ο πηγαίος κώδικας της εφαρμογής βρίσκεται στο ακόλουθο σύνδεσμο:  
<https://github.com/ElSarik/Thesis>

### **4.1 Η ΣΥΛΛΗΨΗ ΤΗΣ ΙΔΕΑΣ**

Ένα διάστημα πριν την εκδήλωση ενδιαφέροντος για την υλοποίηση της παρούσας πτυχιακής εργασίας, βρέθηκα σε μια κατάσταση όπου χρειαζόταν να δακτυλογραφώ πληροφορίες σε λογιστικά φύλλα τύπου Excel, τις οποίες πρώτα εξήγαγα από σαρωμένα έντυπα μορφής pdf.

Εκείνη την περίοδο, καθώς υλοποιούσα την προαναφερόμενη διαδικασία, ξεκίνησα να σκέφτομαι διάφορους τρόπους με τους οποίους θα μπορούσε αυτή η πράξη να αυτοματοποιηθεί, μιας και ήταν κάπως χρονοβόρα για την ολοκλήρωσή της. Με αυτό το σκεπτικό, μετά από μια σύντομη έρευνα ανακάλυψα την τεχνολογία της οπτικής αναγνώρισης χαρακτήρων, και αμέσως σκέφτηκα πως θα μπορούσα να πραγματοποιήσω μια περαιτέρω έρευνα επάνω στον συγκεκριμένο τομέα, ικανή να μπορέσει να αποτελέσει την πτυχιακή μου εργασία.

Το παρόν κείμενο, καθώς επίσης και η υλοποίηση της εφαρμογής που το συνοδεύει, αποτελεί το αποτέλεσμα αυτής της έρευνας διαρκείας, περίπου, έξι μηνών.

Ως γλώσσα προγραμματισμού για την κατασκευή της συγκεκριμένης εφαρμογής, επέλεξα να χρησιμοποιήσω την Python, και συγκεκριμένα την έκδοση 3.8 για να μην υπάρχουν τυχόν προβλήματα συμβατότητας με βιβλιοθήκες που ενδεχομένως να χρειάζονται κατά τη διάρκεια της υλοποίησης.

## **4.2 DATASETS**

Από τα πρώτα στάδια της προσωπικής μου μελέτης επάνω στην οπτική αναγνώριση χαρακτήρων, παρατήρησα η ιδιαίτερη σημασία που έχουν τα σετ δεδομένων (datasets) για την σωστή εκπαίδευση και πρόβλεψη των νευρωνικών μοντέλων.

Η βάση δεδομένων MNIST, του Εθνικού Ινστιτούτου Τεχνολογίας και Προτύπων (NIST), των Ηνωμένων Πολιτειών, αποτέλεσε την πρώτη επαφή με βάσεις δεδομένων που σχετίζονται σε μεγάλο βαθμό με την οπτική αναγνώριση. Η συγκεκριμένη βάση περιέχει περίπου 70.000 ασπρόμαυρες, χειρόγραφες εικόνες αριθμητικών χαρακτήρων, μεγέθους 28x28 pixels έκαστος, εκ των οποίων οι 60.000 προορίζονται προς εκπαίδευση μοντέλων και οι υπόλοιπες 10.000 προς επαλήθευση των αποτελεσμάτων του μοντέλου.

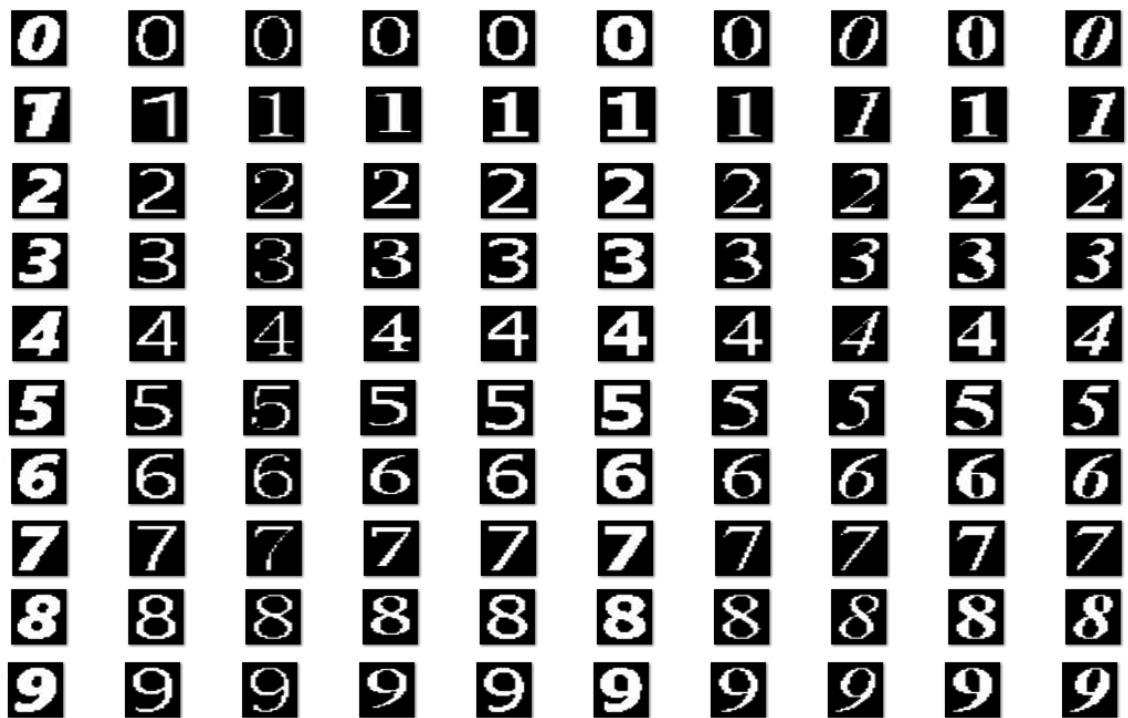


Εικόνα 10: Ένα μικρό δείγμα από το MNIST dataset.

Παίρνοντας ένα μικρό μέρος του MNIST, εξετάστηκε ο τρόπος με τον οποίο το συγκεκριμένο dataset αλληλοεπιδρά με νευρωνικά μοντέλα στο σκέλος της εκπαίδευσης και της πρόβλεψης. Τα αποτελέσματα ήταν ιδιαίτερα ενθαρρυντικά, καθώς το εκπαιδευμένο μοντέλο ήταν σε θέση να αναγνωρίσει παρόμοιους χαρακτήρες σε έναν αρκετά ικανοποιητικό βαθμό.

Μεγάλο μέρος της ακρίβειας των μοντέλων που έχουν εκπαιδευτεί με το MNIST έρχεται από το γεγονός πως υπάρχει μια πληθώρα διαφορετικών, και μοναδικών εικόνων για κάθε ξεχωριστό χαρακτήρα. Αυτό σημαίνει πως κατά την εκπαίδευση, το νευρωνικό δίκτυο προσπαθεί να εξάγει ορισμένα κοινά χαρακτηριστικά μεταξύ των διαφορετικών εικόνων που ανήκουν στο ίδιο ψηφίο, όπως για παράδειγμα συγκεκριμένες ακμές, κυκλικά σημεία, κλπ. Έτσι λοιπόν, για να μπορέσει ένα dataset να χαρακτηριστεί ως πετυχημένο, χρειάζεται να περιέχει πολλές διαφοροποιήσεις ενός κοινού χαρακτήρα ή συμβόλου.

Η ιδέα για την υλοποίηση του προγράμματος ήταν η δημιουργία μιας αυτόματης γεννήτριας παρόμοιων εικόνων με το MNIST, η οποία θα ήταν σε θέση να κατασκευάσει ασπρόμαυρες εικόνες, κάθε μια με έναν ξεχωριστό χαρακτήρα σε διαφορετική γραμματοσειρά.



Εικόνα 11: Ένα μικρό δείγμα από την γεννήτρια εικόνων.

Η εικόνα 11 παρουσιάζει το αποτέλεσμα αυτής της αυτόματης γεννήτριας εικόνων. Κάθε εικόνα αποτελείται από διαστάσεις 50x50 pixel, τριών χρωματικών καναλιών (rgb), μαύρου φόντο με έναν μόνο λευκό αριθμητικό χαρακτήρα διαφορετικών γραμματοσειρών σε κάθε εικόνα. Η επιλογή τριών καναλιών πάρθηκε για την ευκολότερη επεξεργασία των παραγόμενων εικόνων στη συνάρτηση `Augment_Image` (Πίνακας 4). Μιας και πολλές γραμματοσειρές διαφέρουν μεταξύ τους ως προς τον σχεδιασμό των χαρακτήρων, όπως το πάχος και τη καμπυλότητα, μπορούν εύκολα να παρέχουν στο δίκτυο μια διαφορετική «έκδοση» του ίδιου χαρακτήρα, με σκοπό την καλύτερη εκπαίδευση και απόδοση του.

#### 4.2.1 Γεννήτρια παραγωγής εικόνων

Κατά την εκτέλεση της εφαρμογής, η συνάρτηση `generate_store_dataset` καλείται όταν ο χρήστης επιθυμεί να ξεκινήσει την διαδικασία δημιουργίας ενός νέου μοντέλου. Η συγκεκριμένη συνάρτηση αποτελεί και την κύρια δομή της γεννήτριας εικόνων. Στην προκειμένη περίπτωση, ο χρήστης καλείται να επιλέξει ανάμεσα από ορισμένες επιλογές που του δίνονται από το γραφικό περιβάλλον της εφαρμογής, με



σκοπό να δημιουργηθεί το κατάλληλο dataset για τις ανάγκες του χρήστη. Περισσότερα στο κεφάλαιο 4.5 – Γραφική Διεπαφή.

```
font_name_array = ['arial', 'times', 'timesi', 'timesbd',  
'timesbi', 'bahnschrift', 'cambria', 'constan', 'lucon',  
'calibri', 'impact', 'segoepr', 'segoesc', 'comic',  
'pala', 'verdana', 'verdanai', 'verdanab', 'verdanaz',  
'trebuc', 'trebucit', 'trebucbd', 'trebucbi']  
  
font_size = 35 # Font size of each character
```

Πίνακας 1: Ορισμός γραμματοσειρών.

Αρχικά ορίζονται στην λίστα `font_name_array` μια σειρά από ονόματα διαφορετικών γραμματοσειρών, τα οποία χρησιμοποιούνται στην συνέχεια από την γεννήτρια για την εμφάνιση των χαρακτήρων σε κάθε δοθούμενη γραμματοσειρά. Επιπλέον εντάσσεται στην μεταβλητή `font_size` το μέγεθος με το οποίο θα εμφανιστούν οι χαρακτήρες επάνω στην εικόνα.

```
if(selected_dataset == 'Latin'):  
    for l in range (65, 91):  
        letters.append(chr(l))  
        directory_letters.append(chr(l))  
        letters_in_image.append(chr(l))
```

Πίνακας 2: Συλλογή χαρακτήρων.

Η εφαρμογή λαμβάνει την επιλογή του χρήστη για την δημιουργία ενός dataset, και στην συνέχεια αντιστοιχεί αυτήν την επιλογή με τους απαραίτητους χαρακτήρες. Συγκεκριμένα, αν για παράδειγμα ο χρήστης έχει επιλέξει την επιλογή για λατινικούς χαρακτήρες, θα χρησιμοποιηθούν μόνο οι χαρακτήρες που ανήκουν στο λατινικό αλφάβητο, με κωδικούς αριθμούς από το 65 έως το 90 σε Unicode. Αυτοί οι κωδικοί εισάγονται σε τρεις ξεχωριστές λίστες `letters`, `directory_letters`, `letters_in_image` ως χαρακτήρες και στην συνέχεια χρησιμοποιούνται για την δημιουργία των εικόνων.

```
#Creating the database folder.  
os.mkdir(dataset_directory_root)  
  
for l in range (0, len(letters)):
```

```

counter = 1

# Setting up the directory path for each character
letter_directory = dataset_directory_root + '/'
                  + directory_letters[l]

# Creating the directories
os.mkdir(letter_directory)

for font_name in font_name_array:
    try:
        font = ImageFont.truetype(font_name, font_size)

        # Setting up the picture name
        picture_name = letters[l] + '_' + str(counter)
                      + '.jpg'

        # Create a new RGB image for every character
        # with size x,y (PIL)
        image = Image.new('RGB', (img_width, img_height))

        # Enable drawing on the image
        draw = ImageDraw.Draw(image)

        # Draw the character on the image
        draw.text((10, int(img_height/(font_size/2))),
                  letters_in_image[l],
                  fill = (255, 255, 255), font = font)

        image = Augment_Image(image)

        # Moving the directory to the character folder
        os.chdir(letter_directory)

        # Using PIL to save the image
        image.save(picture_name)

        counter = counter + 1
    except:
        pass

TrainModel(dataset_directory_root, exec_directory_root,
            img_width, img_height)

```

Πίνακας 3: Γεννήτρια εικόνων.

Αμέσως μετά την εισαγωγή των επιλεγμένων χαρακτήρων εντός των λιστών, δημιουργείται ένας φάκελος με όνομα Dataset στο ίδιο σημείο από το οποίο εκτελείται η εφαρμογή. Ο συγκεκριμένος φάκελος προορίζεται ως μέσο αποθήκευσης των εικόνων

που δημιουργούνται από την γεννήτρια, και κατά συνέπεια ως ένας τρόπος ελέγχου της ορθότητας των εικόνων προτού εισαχθούν στο μοντέλο εκπαίδευσης. Εάν υπάρχει ήδη ο φάκελος Dataset, από κάποια προηγούμενη εκτέλεση του προγράμματος, τότε θα το ήδη υπάρχον περιεχόμενο θα διαγραφτεί, και στη συνέχεια θα αντικατασταθεί με τις νέες εικόνες. Εντός του φακέλου Dataset δημιουργούνται υποφάκελοι με τα ονόματα των χαρακτήρων προς δημιουργία, όπου και αποθηκεύονται οι εικόνες που παράγονται από την γεννήτρια.

Χρησιμοποιώντας τις συναρτήσεις `ImageFont`, `Image` και `ImageDraw` από την βιβλιοθήκη Python Image Library (PIL ή αλλιώς Pillow), ορίζεται πως στην μεταβλητή `font` θα εισαχθεί η τρέχουσα επιλεγμένη γραμματοσειρά από την λίστα `font_name_array`, μαζί την τιμή της `font_size` για το μέγεθος των χαρακτήρων εντός της εικόνας.

Στη συνέχεια προετοιμάζεται το όνομα που θα λάβει η εικόνα κατά την ολοκλήρωση της δημιουργίας της όταν αποθηκευτεί στο δίσκο, το οποίο λαμβάνει την μορφή: `χαρακτήρας_αριθμός`, ο οποίος αριθμός αποτελείται από έναν αυξανόμενο μετρητή `counter` ως τρόπο διαφοροποίησης μεταξύ όλων των εικόνων. Έπειτα δημιουργείται μια εικόνα `image` στην μνήμη του υπολογιστικού συστήματος με χρωματική μορφή RGB και μέγεθος τις τιμές των μεταβλητών `img_width` και `img_height`. Όταν έχει δημιουργηθεί η εικόνα, καλείται η `ImageDraw` επάνω στην εικόνα για να επιτρέψει την μετέπειτα επεξεργασία της. Η `draw` πραγματοποιεί την αρχική τοποθέτηση κάθε χαρακτήρα επάνω στην εικόνα, με λευκό χρώμα και την τρέχουσα γραμματοσειρά.

Τέλος, όταν έχει ολοκληρωθεί η διαδικασία της παραγωγής εικόνων, συμπεριλαμβανομένης και της ενίσχυσης των εικόνων μέσω της συνάρτησης `Augment_Image` (κεφάλαιο 4.2.2), καλείται η συνάρτηση `TrainModel` λαμβάνοντας ως ορίσματα την τοποθεσία των παραγόμενων εικόνων, του αρχείου εκτέλεσης της εφαρμογής και το μέγεθος των εικόνων (ύψος και πλάτος). Η συγκεκριμένη συνάρτηση είναι υπεύθυνη για την διαδικασία εκκίνησης της εκπαίδευσης.

#### 4.2.2 Ενίσχυση εικόνων

Αμέσως μετά η εικόνα περνάει από την συνάρτηση `Augment_Image`, η οποία δημιουργεί μια περιοχή γύρω από τον χαρακτήρα της εικόνας. Αυτή η περιοχή τοποθετεί τον χαρακτήρα της εικόνας ακριβώς στο κέντρο της και στη συνέχεια αντικαθιστά την αρχική εικόνα. Η νέα, πλέον, εικόνα επιστρέφει πίσω στην συνάρτηση `generate_store_dataset`, όπου και στη συνέχεια θα αποθηκευτεί εντός του κατάλληλου φακέλου, ενώ κατά την ολοκλήρωση της συνάρτησης θα ξεκινήσει η διαδικασία της εκπαίδευσης.

```
def Augment_Image(image):

    global img_height
    global img_width

    #Converting the image from PIL to CV2 format.
    cv2Image = np.array(image)
    cv2Image = cv2.cvtColor(cv2Image, cv2.COLOR_RGB2BGR)

    #Grayscaleing the image and getting its threshold.
    gray = cv2.cvtColor(cv2Image, cv2.COLOR_BGR2GRAY)
    thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY
                           + cv2.THRESH_OTSU)[1]

    #Morphologically closing the image to connect
    #all the character parts.
    kernel = np.ones((13,13),np.uint8)
    closing = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE,
                               kernel)

    #Detecting contours inside the images.
    cnts, hierarchy = cv2.findContours
                        (closing, cv2.RETR_EXTERNAL,
                        cv2.CHAIN_APPROX_SIMPLE)

    for c in cnts:

        #Bounding the contours into a rectangle,
        #then expanding the rectangle.
        (x, y, w, h) = cv2.boundingRect(c)
        x = x-3
        y = y-3
        w = w+6
        h = h+6

        if ((x-3) < 0):
            x = 0
        if ((y-3) < 0):
            y = 0
```

```

#Creating a new image based on the expanded rectangle
roi = thresh[y:y+h, x:x+w]

#Resizing the new image.
roi = cv2.resize(roi, (img_width, img_height))

cv2Image = cv2.cvtColor(roi, cv2.COLOR_BGR2RGB)
pilImage = Image.fromarray(roi)

return pilImage

```

Πίνακας 4: Συνάρτηση Augment\_Image.

Συγκεκριμένα, η Augment\_Image μετατρέπει την εικόνα από PIL σε μορφή συμβατή με την βιβλιοθήκη OpenCV για την ευκολότερη επεξεργασία τους. Αρχικά η εικόνα λαμβάνει μια ασπρόμαυρη μορφή στα χρωματικά της κανάλια μέσω της cv2.cvtColor, κάτι που στην συνέχεια επιτρέπει την χρωματική οριοθέτηση εντός της εικόνας με την χρήση της cv2.threshold, και της δυαδικής μεθόδου του Otsu για την καλύτερη δυνατή οριοθέτηση μεταξύ των μαύρων pixel του φόντου και των λευκών pixel των χαρακτήρων. Η συγκεκριμένη μέθοδος δέχεται ως είσοδο μια εικόνα σε κλίμακα του γκρι (grayscale) και ένα κατώφλι (threshold). Η μέθοδος Otsu (Murzova & Seth, 2020) ελέγχει τη χρωματική ένταση κάθε pixel της εικόνας και τη συγκρίνει με το κατώφλι. Αν η ένταση είναι μεγαλύτερη από το κατώφλι, τότε το pixel ορίζεται ως λευκό, ενώ αν είναι χαμηλότερη, τότε ορίζεται ως μαύρο. Το αποτέλεσμα αποθηκεύεται στην μεταβλητή thresh, και στην συνέχεια δέχεται τον μορφολογικό μετασχηματισμό closing, ο οποίος επιχειρεί να ενώσει χωρισμένα τμήματα που μπορεί να υπάρχουν σε έναν χαρακτήρα. Για παράδειγμα, ο πεζός, λατινικός χαρακτήρας «i» αποτελείται από δύο σκέλη. Την κάθετη γραμμή και την τελεία στην κορυφή. Ο συγκεκριμένος μετασχηματισμός ενώνει τα δύο αυτά τμήματα, κάνοντας τον χαρακτήρα να μοιάζει με τον χαρακτήρα «l». Ο μετασχηματισμός λειτουργεί πραγματοποιώντας διεύρυνση (dilation) στην εικόνα, ακολουθούμενη από συστολή (erosion). (Rosebrock, 2021) Ο συνδυασμός των δύο παράγει την ένωση που επιθυμούμε για το επόμενο βήμα της συνάρτησης, όπου χρησιμοποιείται η συνάρτηση cv2.findContours, για τον εντοπισμό του περιγράμματος κάθε χαρακτήρα. Μιας και κάθε χαρακτήρας έχει το δικό του, ξεχωριστό πλάτος, η τοποθέτηση των χαρακτήρων ακριβώς στο κέντρο της εικόνας αποτελεί μια περίπλοκη διαδικασία. Ο

στόχος της γεννήτριας παραγωγής εικόνων είναι να προσθέσει ολόκληρους τους χαρακτήρες εντός της εικόνας ανεξαρτήτου τοποθεσίας. Με την συνάρτηση `Augment_Image` χρησιμοποιούμε την `cv2.findContours` για να ορίσουμε τον χαρακτήρα ως κέντρο μιας νέας εικόνας, με την ακόλουθη διαδικασία:

Το περίγραμμα αποθηκεύεται στην `cnts` με μορφή συντεταγμένων που φανερώνουν σημεία επάνω στις ακμές και τις γωνίες ενός χαρακτήρα.



Εικόνα 12: Ο χαρακτήρας με το εντοπισμένο περίγραμμα.

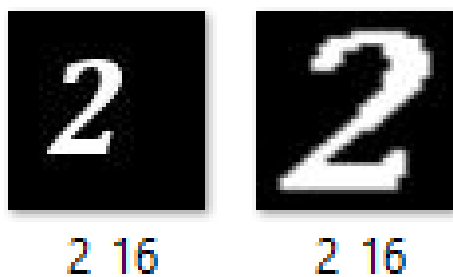
Στη συνέχεια, με την χρήση της `cv2.boundingRect`, το περίγραμμα του χαρακτήρα περιτριγυρίζεται από ένα ορθογώνιο πλαίσιο, και έπειτα δημιουργείται μια νέα εικόνα `Region Of Interest - roi`, η οποία περιλαμβάνει τον αρχικό χαρακτήρα στο κέντρο της εικόνας. Με το τρέχον βήμα γίνεται μια προσπάθεια ομοιομορφίας τόσο των χαρακτήρων προς εκπαίδευση, όσο και προς πρόβλεψη, καθώς η ίδια τεχνική χρησιμοποιείται και στην αναγνώριση.



Εικόνα 13: Η ορθογωνιοποίηση του χαρακτήρα.

Τέλος, η νέα πλέον εικόνα παίρνει το απαραίτητο μέγεθος 50x50 pixels από τις μεταβλητές `img_width` και `img_height` μέσω της συνάρτησης `cv2.resize`, και

στη συνέχεια η εικόνα λαμβάνει ξανά την μορφή PIL από OpenCV προτού επιστρέψει στην γεννήτρια για αποθήκευση.



Εικόνα 14: Πριν και μετά την συνάρτηση Augment\_Image.

### 4.3 ΚΑΤΑΣΚΕΥΗ ΜΟΝΤΕΛΟΥ

Κατά την διάρκεια της αρχικής μου έρευνας επάνω στο αντικείμενο της οπτικής αναγνώρισης χαρακτήρων και στις διαφορετικές διαθέσιμες τεχνολογίες υλοποίησης εφαρμογών OCR, πραγματοποιήθηκε μια πρώτη «γνωριμία» με το Keras, και το Pytorch. Από τα πρώτα βήματα που πάρθηκαν για την κατασκευή της εφαρμογής, ήταν μια μικρή υλοποίηση ενός demo με το Keras, στο οποίο εξετάστηκε ευκολία υλοποίησης, επεξεργασίας, καθώς και τα αποτελέσματα της εκπαίδευσης σε ένα μικρό dataset.

Παρατηρήθηκε πως υπήρχε μια ιδιαίτερη απλότητα στην κατασκευή νευρωνικών δικτύων, εξαιτίας του αρθρωτού (modular) τρόπου που όλα τα κομμάτια αλληλοεπιδρούν μεταξύ τους και παράγουν διαφορετικά αποτελέσματα ανάλογα με τις ορισμένες παραμέτρους. Ένα γεγονός που καθιστά την υλοποίηση με το Keras ιδιαίτερα απλή και εύκολη λόγω του περιβάλλοντος που παρέχει.

Αργότερα, με την χρήση του Pytorch, υπήρξε μια ιδιαίτερη πολυπλοκότητα στον τρόπο ανάπτυξης νευρωνικών δικτύων χρησιμοποιώντας την συγκεκριμένη τεχνολογία, με αποτέλεσμα να επιλέξω την επιστροφή μου στην πρώτη μου επιλογή, λαμβάνοντας την τελική απόφαση να υλοποιήσω την εφαρμογή με το Keras.

Από τεχνικής άποψης, αυτές οι δύο τεχνολογίες χαρακτηρίζονται ως εξής: (Sayantini, 2020)

### **Keras:**

- Τεχνολογία ανοιχτού κώδικα (Open Source), κατασκευασμένη σε γλώσσα Python.
- Χαρακτηρίζεται ως μια υψηλόβαθμη διεπαφή προγραμματισμού εφαρμογών (high-level API), το οποίο μπορεί να λειτουργήσει επάνω σε τεχνολογίες όπως το TensorFlow, το CNTK, και το Theano.
- Η ταχύτητα του Keras θεωρείται πιο αργή συγκριτικά με άλλες τεχνολογίες υλοποίησης και εκπαίδευσης νευρωνικών δικτύων, με αποτέλεσμα να χρησιμοποιείται κυρίως σε μικρότερα dataset.
- Η αρχιτεκτονική του χαρακτηρίζεται ως αρκετά απλή, ενώ παράλληλα θεωρείται ευανάγνωστη και περιεκτική, κάτι που επιτρέπει στην λιγότερο συχνή αποσφαλμάτωση του δικτύου.

### **PyTorch:**

- Επίσης τεχνολογία ανοιχτού κώδικα (Open Source), κατασκευασμένη για την γλώσσα Python, βασισμένη στην τεχνολογία Torch.
- Αντίθετα με το Keras, το PyTorch χαρακτηρίζεται ως μια χαμηλόβαθμη διεπαφή προγραμματισμού εφαρμογών (lower-level API), επικεντρωμένο κυρίως στην απευθείας λειτουργία με array expressions.
- Θεωρείται ιδιαίτερα γρήγορη τεχνολογία, σχεδιασμένη με υψηλή αποδοτικότητα, με αποτέλεσμα να λειτουργεί καλύτερα με μεγάλου μεγέθους dataset.
- Η αρχιτεκτονική του είναι ιδιαίτερα περίπλοκη, σχετικά δυσανάγνωστη, αλλά παρέχει μεγαλύτερες δυνατότητες αποσφαλμάτωσης του δικτύου.

#### **4.3.1 Το δίκτυο**

Έχοντας πλέον επιλέξει την τεχνολογία βαθιάς μάθησης με την οποία θα υλοποιούσα την εφαρμογή οπτικής αναγνώρισης, ξεκίνησα να υλοποιώ το νευρωνικό μοντέλο που θα μπορούσε να υποστηρίξει το συγκεκριμένο εγχείρημα. Συγκεκριμένα κατασκευάστηκαν 5 διαφορετικά δίκτυα συνολικά, με το τελευταίο να αποτελεί και το δίκτυο που χρησιμοποιείται κατά την εκτέλεση της εφαρμογής, ενώ όλα τα μοντέλα



βρίσκονται στο αρχείο `predict.py` της εφαρμογής. Τα συγκεκριμένα δίκτυα κατασκευάστηκαν κατά την διάρκεια της συνολικής υλοποίησης της εφαρμογής ως έναν τρόπο εντοπισμού του πιο αξιόπιστου μοντέλου από πλευράς τελικών αποτελεσμάτων σύμφωνα με τις απαιτήσεις της εφαρμογής. Ως κριτήριο επιλογής της τελικής αρχιτεκτονικής που θα χρησιμοποιούνταν στην εφαρμογή αποτέλεσαν τα στατιστικά στοιχεία κάθε δικτύου κατά την διάρκεια της εκπαίδευσης. Αναλύοντας τόσο τα αποτελέσματα των εποχών, όσο και το εξαγόμενο μοντέλο σε συνθήκες πρόβλεψης, το δίκτυο τροποποιούνταν ή άλλαζε οριστικά, με το προηγούμενο δίκτυο να παραμένει στο αρχείο.

Ο πίνακας 5 παρουσιάζει το περιεχόμενο του συγκεκριμένου μοντέλου πριν την εκπαίδευση με αριθμητικούς χαρακτήρες, μέσω της συνάρτησης `summary` που είναι διαθέσιμη έπειτα από την δημιουργία ενός νευρωνικού μοντέλου μέσω της βιβλιοθήκης Keras.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 50, 50, 32)	896
max_pooling2d (MaxPooling2D)	(None, 25, 25, 32)	0
conv2d_1 (Conv2D)	(None, 25, 25, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 64)	0
conv2d_2 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dropout (Dropout)	(None, 2304)	0
dense (Dense)	(None, 128)	295040
dense_1 (Dense)	(None, 10)	1290
Total params: 352,650		
Trainable params: 352,650		
Non-trainable params: 0		

Πίνακας 5: Το σειριακό μοντέλο βαθιάς μάθησης της εφαρμογής.

Το συγκεκριμένο μοντέλο είναι σειριακού τύπου (sequential), αυτό σημαίνει πως έχει την ιδιαιτερότητα να περιέχει έναν μόνο τανυστή (tensor) τόσο στην είσοδο, όσο και στην έξοδο του. Η επιλογή του συγκεκριμένου τύπου δικτύου είναι λόγω του τρόπου λειτουργίας της συγκεκριμένης αναγνώρισης χαρακτήρων, καθώς θα εισάγεται μόνο μια εικόνα κάθε φορά ως είσοδος, και θα εξάγεται ένα μόνο αποτέλεσμα κατά τη διαδικασία της πρόβλεψης.

Χρησιμοποιούνται τρεις στρώσεις Conv2D για την κατασκευή συνελκτικών πυρήνων διαστάσεων 3x3 (convolution kernels), οι οποίοι λειτουργούν ως ένα φίλτρο που τοποθετείται στο επάνω αριστερά τμήμα της εικόνας εισόδου, δηλαδή την έξοδο των προηγούμενων στρώσεων. (Thevenot, 2020) Ο πυρήνας περνάει επάνω από ολόκληρη την εικόνα, με φορά αριστερά προς δεξιά και επάνω προς κάτω, εφαρμόζοντας το γινόμενο της συνέλιξης (convolution product), παράγοντας μια «φίλτραρισμένη» εικόνα ως έξοδο. Στο μοντέλο της εφαρμογής χρησιμοποιείται η συνάρτηση ενεργοποίησης Relu σε κάθε Conv2D.

3	3	2	1	0
0	0 <sub>0</sub>	1 <sub>1</sub>	3 <sub>2</sub>	1
3	1 <sub>2</sub>	2 <sub>2</sub>	2 <sub>0</sub>	3
2	0 <sub>0</sub>	0 <sub>1</sub>	2 <sub>2</sub>	2
2	0	0	0	1

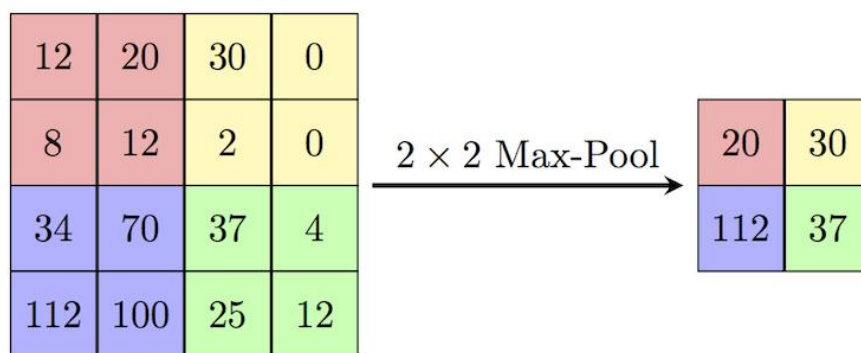
12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Εικόνα 15: Τρόπος λειτουργίας του Conv2D.

Η πρώτη στρώση Conv2D περιλαμβάνει το μέγεθος της εισόδου που περιμένει να λάβει το μοντέλο κατά την διαδικασία της εκπαίδευσης και της πρόβλεψης. Αυτή η

είσοδος ορίζεται ως 50x50x3, δηλαδή 50 pixels ύψος, 50 pixels πλάτος, και 3 χρωματικά κανάλια.

Αμέσως μετά από κάθε στρώση Conv2D χρησιμοποιείται μια στρώση MaxPooling2D (3 στρώσεις συνολικά), μεγέθους 2x2, η οποία έχει ως στόχο την μείωση των δειγμάτων που αντιπροσωπεύει κάθε είσοδο. Αυτό πραγματοποιείται μειώνοντας τις διαστάσεις της εισόδου, επιτρέποντας έτσι στο μοντέλο βαθιάς μάθησης να δημιουργεί εικασίες σχετικά με χαρακτηριστικά που υπάρχουν εντός της περιοχής που αντιπροσωπεύει το MaxPooling2D.



Εικόνα 16: Γραφική αναπαράσταση της λειτουργίας MaxPooling2D.

Η κύρια λειτουργία του MaxPooling είναι η εξαγωγή των pixel και των περιοχών με τα κυρίαρχα χαρακτηριστικά της εικόνας μετά τη συνέλιξη, παρόλα αυτά η συγκεκριμένη τεχνική βοηθάει και με το πρόβλημα της υπερπροσαρμογής του μοντέλου (overfitting), μέσω της παραχώρησης μιας συνοπτικής έκδοσης της εισόδου, ενώ ταυτόχρονα μειώνει το υπολογιστικό κόστος ελαχιστοποιώντας το πλήθος των παραμέτρων του δικτύου. Αυτό επιτυγχάνεται χωρίζοντας την αρχική εικόνα σε τμήματα ίσων διαστάσεων (πχ. 2x2), και εξάγοντας την μέγιστη τιμή κάθε τμήματος σε έναν ξεχωριστό πίνακα προς την έξοδο.

Στην συνέχεια χρησιμοποιούνται οι συναρτήσεις Flatten και Dropout. Η Flatten συμπίεζει τις διαστάσεις της εισόδου σε μια μόνο διάσταση, για παράδειγμα στον πίνακα 5, η είσοδος (None, 6, 6, 64) μετατρέπεται σε (None, 2304) πολλαπλασιάζοντας τις τιμές κάθε διάστασης μεταξύ τους. Αμέσως μετά, η Dropout έχει ως στόχο την εξαίρεση των νευρώνων από την διαδικασία εκπαίδευσης κατά τη

διάρκεια κάθε εποχής. Κάθε νευρώνας στο συγκεκριμένο στρώμα του δικτύου έχει μια πιθανότητα (συνήθως 20% με 50%) να απενεργοποιηθεί, δηλαδή να μην επηρεάσει την έξοδο του στρώματος, όμως οι απενεργοποιημένοι νευρώνες διατηρούν τα βάρη τους, καθώς μπορούν να ενεργοποιηθούν ξανά στην επόμενη εποχή. Στο συγκεκριμένο μοντέλο, χρησιμοποιείται Dropout πιθανότητας 20%.

Η Dropout προσπαθεί να περιορίσει το φαινόμενο της υπερπροσαρμογής του δικτύου, με σκοπό την καλύτερη δυνατή γενίκευση στο συγκεκριμένο dataset.

Τέλος το μοντέλο αποτελείται από δύο βαθιές στρώσεις νευρώνων Dense, με την μια από αυτές να αποτελείται από 128 εξόδους, και την δεύτερη από εξόδους ίσες με το πλήθος των συνολικών διαθέσιμων χαρακτήρων, μιας και επιθυμούμε να λαμβάνουμε μια τιμή που θα αντιστοιχεί σε έναν μόνο χαρακτήρα. Κάθε νευρώνας μιας Dense στρώσης συνδέεται με τις εξόδους όλων των νευρώνων της προηγούμενης στρώσης, ενώ η συγκεκριμένη στρώση θεωρείται από τις πιο συχνά χρησιμοποιούμενες σε νευρωνικά δίκτυα βαθιάς μάθησης.

#### **4.3.2 Datagen, Ανακλητές, Μεταγλώττιση**

Έχοντας πλέον ολοκληρώσει το μοντέλο που θα χρησιμοποιηθεί κατά την διαδικασία εκπαίδευσης, το επόμενο στάδιο αποτελεί τον τρόπο με τον οποίο οι παραγόμενες εικόνες της γεννήτριας θα εισαχθούν εντός του δικτύου.

Για να πραγματοποιηθεί το συγκεκριμένο εγχείρημα, θα πρέπει οι εικόνες να χωριστούν σε δύο σετ δεδομένων, συγκεκριμένα σε train\_set και validation\_set. Αυτός ο χωρισμός χρειάζεται για την ορθή εκπαίδευση του μοντέλου επάνω στο κύριο train\_set, ενώ το validation\_set λειτουργεί ως ένα άγνωστο σετ δεδομένων για να αποφευχθεί η υπερεκπαίδευση του δικτύου.

```

datagen = ImageDataGenerator(
    rescale = 1. / 255,
    validation_split = 0.15,
    rotation_range=8,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
)

ds_train = datagen.flow_from_directory(
    dataset_directory_root,
    target_size = (img_width, img_height),
    batch_size = batch,
    class_mode = 'categorical',
    shuffle = True,
    seed = 123,
    subset = 'training',
)

ds_validate = datagen.flow_from_directory(
    dataset_directory_root,
    target_size = (img_width, img_height),
    batch_size = batch,
    class_mode = 'categorical',
    shuffle = True,
    seed = 123,
    subset = 'validation',
)

```

Πίνακας 6: Datagen, train\_set και validation\_set

Προτού χωρίσουμε τα δεδομένα μπορούμε να ορίσουμε μια μεταβλητή `datagen`, με την οποία έχουμε την δυνατότητα να δημιουργήσουμε περισσότερες εικόνες από το ήδη υπάρχον dataset μέσω της συνάρτησης `ImageDataGenerator` του Keras. Η συγκεκριμένη τεχνική χρησιμοποιείται για την ενίσχυση του dataset δημιουργώντας αντίγραφα των εικόνων και στην συνέχεια αλλάζοντας τα ελάχιστα ως προς την περιστροφή της εικόνας, το ύψος, το πλάτος, το επίπεδο zoom, κλπ. Τα συγκεκριμένα αντίγραφα εισάγονται στη συνέχεια κανονικά μέσα στο μοντέλο κατά την εκπαίδευση. Επί πλέον, εντός του `datagen` ορίζεται και το `validation_split`, το οποίο είναι υπεύθυνο για τον χωρισμό του συνολικού dataset σε `training_set` και `validation_set`. Εντός του προγράμματος, ο χωρισμός γίνεται με τιμή 0,15, δηλαδή το 15% των εικόνων θα εισαχθούν στο `validation_set`.

Το datagen σε σχεδιασμό με το flow\_from\_directory μπορούν να παράγουν αυτά τα δύο απαραίτητα σετ μαζί με ορισμένες παραμέτρους που ορίζονται ανάλογα με τα χαρακτηριστικά που επιθυμούμε να έχουν. Στην εφαρμογή, δηλώνουμε πως οι εικόνες μας βρίσκονται στον φάκελο /Dataset, μεγέθους 50x50, και στην συνέχεια χωρίζουμε τις εικόνες σε μικρότερες ομάδες (batches) των 22 εικόνων. Τέλος ορίζουμε πως τα δύο dataset θα κατηγοριοποιούνται σύμφωνα με την κατηγορία, δηλαδή τον χαρακτήρα, όπου ανήκουν.

Το τελευταίο βήμα προτού ξεκινήσει η διαδικασία εκπαίδευσης είναι ο ορισμός των callbacks. Τα callbacks, ή αλλιώς ανακλητές, λειτουργούν σαν συναρτήσεις που καλούνται όταν πληρούνται ορισμένα κριτήρια κατά την εκπαίδευση του μοντέλου. Μπορούν να χρησιμοποιηθούν για την δημιουργία σημείων ελέγχου, τροποποίηση του ρυθμού εκμάθησης (learning rate), ακόμα και για πρόωρη ολοκλήρωση της εκπαίδευσης.

```
#Defining the model callbacks.
class Callback(tf.keras.callbacks.Callback):
    #Callbacks act as functions that get executed during
    #specific moments during training.
    #Here they work as a way of early stopping the training
    #once the model has reached accuracy and loss within
    #a specified threshold.
    def on_epoch_end(self, epoch, logs={}):
        if ((logs.get('accuracy') > Accuracy_Threshold) &
            (logs.get('loss') < loss_threshold) &
            (logs.get('val_accuracy') > Accuracy_Threshold)
            & (logs.get('val_loss') < loss_threshold)):
            self.model.stop_training = True

#Initiating the model callbacks.
callbacks = Callback()
```

Πίνακας 7: Οι ανακλητές του προγράμματος.

Για την υλοποίηση της εφαρμογής χρησιμοποιείται πρόωρος τερματισμός της εκμάθησης όταν στο τέλος μιας εποχής οι τιμές των accuracy και val\_accuracy είναι μεγαλύτερες από το Accuracy\_Threshold που ορίζεται με τιμή 0,999, και οι τιμές των loss και val\_loss είναι μικρότερες του loss\_threshold με τιμή 0,02. Η επιλογή του συγκεκριμένου callback έγινε με βάση το γεγονός πως πιθανώς να μην είναι απαραίτητη η ολοκλήρωση των ορισμένων εποχών για τα καλύτερα

αποτελέσματα του μοντέλου, καθώς υπάρχει μεγάλη πιθανότητα τα αποτελέσματα να αρχίσουν σταδιακά να χειροτερεύουν κατά την συμπλήρωση του ορίου εποχών, ενώ παράλληλα χρησιμοποιούνται πόροι του συστήματος για μεγαλύτερο χρονικό διάστημα από όσο χρειάζεται. Η απόδοση του δικτύου στο validation set χρησιμοποιείται για τον πρόωρο τερματισμό της εκπαίδευσης για την πρόληψη της υπερεκπαίδευσης.

```
#The model will be compiled with the following parameters.
model.compile(
    loss='categorical_crossentropy',
    optimizer = keras.optimizers.Adam(learning_rate =
        0.0005), #rmsprop, adam, SGD
    metrics = ['accuracy']
)
```

Πίνακας 8: Μεταγλώττιση του μοντέλου.

Τέλος πραγματοποιείται η παραμετροποίηση του νευρωνικού δικτύου για την εκπαίδευση, χρησιμοποιώντας `categorical_crossentropy` ως την συνάρτηση απώλειας του μοντέλου, τον βελτιστοποιητή Adam με ρυθμό εκμάθησης 0,0005, και την μεταβλητή `metrics` με τιμή `accuracy` για τον υπολογισμό της ακρίβειας του μοντέλου. Η συνάρτηση `compile()` θέτει τις παραμέτρους που της έχουν οριστεί στο δίκτυο για να καθορίσουν τον τρόπο με τον οποίο το δίκτυο θα εκπαιδευτεί. Τέλος, η συνάρτηση απώλειας `categorical_crossentropy` χρησιμοποιείται για τον υπολογισμό της εντροπικής απώλειας μεταξύ των προβλέψεων του δικτύου για μια είσοδο και της σωστής κατηγορίας στην οποία ανήκει αυτή η είσοδος.

#### 4.3.3 Αξιολόγηση του μοντέλου

Η εκπαίδευση του μοντέλου πραγματοποιείται αφού πρώτα το πρόγραμμα ρωτήσει τον χρήστη το πλήθος των εποχών που επιθυμεί για την εκπαίδευση. Επιβεβαιώνοντας την επιλογή, ξεκινάει η διαδικασία.

```
training_results = model.fit(ds_train,
                             validation_data = ds_validate,
                             epochs = epochs, verbose = 2,
                             callbacks=[callbacks])
```

Πίνακας 9: Η συνάρτηση εκπαίδευσης.

Συγκεκριμένα η συνάρτηση `fit` λαμβάνει ως ορίσματα το `dataset` εκπαίδευσης `ds_train`, το `dataset` επαλήθευσης `ds_validate`, το πλήθος των εποχών που έχει ορίσει ο χρήστης, την παράμετρο `verbose` που εμφανίζει τα αποτελέσματα μετά από κάθε εποχή, και τέλος τους ανακλητές `callbacks`. Τα αποτελέσματα της εκπαίδευσης αποθηκεύονται στην μεταβλητή `training_results` για μετέπειτα χρήση στην αναπαράσταση των αποτελεσμάτων σε γραφική μορφή.

```
Epoch 1/10
35/35 - 5s - loss: 1.7895 - accuracy: 0.4133 - val_loss:
0.6705 - val_accuracy: 0.7923
Epoch 2/10
35/35 - 2s - loss: 0.5893 - accuracy: 0.8027 - val_loss:
0.4610 - val_accuracy: 0.8308
Epoch 3/10
35/35 - 2s - loss: 0.3460 - accuracy: 0.8827 - val_loss:
0.0824 - val_accuracy: 0.9692
Epoch 4/10
35/35 - 2s - loss: 0.2197 - accuracy: 0.9267 - val_loss:
0.0631 - val_accuracy: 0.9769
Epoch 5/10
35/35 - 2s - loss: 0.1579 - accuracy: 0.9480 - val_loss:
0.0703 - val_accuracy: 0.9846
Epoch 6/10
35/35 - 2s - loss: 0.1116 - accuracy: 0.9587 - val_loss:
0.0150 - val_accuracy: 1.0000
Epoch 7/10
35/35 - 2s - loss: 0.1008 - accuracy: 0.9693 - val_loss:
0.0435 - val_accuracy: 0.9769
Epoch 8/10
35/35 - 2s - loss: 0.0663 - accuracy: 0.9787 - val_loss:
0.0044 - val_accuracy: 1.0000
Epoch 9/10
35/35 - 2s - loss: 0.0480 - accuracy: 0.9853 - val_loss:
0.0129 - val_accuracy: 1.0000
Epoch 10/10
35/35 - 2s - loss: 0.0992 - accuracy: 0.9667 - val_loss:
0.0588 - val_accuracy: 0.9769

Training is Complete!
```

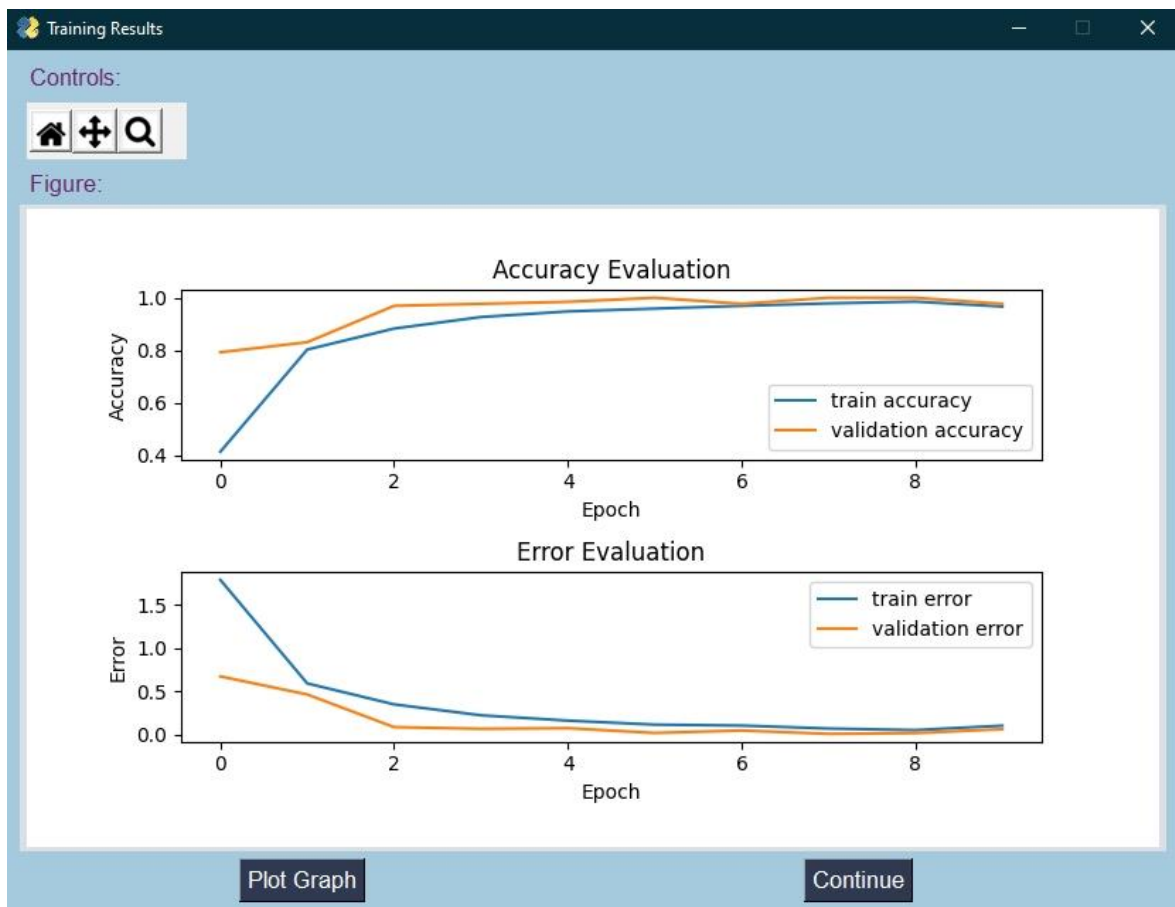
Πίνακας 10: Αποτελέσματα εκπαίδευσης αριθμητικών χαρακτήρων για 10 εποχές.



Ο πίνακας 10 παρουσιάζει τα αποτελέσματα που λαμβάνουμε στο τέλος κάθε εποχής κατά την διάρκεια της εκπαίδευσης. Το `Epoch` αναφέρεται στα αποτελέσματα της τρέχουσας εποχής, ο αριθμός `35/35` αναφέρεται στο πλήθος των ομάδων (batches) που επεξεργάστηκε το μοντέλο, και στην συνέχεια εμφανίζονται τα στατιστικά στοιχεία για την απώλεια (`loss`), ακρίβεια (`accuracy`), απώλεια επαλήθευσης (`val_loss`), και ακρίβεια επαλήθευσης (`val_accuracy`).

Για ένα σωστά εκπαιδευμένο μοντέλο χρειάζεται να λαμβάνουμε `loss` και `val_loss` κοντά στην τιμή 0, και ταυτόχρονα `accuracy` και `val_accuracy` κοντά στην τιμή 1.

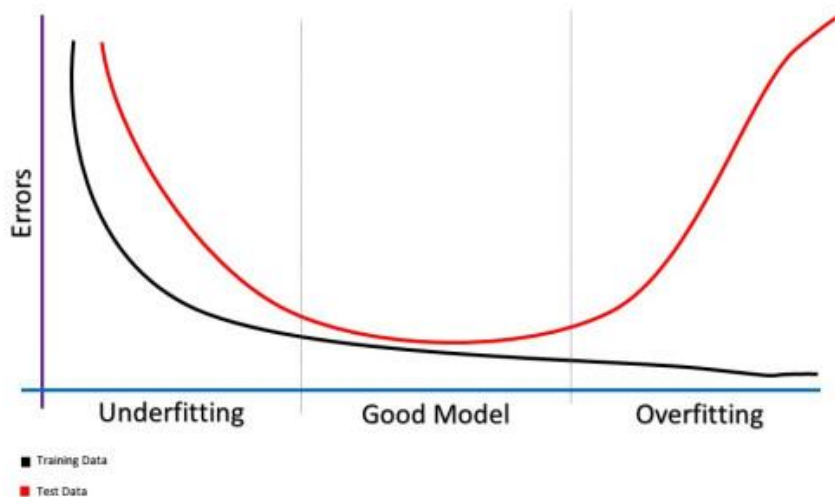
Με την ολοκλήρωση της εκπαίδευσης μπορούμε να αναπαραστήσουμε τα αποτελέσματα ως γράφημα, για την καλύτερη κατανόηση της απόδοσης του δικτύου. Περισσότερα στο κεφάλαιο 4.5 – Γραφική Διεπαφή.



Εικόνα 17: Γραφική αναπαράσταση αποτελεσμάτων Πίνακα 10.

#### 4.3.4 Υποπροσαρμογή και Υπερπροσαρμογή

Ένα από τα συχνότερα προβλήματα που μπορεί να εμφανιστούν κατά δημιουργία και την επεξεργασία των νευρωνικών μοντέλων είναι το φαινόμενο της υποπροσαρμογής (underfitting) και της υπερπροσαρμογής (overfitting)



Εικόνα 18: Υποπροσαρμογή και Υπερπροσαρμογή μοντέλου.

Αυτές οι δύο έννοιες αναφέρονται κυρίως στον τρόπο με τον οποίο ένα μοντέλο εκπαιδεύεται επάνω σε ένα dataset. Όταν υπάρχει υποπροσαρμογή του δικτύου, τότε το μοντέλο πιθανώς να είναι πολύ αδύναμο για την επεξεργασία του συγκεκριμένου dataset, κάτι που φαίνεται από την ύπαρξη μεγάλου ποσοστού σφαλμάτων τόσο στο σετ εκμάθησης, όσο και στο σετ επαλήθευσης. Ως αποτέλεσμα, το τρέχον μοντέλο να μην μπορεί να μάθει επάνω στο περιεχόμενο του dataset.

Από την άλλη πλευρά όταν το μοντέλο βρίσκεται υπό κατάσταση υπερπροσαρμογής, τότε πραγματοποιεί πολύ λίγα σφάλματα στο σετ εκμάθησης, αποτυγχάνει στο σετ επαλήθευσης. Αυτό δηλώνει πως το μοντέλο «παπαγαλίζει» το training set, χωρίς να μαθαίνει να αναγνωρίζει πρότυπα που μπορούν να το βοηθήσουν στην γενίκευση δεδομένων που δεν γνωρίζει.

Το ιδανικό σενάριο εκπαίδευσης περιλαμβάνει παρόμοια απόδοση τόσο στην εκπαίδευση όσο και στην επαλήθευση, κάτι που επιτυγχάνεται έπειτα από σωστή προσαρμογή του δικτύου επάνω στα ζητούμενα δεδομένα.

#### **4.4 ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ**

Έχοντας πλέον αυτοματοποιήσει την παραγωγή εικόνων για το dataset της εφαρμογής, και έχοντας καθορίσει το νευρωνικό μοντέλο που θα χρησιμοποιηθεί για την εκπαίδευση και την πρόβλεψη, ήρθε η στιγμή να αναφερθούμε στα δεδομένα εισόδου. Δηλαδή τις εικόνες που θα δέχεται το μοντέλο, θα τις επεξεργάζεται και θα επιστρέφει μια πρόβλεψη για το περιεχόμενο τους.

Αξίζει να σημειωθεί πως το συγκεκριμένο σκέλος ίσως είναι και το δυσκολότερο ως προς την υλοποίηση, καθώς τόσο ο εντοπισμός χαρακτήρων, όσο και ο διαχωρισμός τους από άλλους χαρακτήρες μπορεί να αποτελέσει μια ιδιαίτερα περίπλοκη διαδικασία.

Ο τρόπος με τον οποίο πραγματοποιήθηκε η προσέγγιση του συγκεκριμένου προβλήματος ήταν ο ακόλουθος. Αρχικά δόθηκε στο χρήστη η δυνατότητα ορισμού μιας περιοχής εντός της εικόνας που θα τον ενδιέφερε να πραγματοποιηθεί αναγνώριση. Περισσότερα στο κεφάλαιο 4.5 – Γραφική Διεπαφή. Η απόφαση πάρθηκε έπειτα από την ανακάλυψη πως η αρχική μέθοδος που χρησιμοποιήθηκε για τον αυτόματο εντοπισμό χαρακτήρων επέστρεφε πολλές, μικρές περιοχές εντός της εικόνας, οι οποίες δεν περιείχαν κάποιο χαρακτήρα. Αυτές οι περιοχές αποτελούνταν κυρίως από θόρυβο που υπήρχε στην εικόνα, με ορισμένες από αυτές να είναι ιδιαίτερα δύσκολο να αφαιρεθούν, και εξίσου περίπλοκο να μην εντοπιστούν από την εφαρμογή. Με την τρέχουσα μέθοδο χειροκίνητου εντοπισμού, μπορεί πλέον ο χρήστης να περιορίσει την περιοχή στην οποία επιθυμεί να ανιχνεύσει χαρακτήρες.

Όταν έχουν επιλεγεί από το χρήστη οι περιοχές προς αναγνώριση, πραγματοποιείται η ίδια διαδικασία που αναφέραμε στο κεφάλαιο 4.2.2 – Ενίσχυση Εικόνων, όπου προσπαθούμε να φέρουμε τις επιλεγμένες περιοχές στην ίδια κατάσταση με τις εικόνες χαρακτήρων που παράγει η γεννήτρια, για τα καλύτερα δυνατά αποτελέσματα. Η συγκεκριμένη διαδικασία πραγματοποιείται από την συνάρτηση `image_processing` του αρχείου `detect.py`.

Κάθε επιλεγμένη περιοχή εισάγεται στην συνάρτηση `contour_detection`, όπου ανιχνεύεται το περίγραμμα των χαρακτήρων, και στη συνέχεια εισάγονται στην κύρια συνάρτηση `character_extraction` για την απομόνωση των χαρακτήρων που εντοπίστηκαν. Προτού όμως ξεκινήσει η διαδικασία, καλείται η συνάρτηση `sort_contours` για να δημιουργήσει τα ορθογώνια περιγράμματα γύρω από τους χαρακτήρες, και στη συνέχεια να τα ταξινομήι στην σωστή τους θέση από τα αριστερά προς τα δεξιά, έτσι ώστε οι χαρακτήρες να εμφανίζονται με την ίδια σειρά που υπάρχουν στην εικόνα.

```
def sort_contours(contours):  
  
    contours_bound_rect = []  
  
    for c in contours:  
        (x,y,w,h) = cv2.boundingRect(c)  
  
        contours_bound_rect.append([x,y,w,h])  
  
    sorted_contours = sorted(contours_bound_rect,  
                             key=itemgetter(0))  
  
    return sorted_contours
```

Πίνακας 11: Η συνάρτηση `sort_contours`.

Η συγκεκριμένη ενέργεια είναι ιδιαίτερα σημαντική, μιας και η σειρά εντοπισμού των χαρακτήρων μπορεί να είναι εσφαλμένη λόγω των διάφορων υψών που έχουν οι χαρακτήρες και του τρόπου λειτουργίας της `cv2.boundingRect()`. Με την `sort_contours` γίνεται μια προσπάθεια τοποθέτησης αυτών των χαρακτήρων στην σωστή τους θέση, για την τελική εμφάνιση των προβλέψεων.

Επιστρέφοντας στην κύρια συνάρτηση με τα ταξινομημένα ορθογώνια περιγράμματα, ξεκινάει πλέον η διαδικασία διαχωρισμού.

```
for sr in sorted_rectangles:  
  
    #Expanding the image width and height  
    #similarly to the Augment_Image process  
    x = sr[0]  
    y = sr[1]  
    w = sr[2]  
    h = sr[3]
```

```

x = x-2
y = y-2
h = h+4
w = w+4

if((x-3) < 0):
    x = 0
if((y-3) < 0):
    y = 0

#Creating Regions of Interest around the character
roi = selection_img[y:y+h, x:x+w]

#Calculating the avelrage width of all characters
totalWidth = totalWidth + roi.shape[1]

try:
    # Avg + 25% of Avg
    averageWidth = int(int(totalWidth /
                           len(sorted_rectangles))
                       + (0.25 * int(totalWidth /
                                     len(sorted_rectangles))))
except:
    pass

#Second loop to split characters if necessary
for sr in sorted_rectangles:

    x = sr[0]
    y = sr[1]
    w = sr[2]
    h = sr[3]

    x = x-2
    y = y-2
    h = h+4
    w = w+4

    if((x-3) < 0):
        x = 0
    if((y-3) < 0):
        y = 0

    try:
        #Creating the final region of interest
        roi = selection_img[y:y+h, x:x+w]

        #If the image width exceeds the average
        if (roi.shape[1] > averageWidth):

```

```

        #Splitting the image into 2 images
        roi_1 = selection_img[y:y+h, x:int(x + w/2)]
        roi_2 = selection_img[y:y+h, x + int(w/2)
                               :x + w]

        #Resizing the 2 images to the proper size
        roi_1 = cv2.resize(roi_1, (50, 50))
        roi_2 = cv2.resize(roi_2, (50, 50))

        #Adding the images to a list for recognition
        rectangled_letters.append(roi_1)
        rectangled_letters.append(roi_2)

    else:
        #Image was below the average width,
        #resizing and appending the image
        #with no changes to it.
        roi = cv2.resize(roi, (50, 50))
        rectangled_letters.append(roi)

except:
    pass

Result = Predict(rectangled_letters)
return Result

```

Πίνακας 12: Διαδικασία απομόνωσης χαρακτήρων.

Το πρώτο στάδιο περιλαμβάνει τον υπολογισμό του μέσου όρου πλάτους από κάθε ορθογώνιο περίγραμμα. Μιας και σε ορισμένες γραμματοσειρές ορισμένοι συνδυασμοί χαρακτήρων μπορεί να βρίσκονται υπερβολικά κοντά μεταξύ τους, λόγω του proportional spacing (Computer Hope, 2017), σε σημείο που ίσως ενώνονται κατά τη διάρκεια της επεξεργασίας της αρχικής εικόνας, γίνεται μια προσπάθεια διαχωρισμού τέτοιων χαρακτήρων.

Η μεταβλητή `averageWidth` χρησιμοποιείται για αυτόν ακριβώς το σκοπό. Λειτουργεί ως ένα όριο πλάτους, κάτω από το οποίο θα πρέπει να βρίσκονται όλοι οι χαρακτήρες για να τους χαρακτηρίσουμε ως μονούς. Αν κάποιο περίγραμμα ξεπερνάει το συγκεκριμένο κατώφλι, τότε θεωρούμε πως περιλαμβάνονται δύο ενωμένοι χαρακτήρες, και σε αυτήν την περίπτωση χωρίζεται το περίγραμμα στην μέση με σκοπό να αφήσουμε το μοντέλο να αποφασίσει το ίδιο για το αποτέλεσμα.

Αυτή η τεχνική μπορεί να παρουσιάσει απρόοπτα αποτελέσματα στην διαδικασία της πρόβλεψης, καθώς δεν υπάρχει τρόπος να γνωρίζουμε αν οι ενωμένοι χαρακτήρες έχουν χωριστεί σωστά.

## 4.5 ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ

Ένα από τα μεγαλύτερα προβλήματα στην υλοποίηση εφαρμογών οπτικής αναγνώρισης είναι η διαδικασία του εντοπισμού χαρακτήρων σε μια είσοδο. Αποτελεί το τελικό στάδιο του όλου εγχειρήματος, όπως επίσης και το πιο σημαντικό, καθώς πλέον μπορούμε να εξετάσουμε πως το υλοποιημένο μοντέλο αλληλοεπιδρά με εισόδους που δίνονται από τον χρήστη.

Με στόχο τα καλύτερα δυνατά αποτελέσματα της εφαρμογής, και την παροχή δυνατότητας για ελεύθερη επιλογή περιοχών αναγνώρισης εντός της εικόνας εισόδου από τον χρήστη, έγινε προσθήκη μιας γραφικής διεπαφής απευθυνόμενη προς τον ίδιο τον τελικό χρήστη.

Φορτώνοντας μια εικόνα εντός του προγράμματος, εμφανίζεται ένα νέο παράθυρο μέσω της βιβλιοθήκης OpenCV, το οποίο εμφανίζει την επιλεγμένη εικόνα του χρήστη. Στο συγκεκριμένο παράθυρο ο χρήστης μπορεί να κρατήσει πατημένο το αριστερό κλικ και να σύρει τον κέρσορα του προς οποιαδήποτε κατεύθυνση για να δημιουργήσει τετράγωνα και ορθογώνιες περιοχές επάνω στην εικόνα.



Εικόνα 19: Παράθυρο γραφικής διεπαφής για δημιουργία περιοχών.

Ελευθερώνοντας το αριστερό κλικ δημιουργείται η περιοχή που όρισε ο χρήστης εντός της εικόνας. Όπως φαίνεται και στην Εικόνα 17, ο χρήστης μπορεί να ορίσει παραπάνω από μια περιοχές, ενώ ταυτόχρονα μπορούν δύο ή και περισσότερες περιοχές να αλληλεπικαλύπτονται. Η σειρά δημιουργίας των συγκεκριμένων περιοχών αποτελεί και

τη σειρά εμφάνισης τους κατά την διάρκεια των αποτελεσμάτων, στο κεφάλαιο 4.6 – Αποτελέσματα.

Σε περίπτωση που υπάρξει κάποια λάθος επιλογή από τον χρήστη, υπάρχει η δυνατότητα της αναίρεσης της τελευταίας περιοχής που έχει δημιουργηθεί κάθε χρονική στιγμή με τον συνδυασμό πλήκτρων ‘Ctrl + z’. Οι αναιρούμενες περιοχές δεν εμφανίζονται στα αποτελέσματα της πρόβλεψης από το μοντέλο.

Όταν ο χρήστης ολοκληρώσει τις επιλογές τους, μπορεί να προχωρήσει στην διαδικασία πρόβλεψης πατώντας το πλήκτρο ‘q’ ή ‘Q’, το οποίο θα κλείσει το συγκεκριμένο παράθυρο και θα εμφανίσει τα αποτελέσματα.

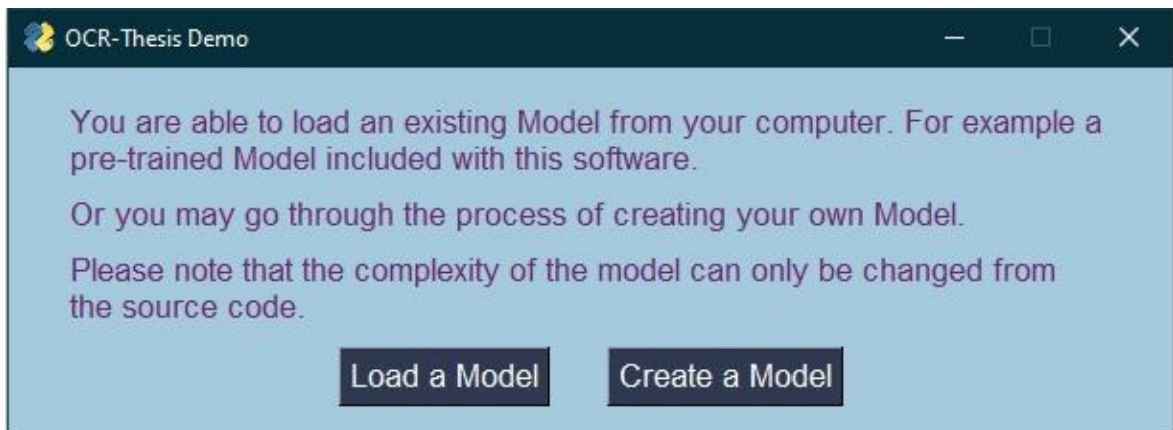
Από τα τελευταία στάδια της υλοποίησης της συγκεκριμένης εφαρμογής ήταν η ευκολία και ελευθερία χρήσης της εφαρμογής από οποιονδήποτε χρήστη. Ένας τρόπος επίτευξης αυτού του στόχου ήταν η προσθήκη γραφικής διεπαφής χρήστη (Graphical User Interface - GUI) για την αρμονική σύνδεση όλων των τμημάτων της εφαρμογής, και ταυτόχρονα παρέχοντας στον χρήστη δυνατότητες που θα ήταν ιδιαίτερα δύσκολο να υλοποιηθούν χωρίς κάποιο σύστημα γραφικής διεπαφής.

Έπειτα από μια μικρή έρευνα βιβλιοθηκών που παρείχαν τη δυνατότητα προσθήκης ενός GUI σε εφαρμογές γραμμένες σε Python, επιλέχθηκε η χρήση της βιβλιοθήκης PySimpleGUI.

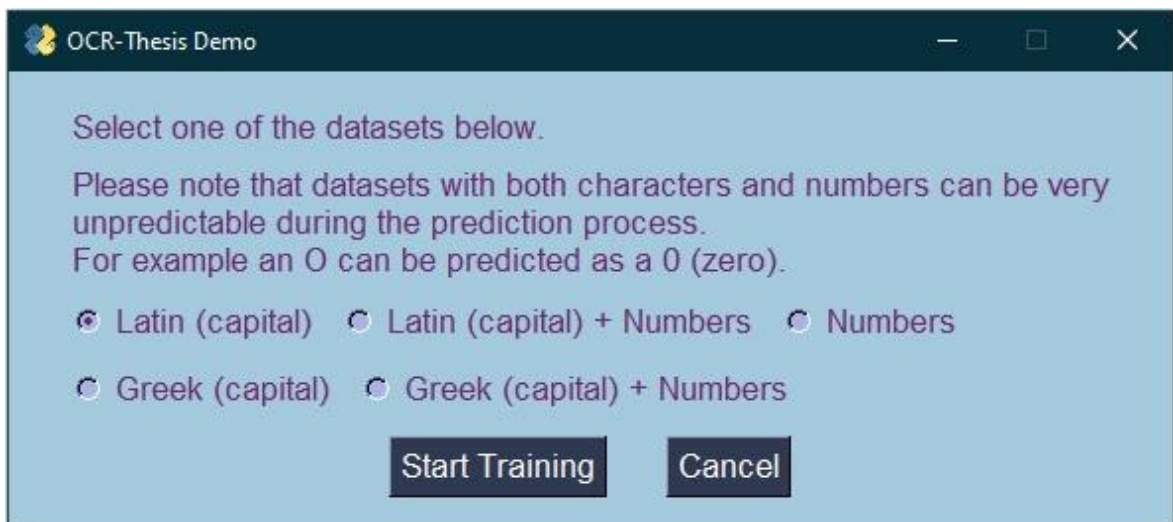
Με την συγκεκριμένη βιβλιοθήκη προστέθηκε η δυνατότητα φόρτωσης εικόνων και μοντέλων από τον αποθηκευτικό χώρο του υπολογιστικού συστήματος με έλεγχο σφαλμάτων, την δυνατότητα επιλογής δημιουργίας ενός συγκεκριμένου dataset μέσω των διαθέσιμων επιλογών, την εμφάνιση των αποτελεσμάτων κατά την διαδικασία της εκπαίδευσης του νευρωνικού μοντέλου, την εμφάνιση χρήσιμων μηνυμάτων προς τον χρήστη, και την εμφάνιση των τελικών αποτελεσμάτων πρόβλεψης του μοντέλου.

Ως παράδειγμα της γραφικής διεπαφής της εφαρμογής μπορούν να χαρακτηριστούν οι ακόλουθες εικόνες:





Εικόνα 20: Παράθυρο επιλογής διαδικασίας φόρτωσης ή δημιουργίας μοντέλου.



Εικόνα 21: Παράθυρο επιλογής dataset για δημιουργία και εκπαίδευση.

## 4.6 ΑΠΟΤΕΛΕΣΜΑΤΑ

Η διαδικασία της αναγνώρισης και κατά συνέπεια της εμφάνισης των αποτελεσμάτων ξεκινάει από την φόρτωση ενός μοντέλου από τον δίσκου του υπολογιστικού συστήματος. Εντός του φακέλου `Models`, του πηγαίου κώδικα της εφαρμογής, υπάρχουν ορισμένα προ-εκπαιδευμένα μοντέλα, καθώς επίσης και οποιαδήποτε μοντέλα δημιουργήσει ο χρήστης χρησιμοποιώντας την διαδικασία `Create a Model`. Κάθε μοντέλο περιέχει δύο αρχεία του ίδιου ονόματος, αλλά διαφορετικού τύπου. Περιλαμβάνεται το αρχείο `.h5` που περιέχει το εκπαιδευμένο μοντέλο, και το αρχείο `.npy` που περιέχει τα ονόματα των κατηγοριών στα οποία το μοντέλο

αντιστοιχεί την έξοδο του για την τελική εμφάνιση προς τον χρήστη. Αυτός είναι ένας τρόπος αντιστοίχισης της πρόβλεψης του μοντέλου, η οποία μπορεί να είναι ένας αριθμός, με ένα χαρακτήρα που αντιπροσωπεύει τον ίδιο χαρακτήρα που μόλις αναγνωρίστηκε.

```
def LoadModel(model_path, classes_path):  
    global model  
    global model_classes_from_file  
  
    #Loading model and storing the classes path.  
    model = load_model(model_path)  
    model_classes_from_file = classes_path
```

Πίνακας 13: Η συνάρτηση φόρτωσης μοντέλων.

Η φόρτωση του μοντέλου πραγματοποιείται με την απλή συνάρτηση `LoadModel` που λαμβάνει ως ορίσματα την τοποθεσία στην οποία βρίσκονται τα δύο προαναφερόμενα αρχεία, και στη συνέχεια φορτώνει σε μια μεταβλητή το μοντέλο και την τοποθεσία των κατηγοριών.

Στην συνέχεια ο χρήστης επιλέγει την εικόνα στην οποία επιθυμεί να πραγματοποιήσει την αναγνώριση, ορίζει τις περιοχές που τον ενδιαφέρουν εντός της εικόνας και πραγματοποιείται η διαδικασία που περιγράφηκε στο κεφάλαιο 4.4 – Επεξεργασία Εικόνας. Τα αποτελέσματα αυτής της διαδικασίας εισάγονται στην συνάρτηση `Predict`, όπου και πραγματοποιείται η πλήρης φόρτωση των κατηγοριών από το αρχείο `.npy`.

```
def Predict(images):  
    global model  
    global model_classes_from_file  
  
    output_label = ''  
  
    #Extracting the class names from the classes file.  
    classes = np.load(model_classes_from_file,  
                      allow_pickle=True)  
    classes_list = classes.tolist()  
    classes_names = []  
    for c in classes_list:  
        classes_names.append(c)  
  
    #Preparing the detected images for prediction.  
    processed_images = ImagePreProcessing(images)
```

```
#Predicting each character image and adding  
#the result to a string.  
for pi in processed_images:  
    prediction = model.predict(pi)  
    classification = prediction.argmax(axis = -1)  
    output_label = output_label +  
                    classes_names[int(classification)]  
  
return output_label
```

Πίνακας 14: Η συνάρτηση πρόβλεψης.

Τα ονόματα των κατηγοριών εισάγονται στην λίστα `classes_names`, ενώ οι εικόνες περνάνε από μια τελική προ-επεξεργασία εντός της συνάρτησης `ImagePreProcessing`, έτσι ώστε οι χαρακτήρες να λάβουν μια μορφή κατάλληλη προς αναγνώριση από την συνάρτηση `model.predict` του Keras. Κάθε χαρακτήρας εισάγεται μέσα στην συγκεκριμένη συνάρτηση ξεχωριστά και το αποτέλεσμα αντιστοιχείται με το κατάλληλο όνομα της κατηγορίας που βρίσκεται στην λίστα `classes_names`. Το τελικό αποτέλεσμα κάθε εικόνας εισάγεται στην μεταβλητή `output_label` και όταν ολοκληρωθεί η διαδικασία, τα αποτελέσματα επιστρέφουν στο αρχείο `run.py`.

Όταν η διαδικασία ολοκληρώνεται, εμφανίζεται ένα νέο παράθυρο στον χρήστη που περιλαμβάνει στο πάνω μέρος την περιοχή που επέλεξε ο χρήστης, καθώς επίσης και μια ένδειξη για το ποια εικόνα εμφανίζεται την συγκεκριμένη στιγμή. Από κάτω φαίνεται το αποτέλεσμα της πρόβλεψης του μοντέλου, και στο τέλος περιλαμβάνει 3 κουμπιά με τίτλο “Previous Result”, “Next Result” και “Finish” που εξυπηρετούν στην πλοήγηση του χρήστη στις άλλες επιλεγμένες περιοχές, εάν αυτές υπάρχουν.



Εικόνα 22: Παράθυρο αποτελεσμάτων.

## 4.7 ΒΕΛΤΙΩΣΕΙΣ

Ως τελευταίο κεφάλαιο της παρούσας πτυχιακής εργασίας θα ήθελα να αναφερθώ σε ορισμένες πιθανές βελτιώσεις που θα μπορούσαν να υλοποιηθούν στην εφαρμογή για την βελτίωση τόσο της ακρίβειας του μοντέλου, όσο και των δυνατοτήτων που παρέχει το ίδιο το πρόγραμμα.

Θεωρώ πως η σημαντικότερη βελτίωση θα ήταν η υλοποίηση ενός έξυπνου αλγορίθμου αυτόματης ανίχνευσης και διαχωρισμού χαρακτήρων. Ο συγκεκριμένος αλγόριθμος ενδεχομένως να μπορούσε να σαρώσει μια λέξη ή ένα κείμενο, να διαχωρίσει τους χαρακτήρες που το αποτελούν, να τους απομονώσει και στη συνέχεια να εισαχθούν εντός του μοντέλου για αναγνώριση. Η συγκεκριμένη προσέγγιση πιθανώς να αντικαθιστούσε την τρέχουσα χρήση των συναρτήσεων εντοπισμού των περιγραμμάτων από τους χαρακτήρες, καθώς θα επικεντρώνονταν στην γενική εμφάνιση των χαρακτήρων.

Μια άλλη βελτίωση θα μπορούσε να είναι η προσθήκη ενός διαδραστικού σχεδιασμού μοντέλων εντός της εφαρμογής δίχως την απαραίτητη συγγραφή κώδικα. Έχοντας κατά νου την γλώσσα προγραμματισμού Scratch, η ιδέα θα μπορούσε να υλοποιηθεί κατασκευάζοντας ορισμένα τμήματα κώδικα που θα αντιπροσωπεύουν τις στρώσεις του νευρωνικού δικτύου. Ο χρήστης θα μπορεί να τροποποιήσει ορισμένες παραμέτρους επάνω στα συγκεκριμένα τμήματα, να τα συνδυάσει, και κατά συνέπεια να δημιουργήσει ένα δικό του δίκτυο εντός της εφαρμογής. Ο χρήστης θα είχε έτσι την δυνατότητα να εξάγει τα δικά του αποτελέσματα χωρίς να χρειάζεται να επεξεργαστεί τον κώδικα της εφαρμογής για την τροποποίηση των ήδη υπαρχόντων μοντέλων.

Τέλος ορισμένες μικρότερες βελτιώσεις θα μπορούσαν να χαρακτηριστούν οι ακόλουθες:

- Βελτίωση του νευρωνικού δικτύου.
- Προσθήκη περισσότερων διαθέσιμων γλωσσών για αναγνώριση.
- Πρόβλεψη του ονόματος της γραμματοσειράς στην οποία ανήκει ένας χαρακτήρας (π.χ. Times New Roman, Arial, κλπ.)
- Σύνδεση της εφαρμογής με λεξικό για τον καλύτερο εντοπισμό λέξεων και διόρθωση λαθών.

## ΑΝΑΦΟΡΕΣ

Ask Any Difference, n.d. *Difference Between OMR and OCR (With Table) / Ask Any Difference*. [Ηλεκτρονικό]

Available at: <https://askanydifference.com/difference-between-omr-and-ocr/>

[Πρόσβαση 5 Ιουνίου 2021].

Brenner, M., 2018. *Artificial Neural Networks: What Every Marketer Should Know / Marketing Insider Group*. [Ηλεκτρονικό]

Available at: <https://marketinginsidergroup.com/content-marketing/artificial-neural-networks-every-marketer-know/>

[Πρόσβαση 28 Μαΐου 2021].

Bresnick, J., 2018. *What Is Deep Learning And How Will It Change Healthcare? / Health IT Analytics*. [Ηλεκτρονικό]

Available at: <https://healthitanalytics.com/features/what-is-deep-learning-and-how-will-it-change-healthcare>

[Πρόσβαση 28 Μαΐου 2021].

Bridgwater, A., 2018. *The 13 Types Of Data / Forbes*. [Ηλεκτρονικό]

Available at: <https://www.forbes.com/sites/adrianbridgwater/2018/07/05/the-13-types-of-data/?sh=10e795633624>

[Πρόσβαση 16 Απριλίου 2021].

Britton, S., 2019. *A Brief History Of OCR / CloudTrade*. [Ηλεκτρονικό]

Available at: <https://www.cloud-trade.com/blogs/2019/03/19/a-brief-history-of-ocr>

[Πρόσβαση 30 Μαΐου 2021].

Chen, J., 2020. *Neural Network / Investopedia*. [Ηλεκτρονικό]

Available at: <https://www.investopedia.com/terms/n/neuralnetwork.asp>

[Πρόσβαση 19 Μαΐου 2021].

Computer Hope, 2017. *Proportional Spacing / Computer Hope*. [Ηλεκτρονικό]

Available at: <https://www.computerhope.com/jargon/p/propspac.htm>

[Πρόσβαση 25 Αυγούστου 2021].

Computer Science Wiki, 2018. *Max-pooling / Pooling / Computer Science Wiki*.

[Ηλεκτρονικό]

Available at: [https://computersciencewiki.org/index.php/Max-pooling\\_/Pooling](https://computersciencewiki.org/index.php/Max-pooling_/Pooling)

[Πρόσβαση 28 Ιουνίου 2021].

'Data' in Lexico Dictionaries, n.d. *DATA / Definition of DATA by Oxford Dictionary / Lexico Dictionaries*. [Ηλεκτρονικό]

Available at: <https://www.lexico.com/definition/data>

[Πρόσβαση 15 Απριλίου 2021].

datareportal, 2021. *Digital Around The World / statcounter*. [Ηλεκτρονικό]

Available at: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>

[Πρόσβαση 17 Μαΐου 2021].

DeepAI a, n.d. *What is a Hidden Layer? / DeepAI*. [Ηλεκτρονικό]

Available at: <https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning>

[Πρόσβαση 19 Μαΐου 2021].

DeepAI b, n.d. *What is Weight (Artificial Neural Network)? / DeepAI*. [Ηλεκτρονικό]

Available at: <https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network>

[Πρόσβαση 24 Μαΐου 2021].

Diffen, n.d. *Data vs Information - Difference and Comparison / Diffen*. [Ηλεκτρονικό]

Available at: [https://www.diffen.com/difference/Data\\_vs\\_Information](https://www.diffen.com/difference/Data_vs_Information)

[Πρόσβαση 15 Απριλίου 2021].

Domo, 2020. *Data Never Sleeps 8.0 / Domo*. [Ηλεκτρονικό]

Available at: <https://www.domo.com/learn/data-never-sleeps-8>

[Πρόσβαση 10 Μαΐου 2021].

Dumoulin, V. & Visin, F., 2018. *A guide to convolution arithmetic for deep learning*.

[Ηλεκτρονικό]

Available at: <https://arxiv.org/abs/1603.07285>

[Πρόσβαση 28 Ιουνίου 2021].

FPT.AI, 2019. *Practical applications of OCR / FPT.AI*. [Ηλεκτρονικό]

Available at: <https://fpt.ai/practical-applications-ocr>

[Πρόσβαση 2 Ιουνίου 2021].

Giovinazzo, W. A., 2017. *Overfit / Underfit – Shaving with Occam’s Razor |*

*Meditations on BI and Data Science*. [Ηλεκτρονικό]

Available at: <https://meditationsonbianddatascience.com/2017/05/17/overfit-underfit-shaving-with-occams-razor/>

[Πρόσβαση 30 Ιουνίου 2021].

Great Learning Team, 2020. *Types of Neural Networks and Definition of Neural Network / Great Learning*. [Ηλεκτρονικό]

Available at: <https://www.mygreatlearning.com/blog/types-of-neural-networks/>

[Πρόσβαση 24 Μαΐου 2021].

Hardesty, L., 2017. *Explained: Neural Networks / MIT News Office*. [Ηλεκτρονικό]

Available at: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>

[Πρόσβαση 19 Μαΐου 2021].

History Computer, n.d. *GISMO of David Shepard / History Computer*. [Ηλεκτρονικό]

Available at: <https://history-computer.com/gismo-of-david-shepard/>

[Πρόσβαση 30 Μαΐου 2021].

How We Read, n.d. *Reading Machines / How We Read*. [Ηλεκτρονικό]

Available at: <http://www.howweread.co.uk/gallery/reading-machines/>

[Πρόσβαση 1 Ιουνίου 2021].

IBM Cloud Education, 2020. *Deep neural network / IBM / Neural Networks*.

[Ηλεκτρονικό]

Available at: <https://www.ibm.com/cloud/learn/neural-networks>

[Πρόσβαση 19 Μαΐου 2021].

Internet Live Stats, n.d. *Google Search Statistics / Internet Live Stats*. [Ηλεκτρονικό]

Available at: <https://www.internetlivestats.com/google-search-statistics/>

[Πρόσβαση 10 Μαΐου 2021].

Karandish, F., n.d. *The Comprehensive Guide to Optical Character Recognition (OCR)* / *Moov AI*. [Ηλεκτρονικό]

Available at: <https://moov.ai/en/blog/optical-character-recognition-ocr/>

[Πρόσβαση 9 Ιουνίου 2021].

Khurana, S., Raghavan, R. G. & Kakde, P. M., 2018. *Applications of OCR You Haven't Thought Of* / *Medium*. [Ηλεκτρονικό]

Available at: <https://medium.com/swlh/applications-of-ocr-you-havent-thought-of-69a6a559874b>

[Πρόσβαση 2 Ιουνίου 2021].

Liu, D., 2017. *A Practical Guide to ReLU* / *Medium*. [Ηλεκτρονικό]

Available at: <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>

[Πρόσβαση 20 Μαΐου 2021].

McLaughlin, E., 2020. *Data Collection* / *TechTarget*. [Ηλεκτρονικό]

Available at: <https://searchcio.techtarget.com/definition/data-collection>

[Πρόσβαση 11 Μαΐου 2021].

M., n.d. *Survey on Image Preprocessing Techniques to Improve OCR Accuracy* / *Medium*. [Ηλεκτρονικό]

Available at: <https://medium.com/technovators/survey-on-image-preprocessing-techniques-to-improve-ocr-accuracy-616ddb931b76>

[Πρόσβαση 9 Ιουνίου 2021].

Murzova, A. & Seth, S., 2020. *Otsu's Thresholding with OpenCV* | *LearnOpenCV*.

[Ηλεκτρονικό]

Available at: <https://learnopencv.com/otsu-thresholding-with-opencv/>

[Πρόσβαση 26 Αυγούστου 2021].

Rosebrock, A., 2021. *OpenCV Morphological Operations* / *pyimagesearch*.

[Ηλεκτρονικό]

Available at: <https://www.pyimagesearch.com/2021/04/28/opencv-morphological-operations/>

[Πρόσβαση 26 Αυγούστου 2021].



- Sayantini, D., 2020. *Keras vs TensorFlow vs PyTorch : Comparison of the Deep Learning Frameworks* / Edureka. [Ηλεκτρονικό]  
Available at: <https://www.edureka.co/blog/keras-vs-tensorflow-vs-pytorch/>  
[Πρόσβαση 27 Ιουνίου 2021].
- Seetharaman, K., 2018. *Financial Applications of Neural Networks* / Aspire Systems. [Ηλεκτρονικό]  
Available at: <https://blog.aspiresys.com/banking-and-finance/financial-applications-neural-networks>  
[Πρόσβαση 28 Μαΐου 2021].
- Sharma, A. V., 2017. *Understanding Activation Functions in Neural Networks* / Medium. [Ηλεκτρονικό]  
Available at: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>  
[Πρόσβαση 20 Μαΐου 2021].
- Shepard, D. H., 1953. *Apparatus For Reading*. US, Ευρεσιτεχνία Αρ. 2663758.
- Steppan, J., 2017. *Sample images from MNIST test dataset*. [Τέχνη].
- Talend, n.d. *What is Data Processing?* / Talend. [Ηλεκτρονικό]  
Available at: <https://www.talend.com/resources/what-is-data-processing/>  
[Πρόσβαση 11 Μαΐου 2021].
- Tanz, O. & Cambron, C., 2017. *Why the future of deep learning depends on finding good data* / TechCrunch. [Ηλεκτρονικό]  
Available at: <http://tcrn.ch/2vJhqp3>  
[Πρόσβαση 23 Μαΐου 2021].
- Taylor, K., n.d. *OCR vs. ICR: What Differentiates the two Character Recognition Software?* / HiTechNectar. [Ηλεκτρονικό]  
Available at: <https://www.hitechnectar.com/blogs/ocr-vs-icr/>  
[Πρόσβαση 7 Ιουνίου 2021].
- Thevenot, A., 2020. *Conv2d: Finally Understood What Happens in the Forward Pass / towards data science*. [Ηλεκτρονικό]

Available at: <https://towardsdatascience.com/conv2d-to-finally-understand-what-happens-in-the-forward-pass-1bbaafb0b148>

[Πρόσβαση 8 Σεπτεμβρίου 2021].

Veen, F. v. & Leijnen, S., 2019. *The Neural Network Zoo* / *The Asimov Institute*.

[Ηλεκτρονικό]

Available at: <https://www.asimovinstitute.org/neural-network-zoo/>

[Πρόσβαση 24 Μαΐου 2021].

Williams, L., 2019. *How Businesses Use Information* / *lumen*. [Ηλεκτρονικό]

Available at: <https://courses.lumenlearning.com/wmopen-introductiontobusiness/chapter/how-businesses-use-information/>

[Πρόσβαση 12 Μαΐου 2021].

Wood, T., n.d. *What is an Activation Function?* / *DeepAI*. [Ηλεκτρονικό]

Available at: <https://deepai.org/machine-learning-glossary-and-terms/activation-function>

[Πρόσβαση 20 Μαΐου 2021].