ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Υλοποίηση Ταξινομητή Εικόνων 10 Τάξεων με Εφαρμογή στην Παράκαμψη Ελέγχου CAPTCHA.** **Implementation of a ten class image classifier with application in CAPTCHA avoidance.**
Ονοματεπώνυμο Φοιτητή	**Κωνσταντίνος Σταυράκης**
Πατρώνυμο	**Κυριάκος**
Αριθμός Μητρώου	**ΜΠΣΠ/16031**
Επιβλέπων	**Γεώργιος Τσιχριντζής, Καθηγητής**

Ημερομηνία Παράδοσης **Σεπτέμβριος 2022**

Τριμελής Εξεταστική Επιτροπή

Γεώργιος Τσιχριντζής Καθηγητής	Ευάγγελος Σακκόπουλος Αναπληρωτής Καθηγητής	Διονύσιος Σωτηρόπουλος Επίκουρος Καθηγητής

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της μεταπτυχιακής διατριβής μου Καθηγητή. κ. Γεώργιο Τσιχριντζή για τις πολύτιμες συμβουλές και την καθοδήγηση που μου παρείχε.

Θερμές ευχαριστίες οφείλω, επίσης, στον Αναπληρωτή Καθηγητή κ. Ευάγγελο Σακκόπουλο και στον Επίκουρο Καθηγητή κ. Διονύσιο Σωτηρόπουλο, για την αμέριστη υποστήριξη και για τις πολύτιμες συμβουλές τους. Η συνδρομή τους ήταν καθοριστικής σημασίας για την εκπόνηση της διατριβής.

Τέλος, ευχαριστώ θερμά την οικογένεια μου για την έμπρακτη συμπαράσταση τους καθ' όλη την διάρκεια εκπόνησης της διατριβής.

Οκτώβριος 2022

Κωνσταντίνος Σταυράκης

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή έχει ως θέμα την Ανάπτυξη Ταξινομητή Εικόνων 10 Τάξεων, βάσει των οποίων καθίσταται δυνατή η πρόβλεψη επιθέσεων. Δηλαδή θα επιχειρείται η εφαρμογή τεχνικών Εξόρυξης Γνώσης (data mining) στην Ασφάλεια με σκοπό την πρόβλεψη επιθέσεων.

Αυτός ο στόχος επιτυγχάνεται με την υλοποίηση ενός ανιχνευτή δικτυακών εισβολών. Συγκεκριμένα, ενός μοντέλου πρόβλεψης ικανού να διακρίνει τις ‘κακές’ συνδέσεις, οι οποίες ονομάζονται εισβολές ή επιθέσεις, από τις ‘καλές’ φυσιολογικές συνδέσεις.

Abstract

The subject of this master's thesis is the Development of a 10-Class Image Classifier, based on which it becomes possible to predict attacks. Therefore, the application of Knowledge Mining techniques (data mining) in Security will be attempted in order to predict attacks.

This goal is achieved by implementing a network intrusion detector. Specifically, a prediction model capable of distinguishing 'bad' connections, which are called intrusions or attacks, from 'good' normal connections.

Keywords

Ασφάλεια, Αποθήκες Δεδομένων, Εξόρυξη Γνώσης.

Περιεχόμενα

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Δεν βρέθηκαν καταχωρήσεις πίνακα εικόνων.

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Δεν βρέθηκαν καταχωρήσεις πίνακα εικόνων.

Κεφάλαιο 1ο

1 Εισαγωγή

Ο στόχος της παρούσης διατριβής είναι η υλοποίηση ενός υλοποίηση ταξινομητή εικόνων 10 τάξεων με εφαρμογή στην παράκαμψη ελέγχου CAPTCHA.

1.1 Περιγραφή του υπό μελέτη προβλήματος

Με τη μεγάλη αύξηση της χρήσης .

1.2 Σκοπός και στόχοι της εργασίας

Οι στόχοι της εργασίας περιλαμβάνουν τα ακόλουθα:

1. Η ανάλυση εννοιών επί θεμάτων Ασφάλειας, Αποθήκες Δεδομένων και Εξόρυξης Γνώσης .

2. Η δημιουργία ενός ανιχνευτή δικτυακών εισβολών. Συγκεκριμένα, ενός μοντέλου πρόβλεψης ικανού να διακρίνει τις ‘κακές’ συνδέσεις, οι οποίες ονομάζονται εισβολές ή επιθέσεις, από τις ‘καλές’ φυσιολογικές συνδέσεις.

1.3 Παραδοτέα της εργασίας

Το τελικό παραδοτέο αποτελείται από ένα φάκελο που περιέχει τα εξής:

1. Το έντυπο κείμενο της εργασίας, το οποίο περιλαμβάνει αναλυτική περιγραφή των προσεγγίσεων που ακολουθήθηκαν σε κάθε μια από τις εργασίες συνοδευόμενα από τα κατάλληλα σχήματα (ΒΔ, ΑΔ κτλ), screenshots κτλ, επισκόπηση της σχετικής με το χώρο βιβλιογραφίας, τα αποτελέσματα της διατριβής.

2. Συνημμένα αρχεία που περιέχουν τα εξής: τα.

3. Η συλλογή πηγών και σχετικής βιβλιογραφίας για τη δημιουργία μίας βάσης γνώσης σχετικά με το θέμα.

1.4 Δομή Μεταπτυχιακής Διατριβής

Η Παρούσα διατριβή αποτελείται από 2 μέρη. Το πρώτο μέρος (κεφάλαια 2-5) αφορά το θεωρητικό τμήμα της διατριβής και περιγράφονται οι σχετικές έννοιες που χρησιμοποιούνται. Το δεύτερο μέρος (κεφάλαιο 6) αναλύεται η υλοποίηση του

Μέρος Ι: Θεωρητικό υπόβαθρο

1. Εισαγωγή

Η ραγδαια προοδος στην τεχνολογια του υλικου τα τελευταια χρονια εχει οδηγησει μεταξυ αλλων στην αναπτυξη και την εκτεταμενη διαδοση συσκευων ληψης πολυμεσικου περιεχομενου (καμερες, κινητα τηλεφωνα, PDAs, κτλ.) με υψηλες δυνατοτητες αποθηκευσης δεδομενων, ενω εχει επισης συμβαλει στην ευρεια και συνεχως αυξανομενη διαθεσιμοτητα της προσβασης στο διαδικτυο (internet). Οι παραγοντες αυτοι ειχαν ως αποτελεσμα τη δημιουργια τεραστιων βασεων πολυμεσικου περιεχομενου (multimedia content), οι οποιες αποτελουν αντικειμενο συνδιαλλαγης αναμεσα στους χρηστες η κατασκευαστηκαν με σκοπο να γινουν διαθεσιμες στο διαδικτυο. Προκειμενου να γινει καλυτερα αντιληπτο το πραγματικο μεγεθος των βασεων αυτων, παρατιθεται ενδεικτικα ο ογκος των δεδομενων που εχει παρατηρηθει προσφατα σε ορισμενους απο τους πιο δημοφιλεις ιστοτοπους (sites) του διαδικτυου που υποστηριζουν την χρηση και το διαμοιρασμο πολυμεσικου περιεχομενου:

i) Flickr /

• Περιεχει πανω απο 5 δισεκατομμυρια εικονες (Σεπτεμβριος 2010)

• Αποθηκευονται περισσοτερες απο 3000 νεες εικονες ανα λεπτο ii) YouTube:

• Παρακολουθουνται περισσοτερα απο 2 δισεκατομμυρια εικονοσειρες (videos) ημερησιως

• Καθε λεπτο αποθηκευονται νεες εικονοσειρες συνολικης διαρκειας μεγαλυτερης απο 24 ωρες (Οκτωβριος 2010)

iii) facebook:

• Περιεχει πανω απο 10 δισεκατομμυρια εικονες (Οκτωβριος 2008)

• Καθε μερα αποθηκευονται περισσοτερα απο 2-3 Terabytes περιεχομενου εικονων

Παραλλήλως με την προαναφερθεισα αυξηση του ογκου και της διαθεσιμοτητας του πολυμεσικου υλικου (εικονες, εικονοσειρες), τυπικες διαδικασιες, οπως η δεικτοδοτηση (indexing), η αναζητηση (search) και η ανακτηση (retrieval) περιεχομενου σε τετοιες συλλογες, αποτελουν ολο και περισσοτερο αναποσπαστο κομματι των καθημερινων δραστηριοτητων των χρηστων τοσο σε προσωπικο οσο και σε επαγγελματικο επιπεδο. Κατα συνεπεια, εχουν προκυψει νεες αναγκες αναφορικα με την αναπτυξη προηγμενων και ευχρηστων συστηματων για τον αποτελεσματικο χειρισμο του πολυμεσικου περιεχομενου. Για το σκοπο αυτο, τα τελευταια χρονια εχουν επικεντρωθει εντονες ερευνητικες προσπαθειες στο σχεδιασμο και την αναπτυξη εξελιγμενων τεχνικων, οι οποιες

θα συμβαλουν καθοριστικα στην αποτελεσματικη εκτελεση των προαναφερθεντων διαδικασιων απο τους χρηστες.

Πιο προσφατα, εχει υιοθετηθει ευρεως η θεμελιωδης αρχη της στροφης των τεχνικων χειρισμου του οπτικου περιεχομενου προς ενα σημασιολογικο επιπεδο [12]. Η σημασιολογικη αναλυση του πολυμεσικου περιεχομενου αποτελει τον ακρογωνιαιο λιθο αυτης της προσπαθειας για ευφυη χειρισμο του περιεχομενου, η οποια επιχειρει να γεφυρωσει το αποκαλουμενο "σημασιολογικο κενο" (semantic gap) [56] μεταξυ των χαρακτηριστικων χαμηλου επιπεδου (π.χ. χαρακτηριστικα χρωματος, υφης, ηχου, κινησης) και των υψηλου επιπεδου σημασιολογικων εννοιων (semantic concepts). Οι τεχνικες αυτης της κατηγοριας στοχευουν στην αποκτηση και τη μοντελοποιηση της σημασιολογικης πληροφοριας που υπαρχει στο πολυμεσικο περιεχομενο και η εφαρμογη τους σε μια πλειαδα διαφορετικων εφαρμογων εχει παρουσιασει πολλα υποσχομενα αποτελεσματα.

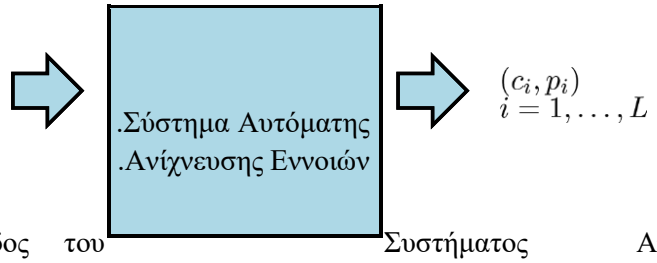
:

2. θεωρητικό υπόβαθρο: Μηχανική Μάθηση σε Πολυμεσικά Δεδομένα

Τα πολυμέσα είναι δεδομένα τα οποία αποτελούνται από συνδυασμό ενός ή περισσότερων τύπων περιεχομένου. Οι συνηθέστεροι τύποι περιεχομένου είναι η ακίνητη εικόνα (φυσική ή συνθετική), η κινούμενη εικόνα (εικονοσειρές, video, animation), ο ήχος (ομιλία, μουσική, ήχοι περιβάλλοντος κλπ) το κείμενο και τα τρισδιάστατα γραφικά (π.χ., σε διαδραστικό περιβάλλον). Συνηθισμένα παραδείγματα
πολυμέσων είναι

- Εικόνες (ένας τύπος δεδομένων).
- Εικονογραφημένο κείμενο (δύο τύποι δεδομένων, π.χ., σελίδες html).
- Video (συνδυασμός εικόνας, ήχου και κειμένου).

Μία *έννοια* (concept) είναι μία σαφώς ορισμένη σημασιολογική οντότητα που μπορεί να χρησιμοποιηθεί για να περιγράψει το περιεχόμενο των πολυμέσων και να επιτρέψει με αυτόν τον τρόπο την αποτελεσματική δεικτοδότηση και ανάκτησή τους. Στόχος της ερευνητικής περιοχής που εξετάζουμε είναι η υλοποίηση ενός *Συστήματος Αυτόματης Ανίχνευσης Εννοιών* (ΣΑΑΕ) το οποίο ανιχνεύει τις έννοιες στα πολυμεσικά έγγραφα με βάση την πρωτογενή τους πληροφορία.
Η είσοδος του συστήματος είναι μία *πολυμεσική μονάδα σχολιασμού* (multimedia annotation unit), η απλά πολυμεσική μονάδα. Η πολυμεσική μονάδα είναι το τμήμα εκείνο ενός πολυμεσικού εγγράφου στο οποίο αντιστοιχίζονται οι έννοιες. Το μέγεθος των πολυμεσικών μονάδων καθορίζει την λεπτομέρεια του σχολιασμού. Στην περίπτωση του video, για παράδειγμα, μία πολυμεσική μονάδα μπορεί να είναι ολόκληρη η ροή του video, μία σκηνή του video ή ένα καρέ. Αν η έννοια αντιστοιχιστεί σε ολόκληρο το video μειώνεται η διακριτική ικανότητα, καθώς δεν γίνεται γνωστό σε ποιο ή ποια σημεία του video υπάρχει η έννοια. Από την άλλη, εάν η πολυμεσική μονάδα είναι το καρέ, τότε αυξάνεται σημαντικά το υπολογιστικό κόστος και η περίσσεια πληροφορίας. Έτσι, συνήθως για τα video η πολυμεσική μονάδα είναι μία σκηνή[1], ενώ για τις εικόνες η πολυμεσική μονάδα είναι συνήθως ολόκληρη η εικόνα. .



Σχήμα 1: Είσοδος και έξοδος του Συστήματος Αυτόματης Ανίχνευσης Εννοιών. Στην έξοδο επιστρέφεται μία πιθανότητα p_i για την ύπαρξη της κάθε έννοιας c_i.
Η πολυμεσική μονάδα εισόδου μπορεί να συνοδεύεται από *μεταδεδομένα* (metadata). Τα μεταδεδομένα είναι πληροφορία που είναι συμπληρωματική προς το περιεχόμενο και μπορεί να είναι τεχνική (π.χ., πληροφορίες για την λήψη μιας εικόνας που εισάγονται από την φωτογραφική μηχανή, όπως η ώρα λήψης, διάφραγμα του φακού κ.α.) ή να προσφέρει πληροφορίες για την πολυμεσική μονάδα (π.χ., επικεφαλίδες εικόνων, ή η απομαγνητοφωνημένη ομιλία στην περίπτωση του video). Τα

μεταδεδομένα μπορεί να είναι ελεύθερο κείμενο ή να είναι δομημένα σύμφωνα με κάποιο πρότυπο (όπως τα IPTC[3] και MPEG-7[4, 5, 6]).

Η έξοδος του συστήματος είναι ένα σύνολο ζευγών (c_i,p_i), $i = 1,...,L$ όπου c_i η i-οστή έννοια και p_i η εκτιμώμενη πιθανότητα εμφάνισης της c_i στην πολυμεσική μονάδα εισόδου. Αξίζει να σημειωθεί πως για του σκοπούς της ανάκτησης ενδιαφέρει κυρίως η κατάταξη που προκύπτει με βάση τις εκτιμώμενες πιθανότητες p_i και όχι η απόλυτη τιμή αυτών των πιθανοτήτων, όπως συμβαίνει στο πρόβλημα της ταξινόμησης (classification). Η διαφορά έγκειται στο ότι στην ταξινόμηση απαιτείται μία αυστηρή απόφαση (υπάρχει/δεν υπάρχει) για την έννοια, ενώ στην ανάκτηση, ενδιαφέρει περισσότερο επιστραφούν πρώτα οι πολυμεσικές μονάδες που είναι πιθανότερο να περιέχουν την έννοια. Στην παρούσα εργασία αν δεν δηλώνεται ρητά η χρήση του ταξινομητή, υπονοείται πως χρησιμοποιείται για τον υπολογισμό των πιθανοτήτων p_i και όχι για την λήψη μίας απόφασης για την ύπαρξη ή μη ύπαρξη της έννοιας στην πολυμεσική μονάδα. Το Σχήμα 1.2 συνοψίζει την είσοδο και την έξοδο του ΣΑΑΕ. Στα επόμενα, όπου αναφέρεται η πολυμεσική μονάδα εισόδου, θα εννοείται πως συμπεριλαμβάνονται και τα μεταδεδομένα.

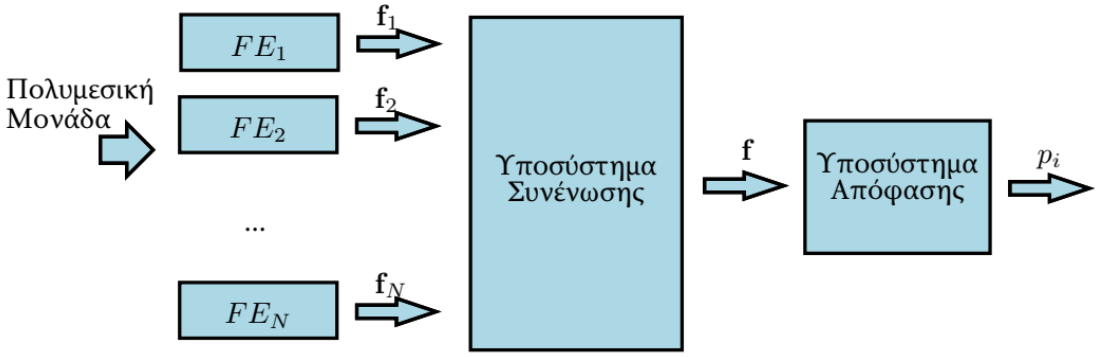
Με βάση αυτούς τους ορισμούς, μπορούμε πλέον να διατυπώσουμε το πρόβλημα της ερευνητικής περιοχής στην οποία εντάσσεται η παρούσα διατριβή:

Δεδομένης μία πολυμεσικής μονάδας καθώς και μιας προκαθορισμένης λίστας εννοιών c_i, $i = 1,...,L$, στόχος είναι ο υπολογισμός των πιθανοτήτων p_i για την ύπαρξη των εννοιών στην πολυμεσική μονάδα.

Την διαδικασία υπολογισμού των πιθανοτήτων p_i την ονομάζουμε *σημασιολογική ανάλυση* των πολυμεσικών δεδομένων, καθώς εξάγεται πληροφορία για έννοιες που γίνονται αντιληπτές από τον άνθρωπο, με βάση το περιεχόμενο χαμηλού επιπέδου. Για την επίλυση του προβλήματος έχουν προταθεί μέθοδοι που βασίζονται

στην μοντελοποίηση της γνώσης (όπως οι μέθοδοι που προτείνονται στα [7] και [8], και αφορούν συγκεκριμένα πεδία εφαρμογής), ωστόσο η πλειοψηφία των μεθόδων της βιβλιογραφίας βασίζονται σε μεθόδους *μηχανικής εκμάθησης* (machine learning): Από τα δεδομένα εισόδου εξάγονται συνοπτικές διανυσματικές περιγραφές των δεδομένων οι οποίες διατηρούν την ωφέλιμη για την ανίχνευση πληροφορία και ονομάζονται *χαρακτηριστικά χαμηλού επιπέδου*. Με βάση τα χαρακτηριστικά χαμηλού επιπέδου εκπαιδεύονται ταξινομητές που χρησιμοποιούνται για τον υπολογισμό των πιθανοτήτων p_i. Σημειώνεται πως το πρόβλημα ταξινόμησης για κάθε έννοια είναι *δυαδικό*, δηλαδή ο ταξινομητής προσπαθεί να διακρίνει μία έννοια c, από το συμπλήρωμά της $\bar{c}$. Έτσι το πρόβλημα της σημασιολογικής ανάλυσης πολυμέσων προσεγγίζεται σαν ένα πρόβλημα αναγνώρισης προτύπων.

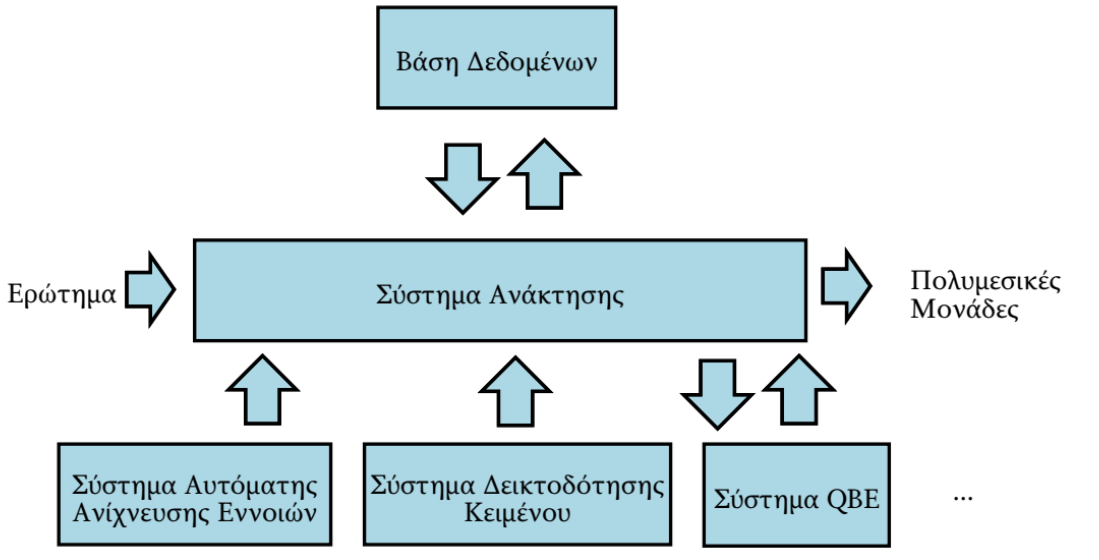
Η γενική αρχιτεκτονική ενός τυπικού ΣΑΑΕ δίνεται στο Σχήμα 1.3. Αρχικά εξάγονται N διανύσματα χαρακτηριστικών (feature extraction) από τους διάφορους τύπους δεδομένων της πολυμεσικής μονάδας εισόδου. Στο υποσύστημα συνένωσης, συνδυάζονται τα διαφορετικά χαρακτηριστικά χαμηλού επιπέδου σε μία ενιαία αναπαράσταση, ενώ στο υποσύστημα απόφασης υπολογίζονται οι πιθανότητες εξόδου.



Σχήμα 1.3: Η γενική αρχιτεκτονική του Συστήματος Αυτόματης Ανίχνευσης Εννοιών σε πολυμέσα. Τα κύρια στάδια της ανίχνευσης είναι η εξαγωγή χαρακτηριστικών (feature extraction), η συνένωση και η απόφαση.

1.2.1 Η λειτουργία του ΣΑΑΕ στην δεικτοδότηση και ανάκτηση πολυμέσων

Για την ανάκτηση πολυμέσων, οι πιθανότητες προϋπολογίζονται από το ΣΑΑΕ και αποθηκεύονται για κάθε πολυμεσική μονάδα. Το Σύστημα Ανάκτησης (Retrieval System), αξιοποιεί αυτή την πληροφορία σύμφωνα με κάποιο Μοντέλο Ανάκτησης (Retrieval Model) το οποίο είναι ένα θεωρητικό μοντέλο που προβλέπει την συνάφεια του κάθε εγγράφου με το ερώτημα του χρήστη [9]. Συνδυάζοντας την έξοδο του ΣΑΑΕ με άλλες πηγές πληροφορίας, όπως το κείμενο, το Σύστημα Ανάκτησης



Σχήμα 1.4: Η γενικότερη δομή ενός συστήματος ανάκτησης πολυμέσων που υποστηρίζει πολλούς τύπους ερωτημάτων. Το ΣΑΑΕ εμπλέκεται κατά την φάση της δεικτοδότησης των πολυμεσικών μονάδων. Οι υπολογισμένες πιθανότητες αξιοποιούνται από το Σύστημα Ανάκτησης κατά την φάση της ανάκτησης.

επιστρέφει πολυμεσικές μονάδες συναφείς με τα ερωτήματα των χρηστών κατά το στάδιο της ανάκτησης, όπως φαίνεται στο Σχήμα 1.4.

Ως παράδειγμα χρήσης του ΣΑΑΕ, δίνεται στο Σχήμα 1.5 η διεπαφή του συστήματος VITALAS [10], το οποίο χρησιμοποιεί ένα ΣΑΑΕ που βασίζεται στην παρούσα εργασία. Ο χρήστης μπορεί να εισάγει

παραδοσιακά ερωτήματα κειμένου, να θέσει ερωτήματα με παραδείγματα, ή να αναζητήσει έννοιες που έχουν εξαχθεί με βάση το περιεχόμενο. Στην περίπτωση αυτή το ερώτημα με βάση τις έννοιες είναι άμεσο. Γενικά ωστόσο, μπορεί το ερώτημα να είναι έμμεσο, δηλαδή το ερώτημα να τίθεται σε ελεύθερο κείμενο και οι όροι που αντιστοιχούν σε έννοιες να ανακτώνται με βάση τις πιθανότητες που έχουν υπολογιστεί από το ΣΑΑΕ.

2.1 Χαρακτηριστικά χαμηλού επιπέδου

Το πρώτο στάδιο της επεξεργασίας της πολυμεσικής μονάδας εισόδου είναι η εξαγωγή των χαρακτηριστικών χαμηλού επιπέδου. Αυτά στην περίπτωση που εξετάζουμε αποτελούν απεικονίσεις των αρχικών δεδομένων σε κάποιον διανυσματικό χώρο με περιορισμένο αριθμό διαστάσεων. Έτσι τα αρχικά δεδομένα που αποτελούνται από εκατομμύρια μεταβλητές (εικονοστοιχεία, δείγματα ήχου κλπ) περιγράφονται με κάποιες εκατοντάδες ή λίγες χιλιάδες στοιχεία.

Κύριος στόχος είναι η μείωση του αριθμού των μεταβλητών διατηρώντας ταυτόχρονα την πληροφορία που είναι ωφέλιμη για την εξαγωγή εννοιών. Με αυτόν τον τρόπο περιορίζεται το υπολογιστικό κόστος ενώ αποφεύγονται τα προβλήματα που δημιουργεί στους ταξινομητές ο μεγάλος αριθμός διαστάσεων (“κατάρα της διάστασης”, [11, 12]). Δεδομένου ότι τα πολυμέσα αποτελούνται από πολλαπλούς τύπους σημάτων, τα χαρακτηριστικά χαμηλού επιπέδου μπορούν να κατηγοριοποιηθούν σε “Οπτικά”, “Ήχου” και “Κειμένου”.

2.1.3Χαρακτηριστικά κειμένου

Η χρήση χαρακτηριστικών κειμένου έχει μελετηθεί εκτενώς από την ερευνητική περιοχή της κατηγοριοποίησης κειμένου. Μία σημαντική παρατήρηση που μπορεί να γίνει για την περίπτωση του κειμένου είναι πως η έρευνα εστιάζει περισσότερο σε τρόπους μείωσης της διάστασης των χαρακτηριστικών (dimensionality reduction) καθώς και στην επιλογή των πιο ωφέλιμων από αυτά (feature selection). Σε ότι αφορά τα ίδια τα χαρακτηριστικά, πολύ συχνά χρησιμοποιείται το μοντέλο του “Σάκου Λέξεων” (bag-of-words, BoW).

Σύμφωνα με το μοντέλο BoW, το κείμενο δεν αντιμετωπίζεται σαν μία ακολουθία από λέξεις με συντακτική δομή, αλλά σαν ένα μη διατεταγμένο σύνολο λέξεων. Η κάθε λέξη έτσι αντιστοιχεί σε μία διάσταση. Η τιμή της διάστασης μπορεί να είναι ο απόλυτος αριθμός των εμφανίσεων της λέξης, η συχνότητα εμφάνισης της λέξης, ή η μέτρηση tf-idf. Για έναν όρο ti ο οποίος εμφανίζεται ni;j φορές σε ένα έγγραφο dj μιας συλλογής D, έχουμε

$$tf\text{-}idf_{i,j} = tf_{i,j} \times idf_i$$

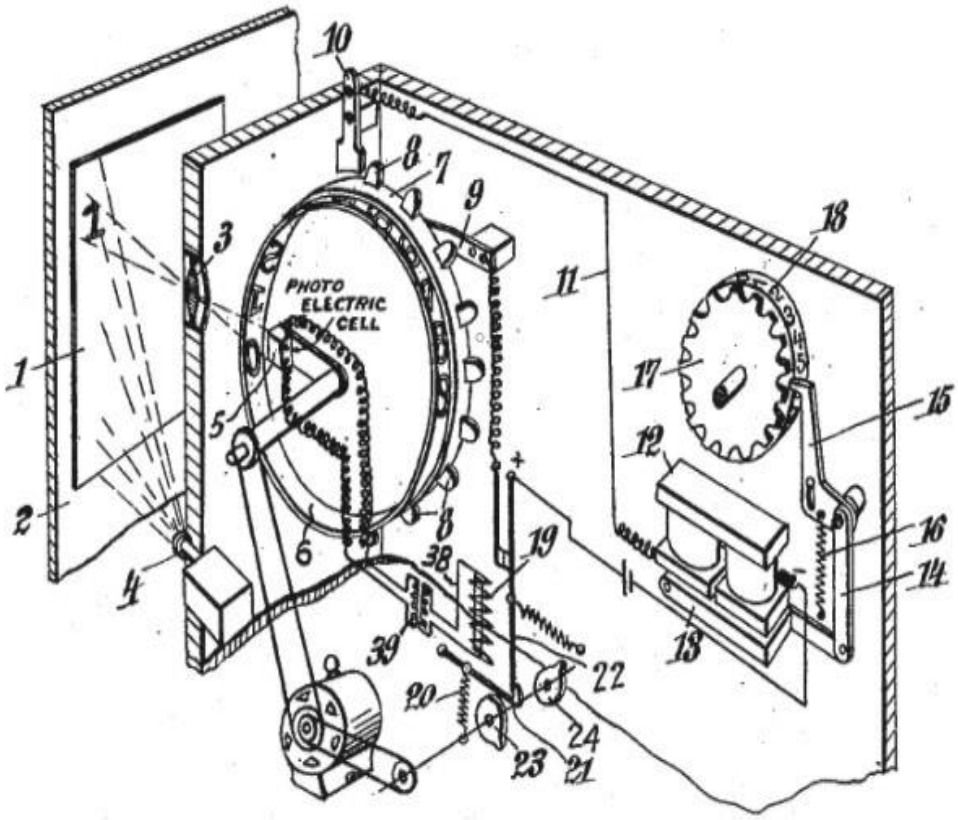
$$idf_i = \log \frac{|\{d : t_i \in d\}|}{|D|}$$

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

3. Οπτική Αναγνώριση Χαρακτήρων και Αναγνώριση Χειρόγραφων Χαρακτήρων

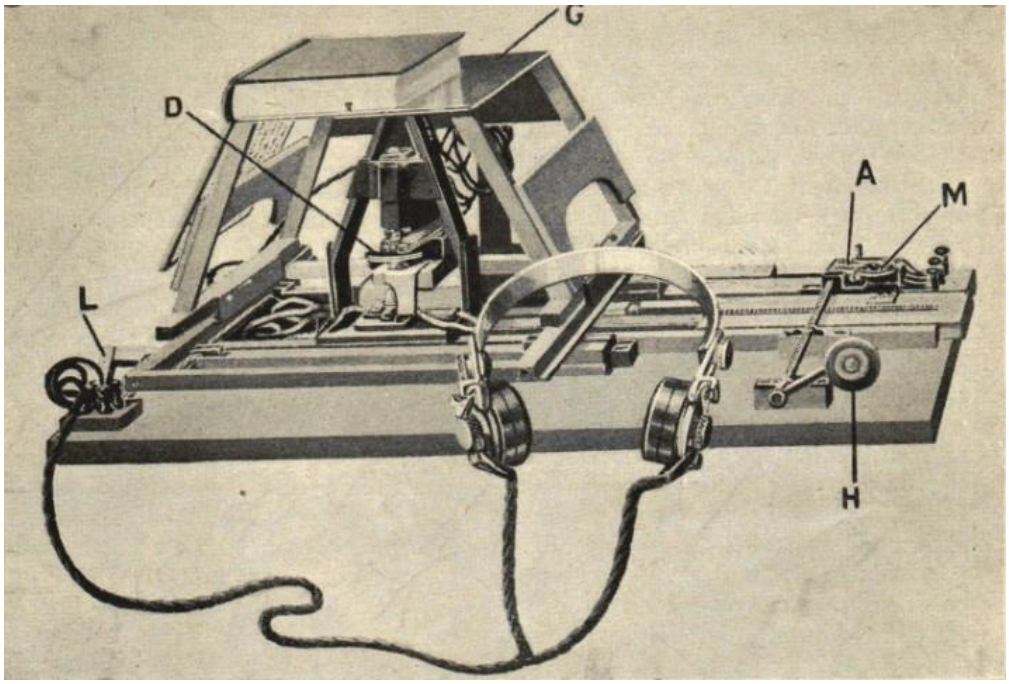
2.1 Ιστορική Αναδρομή

Η ιστορία της αναγνώρισης χειρόγραφων (HWR) δεν είναι πλήρης χωρίς να αναφέρουμε τα συστήματα της οπτικής αναγνώρισης χαρακτήρων (OCR) που προηγήθηκαν. Πολλοί άνθρωποι ονειρεύτηκαν μια μηχανή που θα μπορούσε να αναγνωρίζει χαρακτήρες και αριθμούς αλλά φαίνεται ότι η πρώτη μηχανή OCR αναπτύχθηκε από τον Αυστριακό μηχανικό Gustav Tauschek (1899-1945) το 1929 όπου απέκτησε δίπλωμα ευρεσιτεχνίας πάνω στην OCR, με το όνομα Reading Machine, στην Γερμανία. Το 1935, απέκτησε δίπλωμα ευρεσιτεχνίας για την μηχανή του και στις Ηνωμένες Πολιτείες της Αμερικής. Η μηχανή ανάγνωσης του Tauschek ήταν μια μηχανική συσκευή η οποία χρησιμοποιούσε πρότυπα (ταίριασμα προτύπων) και φωτοανιχνευτή (αισθητήρα φωτός) [4].



Εικόνα 1: Μηχανή Ανάγνωσης του Tauschek

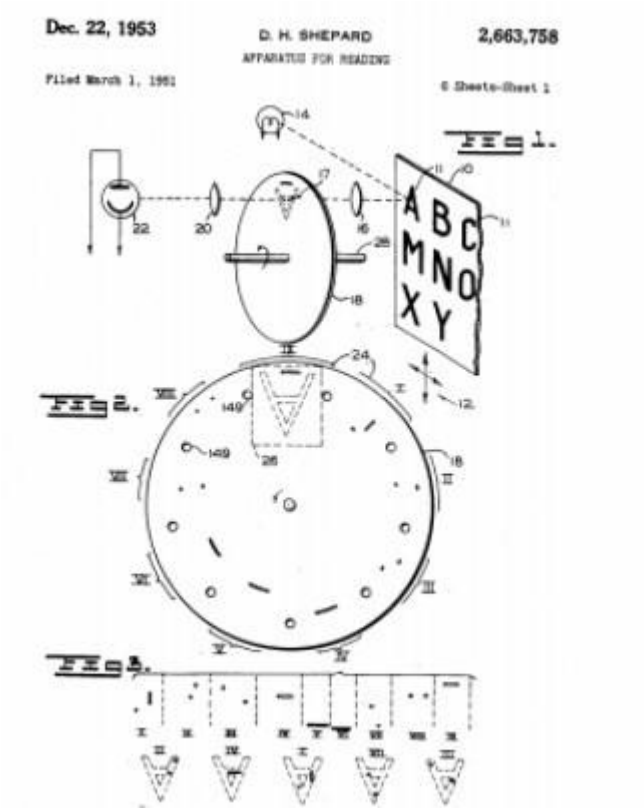
Αρκετοί, αναφέρουν ως πρώτο σύστημα OCR, το Optophone, έναν φορητό σαρωτή εικόνας που αναπτύχθηκε από τον Edmund Fourier d' Albe το 1914, ο οποίος όταν κινούνταν κατά μήκος μίας εκτυπωμένης γραμμής κειμένου, παρήγαγε ηχητικούς τόνους ανάλογα με το γράμμα που συναντούσε.



Εικόνα 2: Ο φορητός σαρωτής Optophone

Ερευνητές της Radio Corporation of America (RCA) δημιούργησαν το 1949, μια πρώιμη μορφή οπτικής αναγνώρισης χαρακτήρων για λογαριασμό της Γενικής Γραμματείας Απόστρατων των ΗΠΑ, το οποίο όμως να μετατρέπει εκτυπωμένους χαρακτήρες σε χαρακτήρες αναγνώσιμους από υπολογιστή, είχε ως στόχο να εκφωνεί τα γράμματα. Η συσκευή είχε υψηλό κόστος και δεν βγήκε στην παραγωγή.

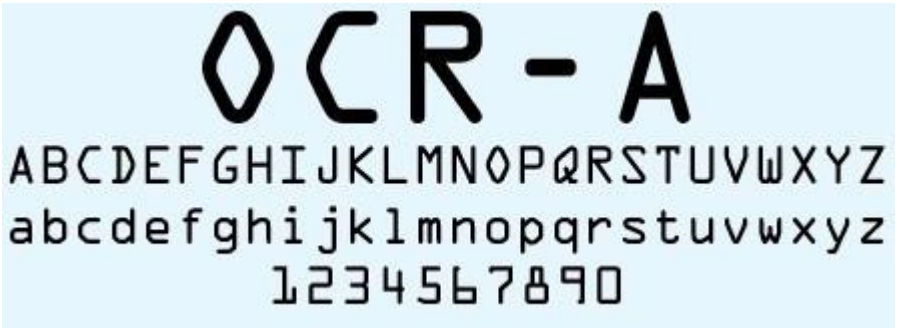
Το 1951, ένας κορυφαίος κρυπταναλυτής της Υπηρεαίας Ασφάλειας των Ενόπλων Δυνάμεων των ΗΠΑ, ονόματι David Hammond Shepard, δημιούργησε το πρώτο σύστημα αναγνώρισης χαρακτήρων, το οποίο και ονόμασε Gismo. Το Gismo ήταν μία μηχανή για την μετατροπή έντυπων εγγράφων σε γλώσσα μηχανής για επεξεργασία μέσω υπολογιστή. Κατοχυρώθηκε με δίπλωμα ευρεσιτεχνίας το 1953. Στην αρχή μπορούσε να αναγνωρίσει 23 από τα 26 γράμματα του λατινικού αλφαβήτου, αλλά μετά από έναν χρόνο δουλειάς, ανέπτυξαν την μηχανή ώστε να μπορεί να αναγνωρίζει και τους 26 χαρακτήρες του λατινικού αλφάβητου. Λίγα χρόνια αργότερα ο Shepard ίδρυσε την εταιρεία, Intelligent Machines Research Co. (IMR) η οποία ανέπτυξε τα πρώτα συστήματα οπτικής αναγνώρισης χαρακτήρων για εμπορική χρήση.



Εικόνα 3: Η μηχανή Gismo

Το 1965, το Reader's Digest, ένα αμερικανικό οικογενειακό περιοδικό γενικού ενδιαφέροντος και η RCA συνεργάστηκαν για να φτιάξουν μια συσκευή OCR για να διαβάζει και να ψηφιοποιεί τους σειριακούς αριθμούς από τα κουπόνια του Reader's Digest από τις διαφημίσεις. Η γραμματοσειρά που χρησιμοποιήθηκε για την εκτύπωση των κουπονιών ήταν η OCR-A font, μια γραμματοσειρά ειδικά σχεδιασμένη για τους σκοπούς της οπτικής αναγνώρισης, τα γράμματα της οποίας είχαν εκτυπωθεί από έναν εκτυπωτή τυμπάνου (drum printer) της ίδιας εταιρείας. Η συσκευή ήταν απευθείας συνδεδεμένη με έναν RCA 301 υπολογιστή, έναν από τους πρώτους ηλεκτρονικούς υπολογιστές τεχνολογίας ημιαγωγών. Είχε επίσης έναν ειδικό αναγνωστέα TWA. Η συσκευή μπορούσε να επεξεργαστεί 1500 έγγραφα ανά λεπτό, απορρίπτοντας ότι δεν μπορούσε να αναγνωρίσει ορθά. Το προϊόν αυτό μπήκε τελικά στην κανονική γραμμή παραγωγής όπου και προωθήθηκε εμπορικά για εργασίες αντίστοιχου τύπου.

Εικόνα 4: Ο υπολογιστής RCA 301



Εικόνα 5: Η ειδικά σχεδιασμένη γραμματοσειρά OCR-A

Το ταχυδρομείο των Ηνωμένων Πολιτειών της Αμερικής χρησιμοποιεί τεχνολογία οπτικής αναγνώρισης από το 1965 βασισμένο σε τεχνολογία που ανέπτυξε ο εφευρέτης Jacob Rainbow. Το ταχυδρομείο της Αγγλίας έκανε την πρώτη χρήση οπτικής αναγνώρισης στην Ευρώπη. Μια διαδικασία που έφερε επανάσταση στα συστήματα πληρωμής λογαριασμών στην Μ. Βρετανία. Το 1971 το καναδικό ταχυδρομείο υιοθέτησε για πρώτη φορά τα συστήματα οπτικής αναγνώρισης. Το σύστημα διάβαζε το όνομα και την διεύθυνση του παραλήπτη και τύπωνε ένα bar code με οδηγίες δρομολόγησης ανάλογα με τον ταχυδρομικό κώδικα του προορισμού. Για να μην συγχέονται τα bar codes με άλλα σχέδια που μπορεί να υπήρχαν πάνω στον φάκελο, η εκτύπωση των bar codes γινόταν με ειδικό μελάνι χρώματος πορτοκαλί, το οποίο είχε υψηλά ανακλαστικά κάτω από υπεριώδες ακτίνες φωτός.

Το 1974, δημιουργήθηκε το πρώτο σύστημα οπτικής αναγνώρισης χαρακτήρων που αναγνώριζε εκτυπωμένο κείμενο διαφόρων γραμματοσειρών. Δημιουργός ήταν ο Ray Kurzweil, ιδρυτής της

εταιρείας Kurzweil Computer Products Inc. Η εταιρεία εστίασε την προσοχή της στην δημιουργία μιας συσκευής που θα βοήθαγε τυφλούς ανθρώπους να διαβάζουν με την βοήθεια υπολογιστή. Το 1978, η εταιρεία άρχισε να διαθέτει προς πώληση εταιρικές εκδόσεις του λογισμικού οπτικής αναγνώρισης που είχε αναπτύξει. Δύο χρόνια αργότερα, η εταιρεία πουλήθηκε στην Xerox, που έδειξε ενδιαφέρον για την τεχνολογία της οπτικής αναγνώρισης.

Το 2000, η τεχνολογία της οπτικής αναγνώρισης έγινε διαθέσιμη ως υπηρεσία στο διαδίκτυο (WebOCR), σε περιβάλλον υπολογιστικού νέφους (cloud), καθώς και σε εφαρμογές για κινητές συσκευές. Ενώ μόλις το 2013 δημιουργήθηκε η MNIST βάση δεδομένων (database) για να εκπαιδεύει μοντέλα μηχανικής μάθησης στην αναγνώριση προτύπων (pattern recognition).

2.2 Οι γενιές της OCR

Τα βιομηχανικά προσβάσιμα OCR μπορούν να χωριστούν σε τέσσερις γενιές με βάση την δύναμη, αποτελεσματικότητα και ικανότητα προσαρμογής τους. Η πρώτη γενιά της οπτικής αναγνώρισης μπορούσε να διαβάσει μόνο επιλεγμένα στυλ κειμένου και σχήματα χαρακτήρων. Η λογική αντιστοίχιση προτύπου (logical template matching) ήταν η βασική μέθοδος που χρησιμοποιούσαν. Η δεύτερη γενιά της οπτικής αναγνώρισης ήταν αρκετά πιο αποτελεσματική, καθώς τα συστήματα της εποχής μπορούσαν να αναγνωρίσουν εκτυπωμένους αλλά και χειρόγραφους χαρακτήρες. Η αναγνώριση των χειρόγραφων χαρακτήρων περιορίστηκε όμως μόνο σε αριθμούς και πολύ λίγα γράμματα και σύμβολα. Για την τρίτη γενιά η πρόκληση ήταν έγγραφα κακής ποιότητας μεγάλα τυπωμένα και χειρόγραφα σύνολα χαρακτήρων. Το χαμηλό κόστος και η υψηλή απόδοση ήταν επίσης σημαντικοί στόχοι, οι οποίοι επιτεύχθηκαν από την δραματική πρόοδο στην τεχνολογία υλικού [5]. Πλέον στην εποχή μας υπάρχουν πολλά διαθέσιμα λογισμικά οπτικής αναγνώρισης με πολύ υψηλή ακρίβεια και χαμηλό κόστος. Συνήθως χρησιμοποιείται για την καταχώρηση δεδομένων, ώστε να μπορούν να μπορούν να επεξεργαστούν, αναζητηθούν, αποθηκευτούν. Η οπτική αναγνώριση είναι ένα πεδίο έρευνας στην αναγνώριση προτύπων καθώς και στην τεχνητή νοημοσύνη.

1870	The very first attempts
1940	The modern version of OCR.
1950	The first OCR machines appear
1960 - 1965	First generation OCR
1965 - 1975	Second generation OCR
1975 - 1985	Third generation OCR
1986 ->	OCR to the people

Εικόνα 6: Οι γενιές της OCR

2.3 Πως λειτουργεί

Δύο είναι οι κύριοι τρόποι εφαρμογής της οπτικής αναγνώρισης, η "η αντιστοίχηση με πρότυπα" και η "εξαγωγή χαρακτηριστικών". Η μέθοδος της αντιστοίχησης προτύπων είναι πιο διαδεδομένη αλλά περιορίζεται αρκετά σε σχέση με την μέθοδο της εξαγωγής χαρακτηριστικών. Η σύγχρονη τεχνολογία χρησιμοποιεί τον συνδυασμό και των δύο τεχνολογιών για την καλύτερη επίτευξη αποτελεσμάτων, κυρίως σε χειρόγραφα έγγραφα.

- **Αντιστοίχιση με πρότυπα**

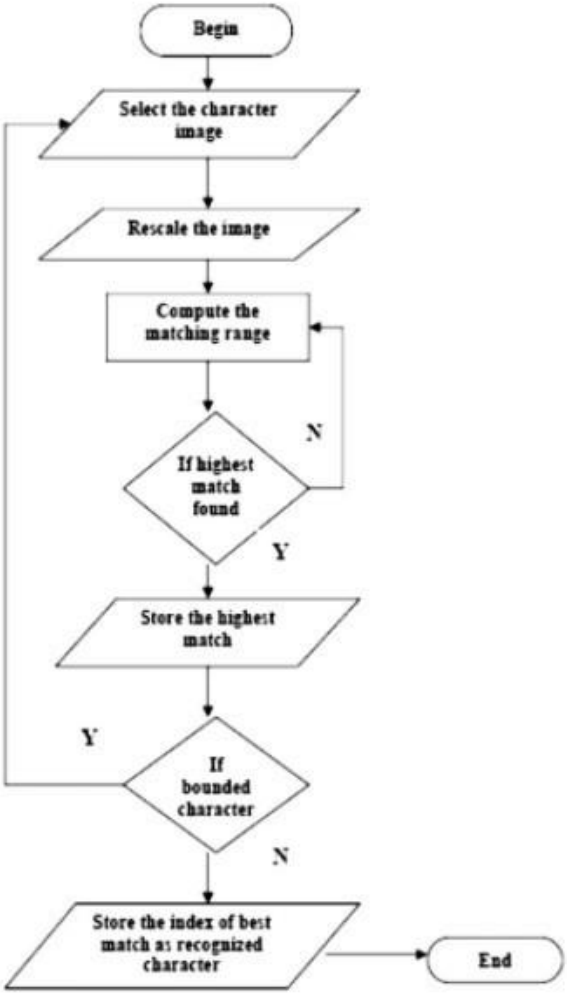
Η αντιστοίχιση με πρότυπα (template matching) είναι μια τεχνική για την εύρεση περιοχών μίας εικόνας που ταιριάζει με ένα πρότυπο. Είναι μια υψηλού επιπέδου τεχνική στον τομέα της μηχανικής ορατότητας (machine visibility) που καθορίζει τα μέρη μιας εικόνας που ταιριάζουν σε προκαθορισμένα πρότυπα. Είναι ευέλικτη και σχετικά απλή στην χρήση, γεγονός που την καθιστά ως την πιο γνωστή μέθοδο [6]. Η αντιστοίχιση με πρότυπα είναι μια ιδιαίτερα χρονοβόρα διαδικασία καθώς απαιτούνται πολλές επαναλήψεις για κάθε χαρακτήρα. Αφορά την αναγνώριση χαρακτήρων από έτοιμα πρότυπα ή περιγράμματα χαρακτήρων. Ο σαρωτής την εικόνα εγγράφου στον υπολογιστή και το λογισμικό της οπτικής αναγνώρισης προσπαθεί να ταιριάξει βάση πιθανότητας, τους χαρακτήρες από το σαρωμένο αρχείο εικόνας με πρότυπα που έχει αποθηκευμένα. Αν η εικόνα αντιστοιχεί με ένα αναγνωρισμένο χαρακτήρα, τότε αντιστοιχίζεται με αυτόν τον χαρακτήρα και αναγνωρίζεται. Οι εικόνες σε ένα λογισμικό οπτικής αναγνώρισης είναι σε μορφή bitmap για κάθε χαρακτήρα κάθε μεγέθους. Το λογισμικό διαβάζει την εικόνα μέσω του σαρωτή, ο οποίος σαρώνει και προσπαθεί να αντιστοιχίσει κάθε χαρακτήρα με την αντίστοιχη λέξη. Παραδείγματος χάρη, αν το σύστημα εντόπιζε έναν χαρακτήρα "Κ" τότε έψαχνε όλα τα πρότυπα από το Α μέχρι το ω σε όλα τα αποθηκευμένα μεγέθη και αν εντόπιζε μία εικόνα που να έμοιαζε με το "Κ", το αντιστοίχιζε.

- **Εξαγωγή Χαρακτηριστικών**

Γνωστή και ως ευφυής αναγνώριση χαρακτήρων (intelligent character recognition), η εξαγωγή χαρακτηριστικών πρόκειται για ένα είδος οπτικής αναγνώρισης που δεν βασίζεται σε ακριβείς αντιστοιχίσεις με πρότυπα όπως είδαμε στην προηγούμενη μέθοδο, αλλά λειτουργεί με ένα πιο σοφιστικό τρόπο αναγνώρισης χαρακτήρων, όπως η ανίχνευση επιμέρους συστατικών στοιχείων, όπως οι γωνίες, γραμμές, ενώσεις κ.α ενός χαρακτήρα. Η εφαρμογή των αντιστοιχίσεων γίνεται με την μορφή κανόνων. Για παράδειγμα ένας κανόνας θα μπορούσε να είναι ο εξής: Αν εντοπιστούν δύο κάθετες γραμμές που κλίνουν η μια προς την άλλη, "/" και "\", οι κορυφές τους ενώνονται και τα κέντρα των συγκλινουσών αυτών γραμμών ενώνονται με μια γραμμή "-", τότε είναι το γράμμα "Α". Με την εφαρμογή αυτού του κανόνα θα μπορούσε να εντοπίσει όλα τα "Α" ανεξάρτητα από το μέγεθος ή τον τύπο γραμματοσειράς που χρησιμοποιήθηκε στο έγγραφο. Η εξαγωγή χαρακτηριστικών είναι ένα πολύ σημαντικό βήμα στην αναγνώριση χειρόγραφων, στο οποίο θα αναφερθούμε αναλυτικότερα παρακάτω, στις φάσεις από τις οποίες αποτελείται ένα σύστημα HWR.

- **Υβριδική Αναγνώριση**

Ο συνδυασμός των παραπάνω τεχνικών αναφέρεται ως υβριδική αναγνώριση και χρησιμοποιείται κυρίως για την αναγνώριση χειρόγραφων χαρακτήρων καθώς όπως αναφέραμε και στο πρώτο κεφάλαιο είναι μια αρκετά πιο πολύπλοκη διαδικασία σε σχέση με την οπτική αναγνώριση χαρακτήρων, όπου χρησιμοποιούνται οι 2 προηγούμενες μέθοδοι που αναφέραμε.



Εικόνα 7: Διάγραμμα ροής της αντιστοίχισης με πρότυπα

2.4 Οι κατηγορίες της HTR

Η αναγνώριση χειρόγραφων ταξινομείται σε δυο κατηγορίες, αναλόγως αν η αναγνώριση γίνεται σε πραγματικό χρόνο ή όχι. Οι δύο αυτές κατηγορίες είναι η online και η offline αναγνώριση αντίστοιχα. Η offline αναγνώριση χειρόγραφου περιλαμβάνει την αυτόματη μετατροπή του κειμένου σε κωδικούς γραμμάτων ώστε να μπορούν να χρησιμοποιηθούν σε εφαρμογές υπολογιστή και σε εφαρμογές επεξεργασίας κειμένου. Η online αναγνώριση ασχολείται με μια ροή δεδομένων που προέρχεται από έναν μορφοτροπέα όσο ο χρήστης γράφει. Συνήθως το υλικό για την συλλογή των δεδομένων είναι ένα tablet ψηφιοποίησης που είναι ευαίσθητο στην πίεση. Όταν ο χρήστης γράφει στο tablet, οι διαδοχικές κινήσεις της πένας μετατρέπονται σε μια σειρά ηλεκτρονικών σημάτων που απομνημονεύονται και αναλύονται από τον υπολογιστή. Η οπτική αναγνώριση (OCR) αναφέρεται συνήθως ως μια διαδικασίας offline αναγνώρισης, που σημαίνει ότι το σύστημα σαρώνει και αναγνωρίζει στατικές εικόνες των χαρακτήρων [7].

- **Online αναγνώριση**

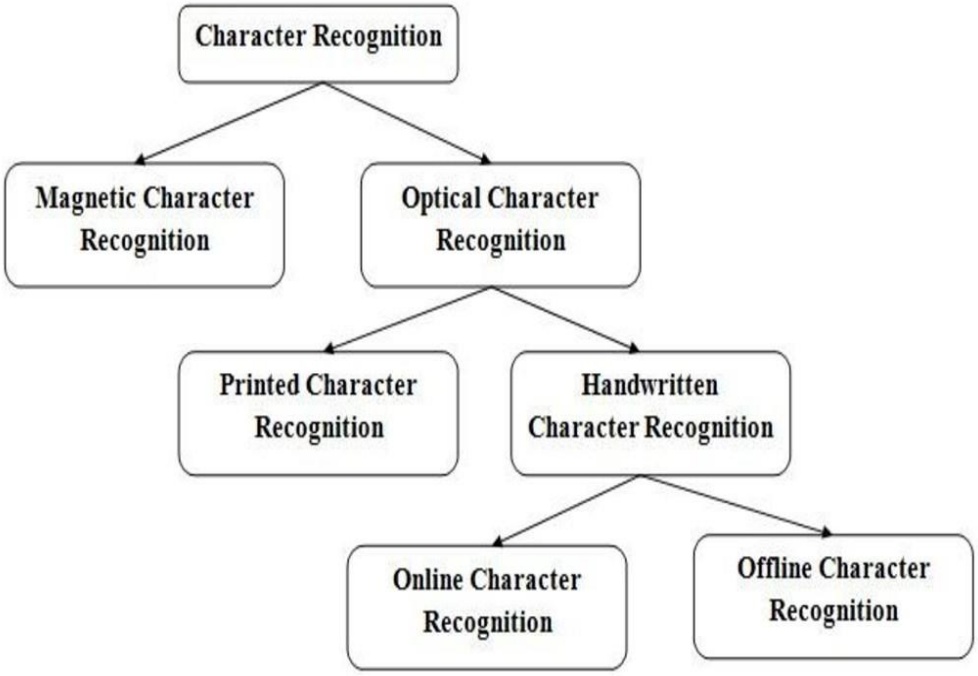
Η online αναγνώριση χειρόγραφου αναφέρεται στην διαδικασία αναγνώρισης χειρόγραφου που έχει γραφτεί σε ψηφιοποιητή. Στην περίπτωση αυτή, το χειρόγραφο αποθηκεύεται σε ψηφιακή μορφή με διαφορετικά μέσα. Συνήθως, χρησιμοποιείται μια ειδική γραφίδα πάνω σε ηλεκτρονική επιφάνεια. Όσο, το στυλό κινείται πάνω στην επιφάνεια, οι δισδιάστατες συντεταγμένες διαδοχικών σημείων αντιπροσωπεύονται σε σειρά. Είναι γενικά αποδεκτό ότι η μέθοδος της online αναγνώρισης χειρόγραφων κειμένων έχει πετύχει καλύτερα αποτελέσματα σε σχέση με την μέθοδο της offline αναγνώρισης. Αυτό, μπορεί να οφείλεται στο γεγονός ότι μπορούν να συλλαμβάνονται περισσότερες πληροφορίες στην περίπτωση της online αναγνώρισης χειρόγραφων, όπως η κατεύθυνση, η ταχύτητα και η σειρά κινήσεων της γραφής [7].

- **Offline αναγνώριση**

Η offline αναγνώριση χειρόγραφου χαρακτήρα αναφέρεται στην διαδικασία αναγνώρισης λέξεων που έχουν σαρωθεί από μια επιφάνεια, όπως ένα φύλλο χαρτί και αποθηκεύονται ψηφιακά σε ασπρόμαυρη μορφή (grayscale). Μετά την αποθήκευση είναι σύνηθες να εκτελούνται και περαιτέρω επεξεργασίες στην εικόνα που επιτρέπουν την καλύτερη αναγνώρισή της. Θα αναφερθούμε αναλυτικότερα στην επόμενη παράγραφο όπου θα περιγράψουμε τις φάσεις ενός μηχανισμού offline αναγνώρισης χειρόγραφων. Η offline αναγνώριση χαρακτήρων μπορεί να ομαδοποιηθεί σε δύο κατηγορίες:

- **Magnetic Character Recognition (MCR) Optical Character Recognition (OCR)**

Στην μέθοδο MRC, οι χαρακτήρες εκτυπώνονται με μαγνητικό μελάνι. Η συσκευή ανάγνωσης μπορεί να αναγνωρίσει τους χαρακτήρες σύμφωνα με το μαγνητικό πεδίο κάθε χαρακτήρα. Η μέθοδος χρησιμοποιείται κυρίως στις τράπεζες για τον έλεγχο ταυτότητας των υπογραφών. Η OCR διαδικασία ασχολείται με την αναγνώριση χαρακτήρων που προσλαμβάνονται από οπτικά μέσα όπως ένας σαρωτής ή κάμερα. Οι εικόνες είναι σε μορφή εικονοστοιχείων (pixels) και μπορεί να είναι τυπωμένοι ή χειρόγραφοι, οποιουδήποτε μεγέθους, σχήματος ή προσανατολισμού. Τα OCR συστήματα μπορούν να υποδιαιρεθούν και αυτά σε δύο κατηγορίες, σε αυτά που αναγνωρίζουν χειρόγραφο χαρακτήρα και σε αυτά που αναγνωρίζουν τυπωμένο χαρακτήρα [7]. Η αναγνώριση χειρόγραφου χαρακτήρα όπως έχουμε προαναφέρει είναι μια πιο δύσκολη διαδικασία, λόγω των πολλών διαφορετικών στυλ γραφής των ανθρώπων, ενώ στην περίπτωση των τυπωμένων χαρακτήρων η αναγνώριση είναι πιο εύκολη λόγω των κλασσικών γραμματοσειρών όπως Times New Roman, Arial και άλλες που μπορεί να είναι τυπωμένοι οι χαρακτήρες.



Εικόνα 8: Ταξινόμηση της αναγνώρισης χαρακτήρα

Τα μειονεκτήματα της offline αναγνώρισης σε σχέση με την online είναι τα εξής: Απαιτεί συνήθως ατελείς τεχνικές προεπεξεργασίας πριν από τα στάδια της εξαγωγής χαρακτηριστικών και αναγνώρισης.

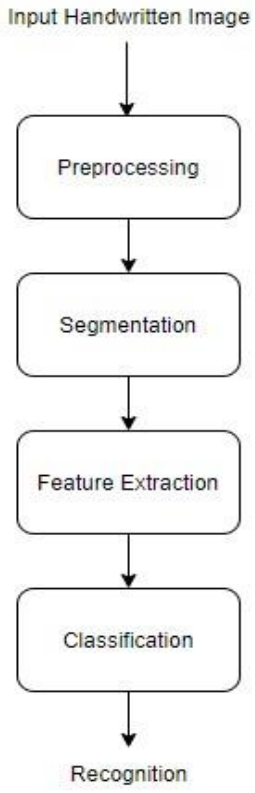
- Δεν λαμβάνουν υπόψη δυναμικές πληροφορίες όπως η κατεύθυνση και η ταχύτητα της γραφής και σε ορισμένες περιπτώσεις η πίεση που ασκήθηκε στο χαρτί κατά την σύνταξη του χαρακτήρα.
- Δεν γίνεται αναγνώριση σε πραγματικό χρόνο.

s.no	comparison	Online characters	Offline characters
1.	Availablility of no of pen strokes	yes	No
2.	Raw data requirements	#samples/sec(e.g.100)	#dots/inch( )e.g.300
3.	Way of writing	Using digital pen on lcd	Paper document
4.	Recognition rates	higher	lower
5.	accuracy	higher	lower

Εικόνα 9: Σύγκριση online και offline αναγνώρισης [7]

2.5 Οι φάσεις της HTR

Τα συστήματα αναγνώρισης χειρόγραφων χαρακτήρων αποτελούνται από αρκετές φάσεις, όπως η προεπεξεργασία (preprocessing), η τμηματοποίηση (segmentation), η εξαγωγή χαρακτηριστικών (feature extraction), η ταξινόμηση (classification) και η αναγνώριση (recognition) [8][9]. Η έξοδος του ενός είναι η είσοδος του επόμενου βήματος. Το έργο της προεπεξεργασίας σχετίζεται με την αφαίρεση του θορύβου, την δυαδικοποίηση, την αραίωση, την αλλαγή μεγέθους κ.α για την βελτίωση της ποιότητας της εικόνας. Η τμηματοποίηση είναι μια λειτουργία που επιδιώκει να αποσυνθέσει μια εικόνα αποτελούμενη από μια ακολουθία χαρακτήρων σε δευτερεύουσες εικόνες μεμονωμένων συμβόλων. Η απόφαση της λειτουργίας, ότι το μοτίβο που απομονώθηκε είναι εκείνο ενός χαρακτήρα ή κάποιας άλλης αναγνωρίσιμης μονάδας, όπως μια λέξη, μπορεί να είναι σωστή ή λάθος [1]. Η εξαγωγή χαρακτηριστικών είναι η διαδικασία ανάκτησης των περισσότερο σημαντικών στοιχείων από τα μη επεξεργασμένα δεδομένα. Με τα πιο σημαντικά στοιχεία εννοούμε ότι με βάσει μόνο αυτά τα στοιχεία ο χαρακτήρας ή λέξη μπορούν να αναπαρασταθούν με ακρίβεια. Ουσιαστικά, είναι η εξαγωγή ενός συνόλου χαρακτηριστικών, τα οποία μεγιστοποιούν το ποσοστό αναγνώρισης με το μικρότερο ποσό στοιχείων. Αυτή η λειτουργία είναι πολύ σημαντική για την ταξινόμηση στο επόμενο βήμα [9]. Η ταξινόμηση βοηθά στην σύγκριση του ποσοστού ακρίβειας και εκπαίδευσης, νέων τεχνικών εξαγωγής χαρακτηριστικών με μερικές από τις ήδη υπάρχουσες [3].



Εικόνα 10: Ένα τυπικό HWR σύστημα

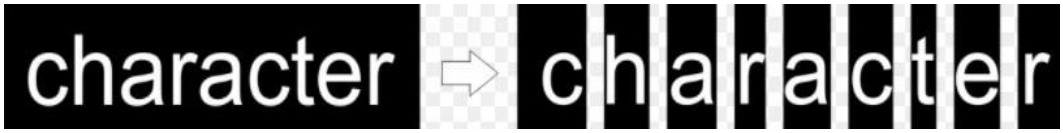
- **Προεπεξεργασία (Preprocessing)**

Ο στόχος της προεπεξεργασίας είναι η εξάλειψη των ανεπιθύμητων χαρακτηριστικών ή του θορύβου σε μια εικόνα χωρίς την αφαίρεση όμως σημαντικών πληροφοριών. Τεχνικές προεπεξεργασίας απαιτούνται σε έγχρωμες, ασπρόμαυρες ή και δυαδικές εικόνες, οι οποίες περιέχουν κείμενο. Δεδομένου ότι η επεξεργασία έγχρωμων εικόνων είναι υπολογιστικά αρκετά κοστοβόρα, τα συστήματα αναγνώρισης χειρόγραφου χαρακτήρα χρησιμοποιούν δυαδικές ή ασπρόμαυρες εικόνες [10]. Η προεπεξεργασία μειώνει τα ασυνεπή στοιχεία και τον θόρυβο, ενισχύει την εικόνα και την προετοιμάζει για τις επόμενες φάσεις της HWR. Ουσιαστικά, αυτό που επιδιώκουμε σε αυτή την φάση είναι να μετατρέψουμε την εικόνα σε κατάλληλη μορφή για τις επόμενες φάσεις ώστε να αυξήσουμε το ποσοστό αναγνώρισης του χαρακτήρα [12]. Ορισμένες λειτουργίες σε αυτή την φάση είναι η δυαδικοποίηση, η μείωση θορύβου, διόρθωση κλίσης, μορφολογικές επεμβάσεις, όπως την αύξηση της αντίθεσης της εικόνας και την αύξηση του πλάτους των γραμμών του χαρακτήρα, αφαίρεση κλίσης και συμπίεση.

- **Τμηματοποίηση (Segmentation)**

Η τμηματοποίηση είναι η διαδικασία απομόνωσης του κειμένου που παρευρίσκεται σε μια εικόνα από το φόντο (background) της εικόνας [10]. Οι τεχνικές που χρησιμοποιούνται είναι η τμηματοποίηση γραμμής (line segmentation), η τμηματοποίηση λέξης (word segmentation) και η τμηματοποίηση χαρακτήρα (character segmentation) [10][11]. Σε επίπεδο κειμένου χρησιμοποιούνται με την σειρά που αναφέρθηκαν. Η

τμηματοποίηση γραμμής, διαχωρίζει τις γραμμές του κειμένου, η τμηματοποίηση λέξης εντοπίζει τις λέξεις μιας γραμμής, ενώ η τμηματοποίηση χαρακτήρα ξεχωρίζει τους χαρακτήρες της λέξης. Στην δική μας υλοποίηση, όπως θα δούμε και στο τέταρτο κεφάλαιο, προσπερνάμε την φάση αυτή.



Εικόνα 11: Τμηματοποίηση χαρακτήρων

Εικόνα 12: Τμηματοποίηση Λέξης

- **Εξαγωγή Χαρακτηριστικών (Feature Extraction)**

Η εξαγωγή χαρακτηριστικών είναι η διαδικασία κατά την οποία γίνεται λήψη πληροφοριών σχετικά με ένα αντικείμενο ή ομάδα αντικειμένων προκειμένου να διευκολυνθεί η διαδικασία της ταξινόμησης. Σε αυτό το στάδιο κάθε χαρακτήρας αναπαριστάται ως διάνυσμα χαρακτηριστικών. Στην συνέχεια αυτό το διάνυσμα χρησιμοποιείται από τους ταξινομητές για να ταυτοποιήσουν την είσοδο με την έξοδο [13]. Λόγω της φύσης του χειρόγραφου, όπου ο βαθμός μεταβλητότητας και ανακρίβειας είναι πολύ υψηλός, η διαδικασία της εξαγωγής χαρακτηριστικών είναι ένα δύσκολο έργο [3]. Μερικές από τις τεχνικές της εξαγωγής χαρακτηριστικών είναι οι εξής: Ανάλυση κύριων συνιστωσών (Principal Component Analysis), αμετάβλητη κλίμακα εξαγωγής χαρακτηριστικών (Scale Invariant Feature Extraction), γραμμική μεροληπτική ανάλυση (Linear Discriminant Analysis), ιστογράμματα (Histogram), κωδικός αλυσίδας (Chain Code), αυτοκωδικοποιητές (autoencoders) [11]. Όλα αυτά τα χαρακτηριστικά χρησιμοποιούνται για την εκπαίδευση του εκάστοτε συστήματος.

Υπάρχουν δύο μεγάλες κατηγορίες χαρακτηριστικών:

<u>Στατιστικά και δομικά χαρακτηριστικά</u>.

- *Στατιστικά χαρακτηριστικά (Statistical Features)*

Τα στατιστικά χαρακτηριστικά λαμβάνονται από την στατιστική κατανομή του κάθε σημείου, όπως ζωνών, στιγμών, διασταυρώσεων και ιστογράμματα προβολής [14]. Αναφέρονται επίσης και ως παγκόσμια χαρακτηριστικά, καθώς συνήθως εξάγονται σε υποεικόνες, όπως πλέγματα.

- *Δομικά χαρακτηριστικά (Structural Features)*

Ο χώρος των δοκιμών ή τοπολογικών, όπως διαφορετικά αναφέρονται, χαρακτηριστικών, εξάγεται ώστε κάθε τιμή να περιέχει πληροφορίες σχετικά με την δομή της εικόνας. Οι τιμές των χαρακτηριστικών υπολογίζονται από τις δοκιμές και γεωμετρικές ιδιότητες του χαρακτήρα.

Παραδείγματα είναι ο αριθμός οριζόντιων ή κατακόρυφων γραμμών, καμπυλών, διασταυρώσεων, αναλογία διαστάσεων, κυρτότητες και κοιλότητες.

- **Ταξινόμηση (Classification)**

Η λήψη των αποφάσεων γίνεται στην φάση αυτή. Για την αναγνώριση των χαρακτήρων, χρησιμοποιούνται τα εξαγόμενα χαρακτηριστικά από την διαδικασία της εξαγωγής χαρακτηριστικών. Ένας ταξινομητής ουσιαστικά δημιουργεί κλάσεις με ομοιογενείς ιδιότητες και ταξινομεί τις εισόδους στις κλάσεις αυτές [15]. Ο πιο παραδοσιακός ταξινομητής που χρησιμοποιείται στην αναγνώριση χειρόγραφων χαρακτήρων είναι τα νευρωνικά δίκτυα. Πέρα όμως από αυτά υπάρχουν κι άλλοι ταξινομητές, όπως για παράδειγμα, οι αλγόριθμοι Support Vector Machine (SVM), ο αλγόριθμος K-nearest neighbors, η θεωρία Bayesian (Bayesian Theory) αλλά και συνδυασμός αυτών [16]. Στην δική μας υλοποίηση το νευρωνικό δίκτυο χειρίζεται τόσο την διαδικασία της εξαγωγής χαρακτηριστικών όσο και την ταξινόμηση.

- **Μετεπεξεργασία (Postprocessing)**

Μία φάση που δεν είναι υποχρεωτική στην αναγνώριση χειρόγραφων χαρακτήρων είναι αυτή της μετεπεξεργασίας. Σε αυτή την φάση γίνεται σύνδεση με λεξικό, για να επιτευχθεί υψηλότερη ανάλυση σύνταξης και σημασιολογική ανάλυση, η οποία αφορά την επαλήθευση του αναγνωρισμένου χαρακτήρα [12].

Μηχανική Μάθηση, Βαθιά Μάθηση και Νευρωνικά Δίκτυα

3 Μηχανική Μάθηση

3.1.1 Εισαγωγή

Η μάθηση είναι η διαδικασία στην οποία συνδέονται τα γεγονότα με τις συνέπειες. Έτσι, βασικά, η μάθηση είναι ένας τρόπος να τεκμηριωθεί η αρχή της αιτίας και του αποτελέσματος. Η επιστήμη του σχεδιασμού μιας έξυπνης μηχανής αναφέρεται ως μηχανική μάθηση (machine learning) και το εργαλείο που χρησιμοποιείται για τον σχεδιασμό μιας τέτοιας έξυπνης μηχανής είναι τα νευρωνικά δίκτυα. Το νευρωνικό δίκτυο μπορεί να θεωρηθεί ως ένα μαύρο κουτί που δίνει μια επιθυμητή έξοδο για μια δεδομένη είσοδο. Επιτυγχάνεται μέσω της διαδικασίας που ονομάζεται εκπαίδευση.

Η μηχανική μάθηση διερευνά την μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα, μειώνοντας την ανθρώπινη παρέμβαση στο ελάχιστο. Ο στόχος είναι η αυτοματοποιημένη δημιουργία μοντέλων, κατάλληλων να αναπαραστήσουν τα δεδομένα και τις σχέσεις που τα διέπουν, επιτρέποντας με αυτό τον τρόπο την εξαγωγή συμπερασμάτων. Ως ορισμός για την μηχανική μάθηση έχει επικρατήσει αυτός που έδωσε ο Tom M. Mitchell και είναι ο εξής: "Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E." [17].

3.1.2 Είδη μηχανικής μάθησης

Όπως και οι άνθρωποι μαθαίνουν με ποικίλους τρόπους, εν γένει και ο τομέας της μηχανικής μάθησης έχει αναπτύξει τρεις τρόπους μάθησης: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση.

- **Επιβλεπόμενη Μάθηση (Supervised Learning)**

Η πλειοψηφία των εφαρμογών μηχανικής μάθησης ανήκουν στην κατηγορία αυτή. Στην περίπτωση της επιβλεπόμενης μάθησης ή μάθησης με επίβλεψη, όπως λέγεται και διαφορετικά, χρησιμοποιούνται μεταβλητές εισόδου και μια μεταβλητή εξόδου. Ουσιαστικά είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους σε γνωστές επιθυμητές εξόδους, με σκοπό την γενίκευση αυτής της συνάρτησης και για εισόδους με άγνωστη έξοδο [19]. Η διαδικασία της μάθησης σταματάει όταν ο αλγόριθμος επιτύχει ένα αποδεκτό βαθμό απόδοσης. Η επιβλεπόμενη μάθηση χρησιμοποιείται για επίλυση προβλημάτων, ταξινόμησης (classification), πρόγνωσης (prediction) και διερμηνείας (interpretation).

- **Μη-επιβλεπόμενη Μάθηση (Unsupervised Learning)**

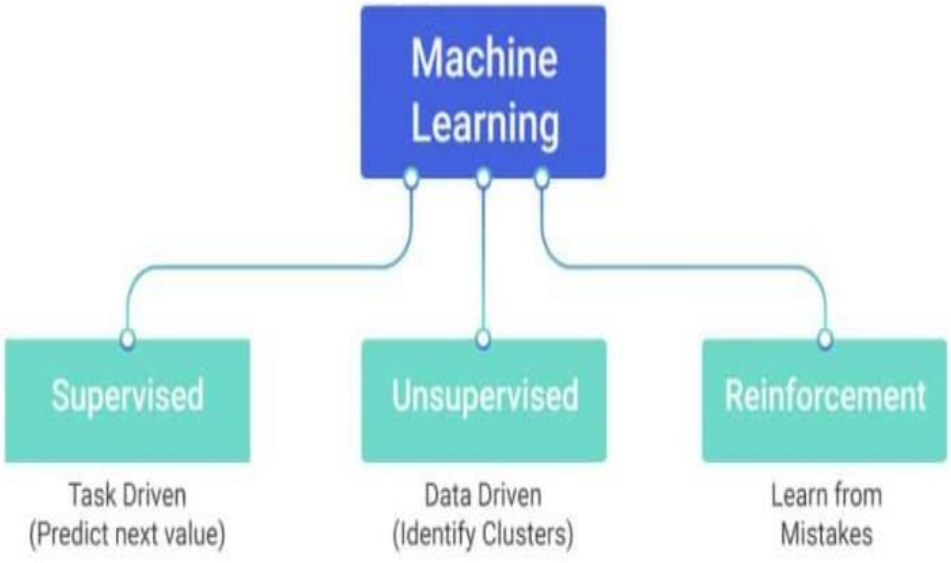
Στην περίπτωση της μη-επιβλεπόμενης μάθησης ή μάθησης χωρίς επίβλεψη, έχουμε μόνο μεταβλητές εισόδου. Ο στόχος της μη-επιβλεπόμενης μάθησης είναι να μοντελοποιήσει την υποκείμενη δομή ή διανομή στα δεδομένα, προκειμένου να μάθει περισσότερα για αυτά. Ονομάζεται μηεπιβλεπόμενη μάθηση, διότι σε αντίθεση με την επιβλεπόμενη μάθηση δεν υπάρχουν γνωστές επιθυμητές έξοδοι. Ουσιαστικά, είναι η διαδικασία όπου ο αλγόριθμος

κατασκευάζει ένα μοντέλο για ένα σύνολο εισόδων χωρίς να γνωρίζει τις εξόδους [18]. Στόχος είναι να ανακαλυφθεί πληροφορία σχετικά με την δομή και την συμπεριφορά των δεδομένων. Χρησιμοποιείται σε προβλήματα ομαδοποίησης (clustering) και ανάλυσης συσχετισμών

(association analysis).

- **Ενισχυτική Μάθηση (Reinforcement Learning)**

Η ενισχυτική μάθηση είναι ένας γενικός όρος ο οποίος έχει δοθεί σε μια οικογένεια τεχνικών στις οποίες ο αλγόριθμος μάθησης προσπαθεί να "μάθει" μια στρατηγική ενεργειών μέσα από την άμεση αλληλεπίδραση με το περιβάλλον. Εφαρμόζεται κυρίως σε προβλήματα σχεδιασμού (planning), όπως στον έλεγχο κίνησης ρομπότ και στη βελτιστοποίηση εργασιών σε εργοστάσια.



Εικόνα 13: Είδη Μηχανικής Μάθησης

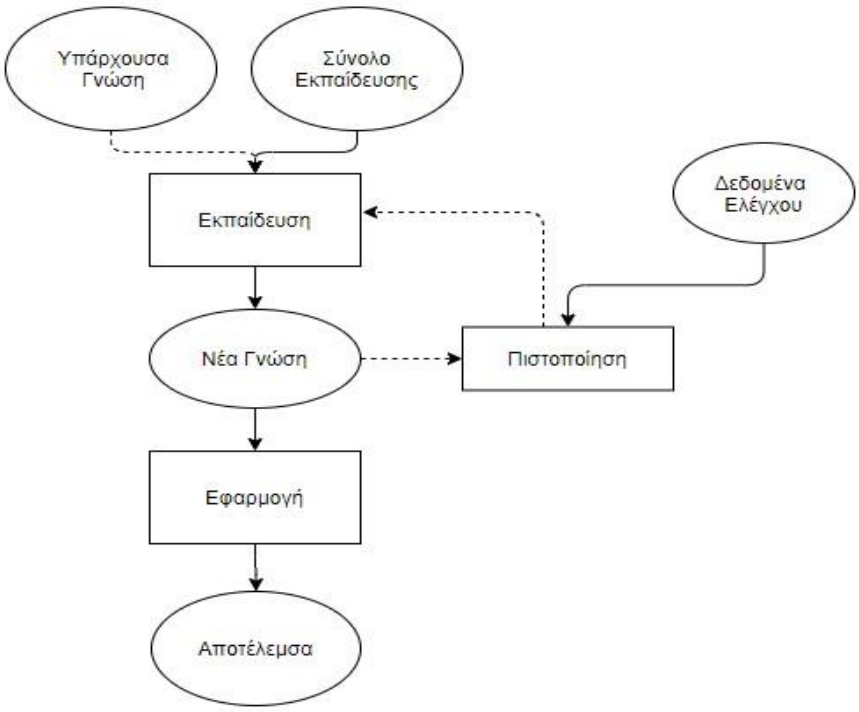
Μεταξύ της επιβλεπόμενης και της μη-επιβλεπόμενης μάθησης υπάρχει η ήμιεπιβλεπόμενη μάθηση, όπου το σύστημα μάθησης λαμβάνει ένα σύνολο δεδομένων εκπαίδευσης που αποτελείται από ένα μικρό αριθμό δεδομένων με γνωστές τις εξόδους τους και ένα μεγάλο αριθμό δεδομένων χωρίς να είναι γνωστές οι έξοδοί τους και στην συνέχεια παράγει προβλέψεις για νέα δεδομένα [20].

Για κάθε πρόβλημα στο χώρο της μηχανικής μάθησης που τίθεται προς επίλυση υπάρχει ένα κατάλληλο είδος μάθησης και για κάθε είδος μάθησης υπάρχει τουλάχιστον ένας κατάλληλος αλγόριθμος που μπορεί να χρησιμοποιηθεί. Ακολουθούν ορισμένοι από τους κοινούς αλγόριθμους μηχανικής μάθησης που χρησιμοποιούνται ευρέως:

1. Γραμμικής Παλινδρόμησης (Linear Regression)

2. Λογιστικής Παλινδρόμησης (Logistic Regression)
3. Δέντρα Απόφασης (Decision Trees)
4. Μηχανές Διανυσμάτων Υποστήριξης (Support-Vector Machines)
5. Naïve Bayes Αλγόριθμος
6. Τυχαίο Δάσος (Random Forest)
7. K-Means Αλγόριθμος
8. kNN (k-Nearest Neighbor)

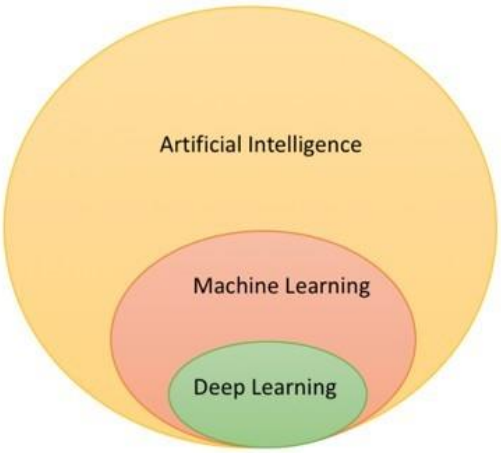
Στην παρακάτω εικόνα αποτυπώνεται ο γενικός τρόπος λειτουργίας των αλγόριθμων μηχανικής μάθησης. Η κυριότερη φάση του κάθε αλγορίθμου είναι η εκπαίδευση.



Εικόνα 14: Γενικός τρόπος λειτουργίας αλγόριθμων μηχανικής μάθησης

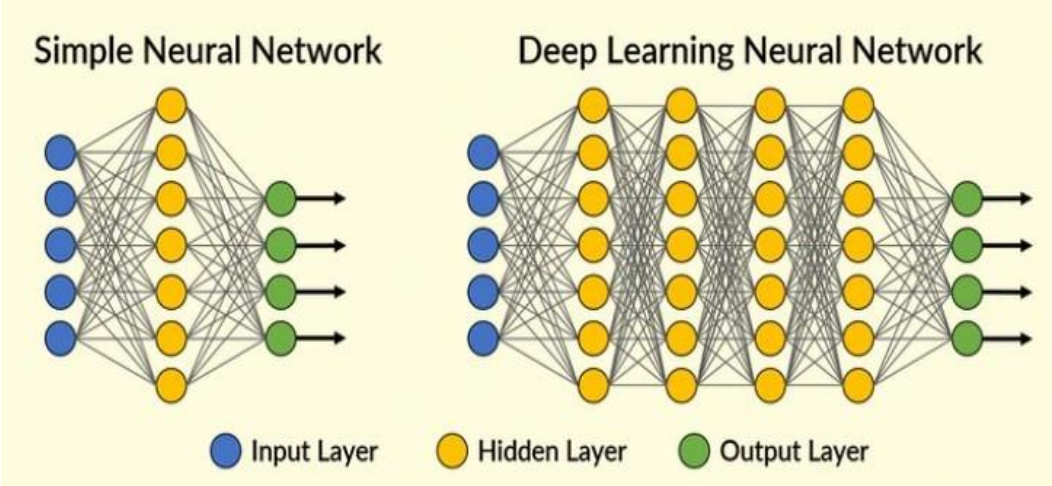
3.2 Βαθιά Μάθηση

Η βαθιά μάθηση (deep learning) είναι μια λειτουργία της τεχνητής νοημοσύνης (AI) που μιμείται τη λειτουργία του ανθρώπινου εγκεφάλου κατά την επεξεργασία δεδομένων και την δημιουργία προτύπων για χρήση στην λήψη αποφάσεων. Η βαθιά μάθηση είναι ένα υποσύνολο της μηχανικής μάθησης και όπως και η μηχανική μάθηση, έχει και αυτή μάθηση με ή χωρίς επίβλεψη και ενισχυτική μάθηση. Η τεχνητή νοημοσύνη είναι ο κλάδος της επιστήμης των υπολογιστών που μελετά την ανάπτυξη μηχανών που “σκέφτονται” και “εργάζονται” όπως οι άνθρωποι.



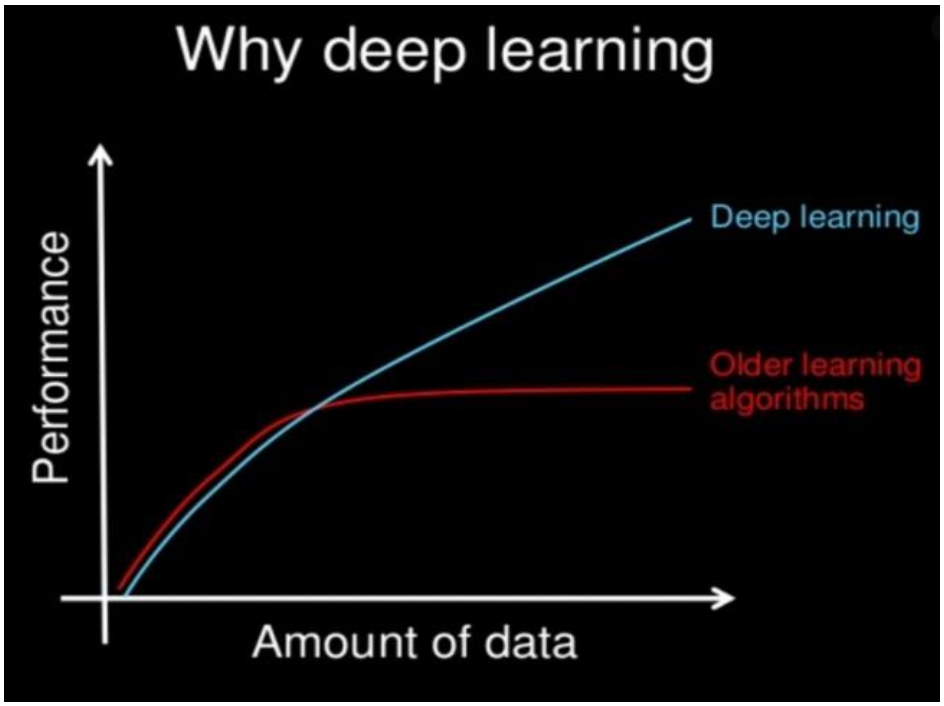
Εικόνα 15: Σχέση τεχνητής νοημοσύνης, μηχανικής μάθησης και βαθιάς μάθησης

Ο όρος “deep” στο deep learning αναφέρεται στην χρήση πολλαπλών επιπέδων στο δίκτυο. Ένα τυπικό νευρωνικό δίκτυο έχει ένα με δυο κρυφά επίπεδα μεταξύ του επιπέδου εισόδου και εξόδου. Όταν ένα δίκτυο περιέχει τρία ή περισσότερα κρυφά επίπεδα καλείται βαθύ νευρωνικό δίκτυο. Τα μοντέλα βαθιάς μάθησης εκπαιδεύονται χρησιμοποιώντας μεγάλα σύνολα δεδομένων με ετικέτες και αρχιτεκτονικές νευρωνικών δικτύων που μαθαίνουν χαρακτηριστικά απευθείας από τα δεδομένα χωρίς την ανάγκη χειροκίνητης εξαγωγής χαρακτηριστικών.



Εικόνα 16: Σύγκριση ενός τυπικού νευρωνικού δικτύου με ενός δικτύου βαθιάς μάθησης

Ένα μεγάλο πλεονέκτημα της βαθιάς μάθησης και ένα βασικό στοιχείο στην κατανόηση του γιατί γίνεται δημοφιλές είναι ότι τροφοδοτείται από τεράστιο όγκο δεδομένων.



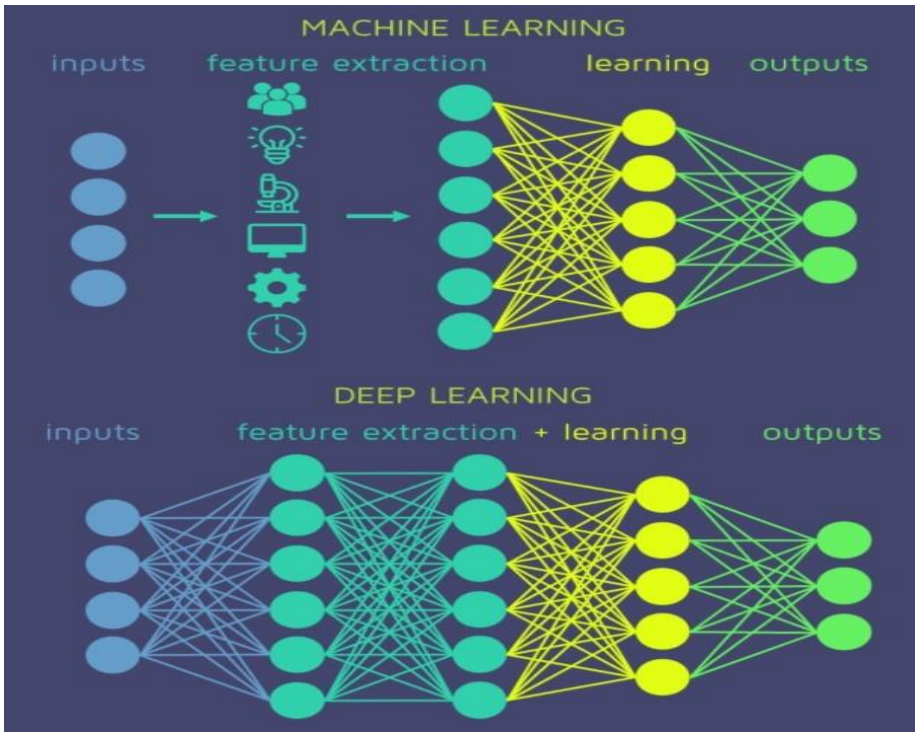
Εικόνα 17: Σχέση όγκου δεδομένων και απόδοσης

Σύγκριση μηχανικής μάθησης και βαθιάς μάθησης

Αυτή είναι μια συνηθισμένη ερώτηση, δηλαδή ποίες είναι οι διαφορές μεταξύ αυτών. Οι αλγόριθμοι βαθιάς μάθησης είναι αλγόριθμοι μηχανικής μάθησης, οπότε ίσως είναι καλύτερο να σκεφτούμε τι κάνει τη βαθιά μάθηση ξεχωριστή στον τομέα της μηχανικής μάθησης. Η απάντηση είναι η δομή του νευρωνικού δικτύου, η χαμηλότερη ανάγκη για ανθρώπινη παρέμβαση και οι μεγαλύτερες απαιτήσεις δεδομένων.

Πρώτα απ΄ όλα, ενώ οι παραδοσιακοί αλγόριθμοι μηχανικής μάθησης έχουν μια μάλλον απλή δομή, όπως η γραμμική παλινδρόμηση ή ένα δέντρο αποφάσεων, η βαθιά μάθηση βασίζεται σε ένα τεχνητό νευρωνικό δίκτυο (ANN). Αυτό το πολυεπίπεδο ANN είναι, όπως και ο ανθρώπινος εγκέφαλος, πολύπλοκος και αλληλένδετος.

Δεύτερον, οι αλγόριθμοι βαθιάς μάθησης απαιτούν πολύ λιγότερη ανθρώπινη παρέμβαση. Σε έναν παραδοσιακό αλγόριθμο μηχανικής μάθησης, ένας προγραμματιστής θα επέλεγε χειροκίνητα τα χαρακτηριστικά και τον ταξινομητή. Στους αλγόριθμους βαθιάς μάθησης όμως τα χαρακτηριστικά εξάγονται αυτόματα και ο αλγόριθμος μαθαίνει από τα δικά του λάθη (βλ. εικόνα παρακάτω).



Εικόνα 18: Μηχανική Μάθηση vs Βαθιά Μάθηση

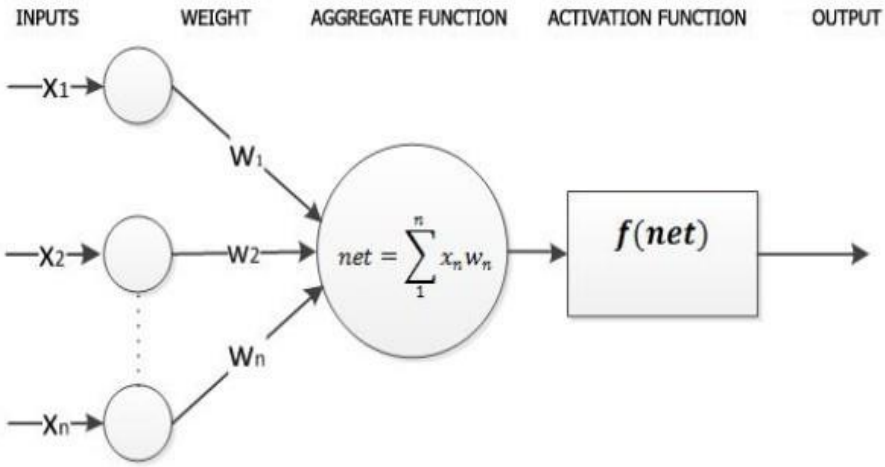
Τρίτον, η βαθιά μάθηση απαιτεί πολύ περισσότερα δεδομένα από έναν παραδοσιακό αλγόριθμο μηχανικής μάθησης για να λειτουργήσει σωστά. Λόγω της σύνθετης δομής με τα πολλαπλά επίπεδα, ένα σύστημα βαθιάς μάθησης χρειάζεται ένα μεγάλο σύνολο δεδομένων για να λειτουργήσει για να εξαλείψει τις διακυμάνσεις και να κάνει ερμηνείες υψηλής ποιότητας.

3.3 Νευρωνικά Δίκτυα

3.3.1 Τεχνητά Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα αποτελούν το δημοφιλέστερο εργαλείο του τομέα της μηχανικής μάθησης. Ο όρος νευρωνικά δίκτυα περιγράφει έναν αριθμό από διαφορετικά μαθηματικά μοντέλα, εμπνευσμένα από αντίστοιχα βιολογικά μοντέλα, δηλαδή μοντέλα που προσπαθούν να μιμηθούν τη συμπεριφορά των νευρώνων του ανθρώπινου εγκεφάλου. Πρόκειται ουσιαστικά για υπολογιστικά συστήματα τα οποία επεξεργάζονται πληροφορίες σε εξωτερικά ερεθίσματα, τις εισόδους, προσομοιώνοντας τη λειτουργία του ανθρώπινου νευρωνικού συστήματος. Αποτελούνται από δομικές υπολογιστικές μονάδες, οι οποίες είναι οργανωμένες σε επίπεδα (layers) και διασυνδεδεμένες μεταξύ τους. Κάθε υπολογιστική μονάδα, ας την αναφέρουμε ως νευρώνα, αποτελείται από πολλαπλές εισόδους και μια μόνο έξοδο. Η σύναψη, η σύνδεση μεταξύ δυο νευρώνων, σχετίζεται με κάποιο βάρος, το οποίο προσδιορίζει τον βαθμό αλληλεπίδρασης μεταξύ αυτών των δυο νευρώνων. Υπολογίζεται λοιπόν, σε κάθε νευρώνα, το άθροισμα των γινομένων κάθε εισόδου του νευρώνα και του αντίστοιχου βάρους της σύναψης. Μέσω μιας συνάρτησης ενεργοποίησης, συνήθως της Rectified Linear Unit (ReLU), δίνεται η έξοδος του νευρώνα, μόνο όμως όταν το άθροισμα που αναφέραμε είναι μεγαλύτερο από μια τιμή κατωφλιού. Η ίδια είναι η πιο δημοφιλής από τις παλαιότερες συναρτήσεις ενεργοποίησης, όπως η

Sigmoid και η Tanh και αυτό γιατί μπορεί να υπολογιστεί χωρίς ιδιαίτερο μεγάλο κόστος. Η Sigmoid χρησιμοποιείται συνήθως στη λογιστική παλινδρόμηση και έχει εύρος τιμών από 0 έως 1. Παρόμοια με την Sigmoid είναι και η Tanh αλλά το εύρος τιμών κυμαίνεται από -1 εως 1 [21]. Στην περίπτωση της ReLU, όταν η είσοδος είναι μικρότερη από μηδέν, τότε η έξοδος ισούται με μηδέν, ενώ όταν η είσοδος είναι μεγαλύτερη από μηδέν, τότε η έξοδος παίρνει την τιμή της εισόδου [22]. Αναλυτικότερα για την ReLU θα μιλήσουμε στο επόμενο κεφάλαιο καθώς είναι η συνάρτηση ενεργοποίησης που χρησιμοποιούμε στην υλοποίηση μας.



Εικόνα 19: Πρότυπο τεχνητού νευρωνικού δικτύου

Υπάρχουν αρκετοί τύποι νευρωνικών δικτύων, με τα συνελικτικά νευρωνικά δίκτυα (CNN) και τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) να είναι τα πιο διαδεδομένα όσον αφορά το κομμάτι της αναγνώρισης χειρόγραφων χαρακτήρων. Ο συνδυασμός αυτών των δύο έχει δείξει ότι επιτυγχάνει τα καλύτερα αποτελέσματα σε αυτό τον τομέα. Έτσι και στην παρούσα διπλωματική επικεντρωθήκαμε σε αυτούς τους δυο τύπους νευρωνικών δικτύων. Πάμε να κάνουμε μια αναφορά σε αυτούς.

3.3.2 Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα ανήκουν στην οικογένεια των πολυεπίπεδων νευρωνικών δικτύων, τα οποία είναι ειδικά σχεδιασμένα για χρήση σε δισδιάστατα δεδομένα, όπως εικόνες και βίντεο. Λόγω της δομής τους έχουν την δυνατότητα να επικεντρώνονται σε υπο-χώρους του προβλήματος, μειώνοντας έτσι τον υπολογιστικό όγκο και τις απαιτήσεις μάθησης. Είναι μια επιλογή αρχιτεκτονικής που αξιοποιεί τις χωρικές και χρονικές σχέσεις για να μειώσει τον αριθμό των παραμέτρων που πρέπει να μάθει [18]. Έχουν σχεδιαστεί για να μαθαίνουν αυτόματα και προσαρμοστικά χωρικές ιεραρχίες χαρακτηριστικών, από χαμηλού σε υψηλού επιπέδου μοτίβα. Τα συνελικτικά νευρωνικά δίκτυα αποτελούν μια μαθηματική δομή που αποτελούνται συνήθως από το επίπεδο της συνέλιξης (convolutional layer), το επίπεδο της συγκέντρωσης (pooling layer) και το πλήρως συνδεδεμένο επίπεδο (fully connected layer). Τα δυο πρώτα, πραγματοποιούν εξαγωγή χαρακτηριστικών, ενώ το πλήρως συνδεδεμένο επίπεδο, χαρτογραφεί τα εξαγόμενα χαρακτηριστικά στην τελική έξοδο [23]. Στις ψηφιακές εικόνες, οι τιμές των pixel αποθηκεύονται σε ένα δισδιάστατο πλέγμα (βλ. Εικόνα 22) και ένα μικρό πλέγμα παραμέτρων που ονομάζεται

φίλτρο (kernel) εφαρμόζεται σε κάθε θέση της εικόνας, γεγονός που καθιστά τα συνελικτικά νευρωνικά δίκτυα πολύ αποδοτικά για την επεξεργασία εικόνας.

Για παράδειγμα μια κοινή ακολουθία βημάτων στα συνελικτικά νευρωνικά δίκτυα είναι η παρακάτω:

1. Επίπεδο Εισόδου (Input Layer)
2. Επίπεδο Συνέλιξης (Convolutional Layer)
3. Συνάρτηση Ενεργοποίησης (Activation Fuction)
4. Επίπεδο Συγκέντρωσης (Pooling Layer)
5. Πλήρως Συνδεδεμένο Επίπεδο (Fully Connected Layer)

- **Επίπεδο Εισόδου**

Είναι η είσοδος ολόκληρου του νευρωνικού δικτύου. Όταν έχουμε σαν είσοδο μια εικόνα, τότε δέχεται ως είσοδο τα pixels τις εικόνας σε μορφή δισδιάστατου πίνακα, όπου κάθε στοιχείο του πίνακα είναι και ένα διαφορετικό pixel της εικόνας.

- **Επίπεδο Συνέλιξης**

Είναι το κύριο δομικό στοιχείο ενός συνελικτικού νευρωνικού δικτύου και αποτελεί το πιο σημαντικό επίπεδο καθώς είναι αυτό που φέρει το μεγαλύτερο υπολογιστικό όγκο. Ο πρωταρχικός στόχος της συνέλιξης είναι να εξάγει διαφορετικά χαρακτηριστικά από την είσοδο. Η συνέλιξη είναι μια γραμμική λειτουργία που περιλαμβάνει τον πολλαπλασιασμό ενός συνόλου βαρών (φίλτρα) με την είσοδο. Τα επίπεδα συνέλιξης αποτελούνται από φίλτρα (kernels), τα οποία στοχεύουν στην εξαγωγή τοπικών χαρακτηριστικών και χρησιμοποιούνται για τον υπολογισμό ενός χάρτη χαρακτηριστικών (feature map). Όταν τα στοιχεία εισόδου, εισέρχονται στο επίπεδο της συνέλιξης, πραγματοποιείται συνέλιξη μεταξύ της εισόδου και του φίλτρου, ολισθένοντας το φίλτρο κατά πλάτος και κατά ύψος της εισόδου, υπολογίζοντας το εσωτερικό γινόμενο μεταξύ του φίλτρου και του αντίστοιχου τμήματος της εικόνας. Το αποτέλεσμα του πολλαπλασιασμού του φίλτρου με ένα τμήμα της εισόδου έχει ως αποτέλεσμα μια τιμή, η οποία αναφέρεται στην συσχέτιση που έχουν τα pixels, αν υποθέσουμε ότι ως είσοδο έχουμε μια εικόνα, μεταξύ τους στο συγκεκριμένο τμήμα της εισόδου. Η εφαρμογή του φίλτρου σε όλα τα τμήματα της εισόδου έχει ως αποτέλεσμα έναν δισδιάστατο χάρτη χαρακτηριστικών. Άλλη μια παράμετρος του επιπέδου αυτού είναι το stride. Ως stride αναφέρεται ο αριθμός των pixels που ολισθαίνει το φίλτρο πάνω στην είσοδο [24].

- **Συνάρτηση Ενεργοποίησης**

Όπως αναφέρθηκε και παραπάνω η συνάρτηση ενεργοποίησης είναι υπεύθυνη για να αποφασίσει εάν ο νευρώνας (κόμβος) θα ενεργοποιηθεί ή όχι με βάση αν τιμή της σύναψης είναι μεγαλύτερη της τιμής κατωφλιού που έχει οριστεί. Για την ενεργοποίηση των κρυφών επιπέδων του συνελικτικού νευρωνικού δικτύου του μοντέλου μας έγινε χρήση της συνάρτησης ενεργοποίησης (activation function) ReLu.

- **Επίπεδο Συγκέντρωσης**

Ένα πρόβλημα με τους χάρτες χαρακτηριστικών (feature maps) είναι ότι είναι πολύ ευαίσθητοι στην θέση των χαρακτηριστικών της εισόδου. Αυτό σημαίνει ότι μικρές μεταβολές στην θέση του χαρακτηριστικού της εικόνας εισόδου θα οδηγήσει στην δημιουργία ενός νέου χάρτη χαρακτηριστικών. Μία προσέγγιση για την αντιμετώπιση αυτής της ευαισθησίας ονομάζεται down sampling. Συγκεκριμένα, δημιουργείται μια χαμηλότερης ανάλυσης εκδοχή του σήματος εισόδου, κρατώντας τα σημαντικά δομικά χαρακτηριστικά της εισόδου και παραβλέποντας τα ασήμαντα που μπορεί να μην είναι χρήσιμα για τον σκοπό της εργασίας που επιδιώκουμε να κάνουμε. Η προσέγγιση του down sampling μπορεί να επιτευχθεί στα συνελικτικά επίπεδα αλλάζοντας το βήμα της συνέλιξης (stride) σε όλη την εικόνα. Μια πιο κοινή προσέγγιση είναι η χρήση ενός επιπέδου συγκέντρωσης. Το επίπεδο αυτό στοχεύει στη σταδιακή μείωση των διαστάσεων της αναπαράστασης της εικόνας και κατά συνέπεια στη μείωση των παραμέτρων και της υπολογιστικής πολυπλοκότητας του νευρωνικού δικτύου. Το επίπεδο συγκέντρωσης είναι ένα νέο επίπεδο το οποίο τοποθετείται μετά το συνελικτικό επίπεδο και συγκεκριμένα μετά την συνάρτηση ενεργοποίησης π.χ. Relu. Η προσθήκη ενός επιπέδου συγκέντρωσης μετά το συνελεκτικό επίπεδο είναι μια κοινή δομή που χρησιμοποιείται στα συνελικτικά νευρωνικά δίκτυα και μπορεί να χρησιμοποιηθεί περισσότερες από μια φορές για την δημιουργία ενός δεδομένου μοντέλου. Η τεχνική της συγκέντρωσης περιλαμβάνει και την επιλογή μιας λειτουργίας συγκέντρωσης, σαν ένα φίλτρο που θα εφαρμοστεί στους χάρτες χαρακτηριστικών.

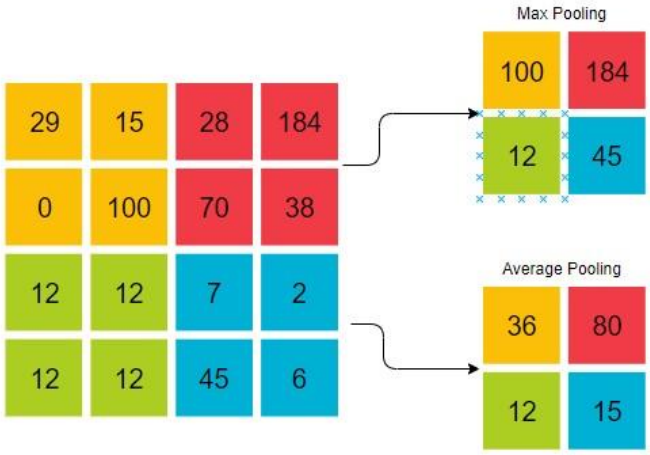
Δύο κοινές λειτουργίες που χρησιμοποιούνται στην τεχνική της συγκέντρωσης είναι:

- **Average Pooling**

Υπολογίζει την μέση τιμή κάθε patch του χάρτη χαρακτηριστικών.

- **Max Pooling**

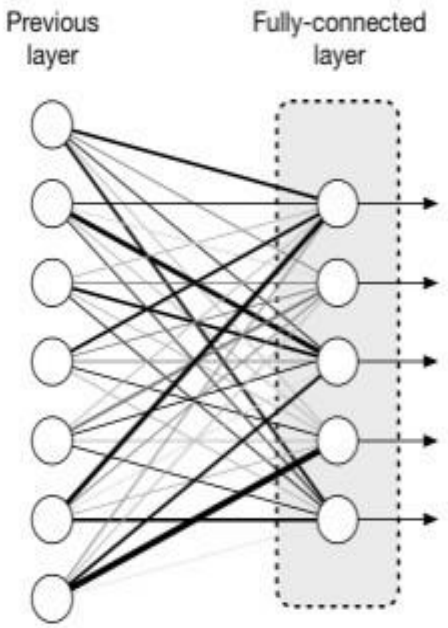
Υπολογίζει την μέγιστη τιμή κάθε patch του χάρτη χαρακτηριστικών [25].



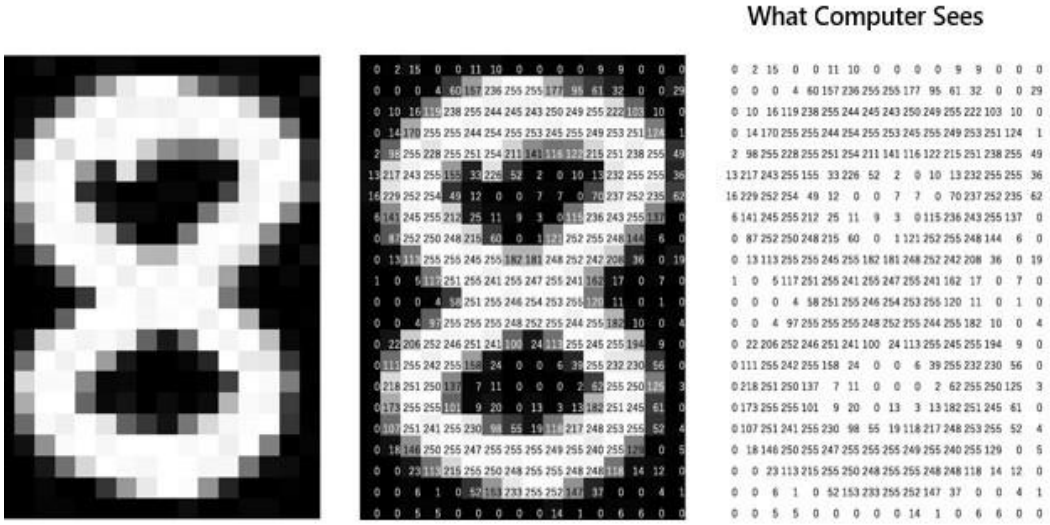
Εικόνα 20: Διαφορά μεταξύ Max και Average Pooling

- **Πλήρως Συνδεδεμένο Επίπεδο**

Το επίπεδο αυτό λειτουργεί με τον τρόπο που λειτουργούν οι αντίστοιχοι νευρώνες ενός απλού τεχνητού νευρωνικού δικτύου με την διαφορά ότι οι νευρώνες από τους οποίους αποτελείται το επίπεδο αυτό είναι πλήρως συνδεδεμένοι με τους νευρώνες του προηγούμενου επιπέδου.



Εικόνα 21: Πλήρως συνδεδεμένο επίπεδο

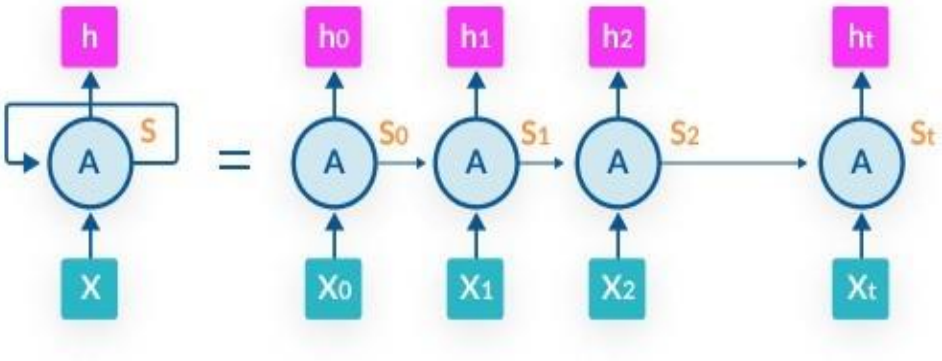


Εικόνα 22: Πως ο υπολογιστής “βλέπει” την εικόνα

3.3.3 Επαναλαμβανόμενα Νευρωνικά Δίκτυα

3.3.3.1 Γενικά

Τα επαναλαμβανόμενα νευρωνικά δίκτυα είναι μια γενίκευση του νευρωνικού δικτύου πρόσθιας τροφοδότησης (feedforward) που έχει εσωτερική μνήμη. Εκτελεί την ίδια λειτουργία για κάθε είσοδο δεδομένων, ενώ η έξοδος της τρέχουσας εισόδου εξαρτάται από τον προηγούμενο υπολογισμό. Μετά την παραγωγή της εξόδου, αντιγράφεται και αποστέλλεται πίσω στο επαναλαμβανόμενο δίκτυο. Για την λήψη της απόφασης, λαμβάνει υπόψη την τρέχουσα είσοδο και την έξοδο που είχε μάθει από την προηγούμενη είσοδο. Σε αντίθεση με τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης, τα RNN μπορούν να χρησιμοποιήσουν την εσωτερική τους κατάσταση (μνήμη) για να επεξεργαστούν ακολουθίες εισόδων. Αυτό τα καθιστά εφαρμόσιμα σε εργασίες όπως είναι η αναγνώριση χειρόγραφου χαρακτήρα. Αν και τα RNN είναι φαινομενικά ισχυρές αρχιτεκτονικές, ένα από τα σημαντικότερα προβλήματα τους είναι η περιορισμένη ικανότητα τους να μοντελοποιούν μακροπρόθεσμες εξαρτήσεις [6]. Για να λυθεί αυτή η δυσκολία εισήχθηκαν οι μακροπρόθεσμες μνήμες, Long Short-Term Memory (LSTM) [7].



Εικόνα 23: Επαναλαμβανόμενο Νευρωνικό Δίκτυο

Στην παραπάνω εικόνα (βλ. Εικόνα 23) είναι το πως λειτουργεί ένα επαναλαμβανόμενο νευρωνικό δίκτυο. Πρώτα, παίρνει σαν είσοδο την X0 από την ακολουθία εισόδων και δίνει σαν έξοδο την h0, όπου μαζί με την X1 είναι η είσοδος για το επόμενο βήμα. Ομοίως, η έξοδος h1 μαζί με την X2 είναι η είσοδος στο επόμενο βήμα και ούτω καθεξής. Με αυτό τον τρόπο, συνεχίζει να θυμάται το πλαίσιο κατά την διάρκεια της εκπαίδευσης.

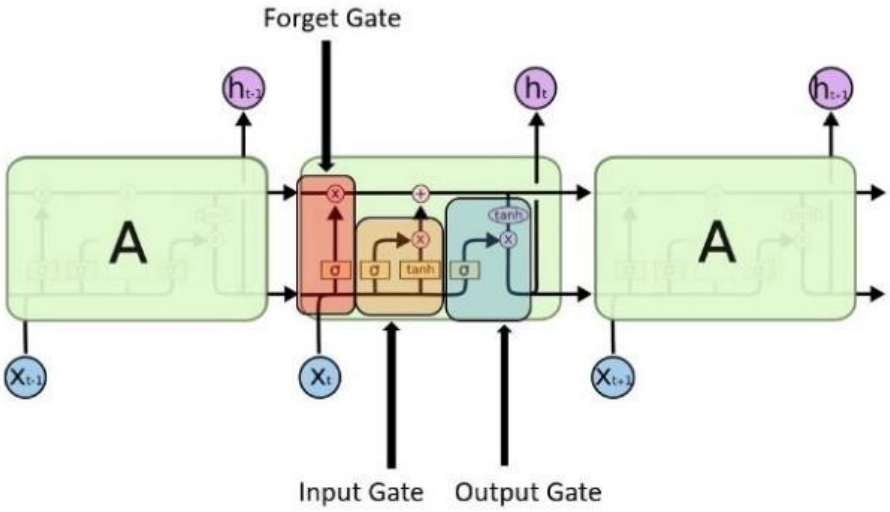
Ο τύπος για την τρέχουσα κατάσταση είναι:

$$h_t = f(h_{t-1}, x_t)$$

3.3.3.2 Long Short-Term Memory

Τα δίκτυα Long Short-Term Memory (LSTM) είναι μια τροποποιημένη έκδοση των επαναλαμβανόμενων νευρωνικών δικτύων (RNN) τα οποία συσσωρεύουν πληροφορίες με την πάροδο του χρόνου και κάνουν ευκολότερη την απομνημόνευση προηγούμενων δεδομένων στην μνήμη. Τα LSTM δίκτυα είναι κατάλληλα για ταξινόμηση, επεξεργασία και πρόβλεψη χρονικών σειρών, δεδομένου χρονικού διαστήματος άγνωστης διάρκειας. Εκπαιδεύει το μοντέλο χρησιμοποιώντας οπίσθια διάδοση (back-propagation). Ένα LSTM δίκτυο ελέγχεται από τρεις

πύλες: την πύλη εισόδου (input gate), την πύλη ξεχασμού (forget gate) και την πύλη εξόδου (output gate) [26]. Η πύλη εισόδου εξαρτάται από την τρέχουσα είσοδο και την έξοδο του προηγούμενου βήματος, όπως ακριβώς συμβαίνει και σε ένα RNN δίκτυο ενώ η πύλη ξεχασμού επιτρέπει στο δίκτυο να διαγράψει πληροφορίες που δεν χρειάζεται πια.



Εικόνα 24: Πύλες ενός LSTM δικτύου

4. Στρατηγικές

Τις τελευταίες δεκαετίες, εφαρμόστηκαν διαφορετικές προσεγγίσεις στο πρόβλημα αναγνώρισης χειρόγραφων λέξεων [3]. Οι κύριες διαφορές μεταξύ των συγγραφέων εμφανίζονται σε δύο βασικά στοιχεία των προτεινόμενων λύσεων:

α) την στρατηγική εξαγωγής χαρακτηριστικών από την εικόνα και

β) τον τρόπο αποκωδικοποίησης της εξόδου του ταξινομητή για να προβλέψει την ακολουθία των χαρακτήρων που αποτελούν μέρος της δεδομένης λέξης.

Δύο βασικές στρατηγικές χρησιμοποιούνται συνήθως για την εξαγωγή χαρακτηριστικών από την εικόνα: εφαρμογή διαφόρων τεχνικών computer vision για την ανίχνευσή τους ή την χρήση απευθείας των pixels της εικόνας ως ακατέργαστα χαρακτηριστικά (raw features). Η πρώτη στρατηγική εφαρμόζεται από τους P. Doetsch, M. Kozielski και H. Ney [54][55] όπου χρησιμοποιούν την Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis) ή αλλιώς PCA για την εξαγωγή χαρακτηριστικών σε πλαίσια σταθερού μεγέθους 8x32. Οι G. Bideault, L. Mioulet, C. Chatelain και T. Paquet στην δημοσίευσή τους με τίτλο, “Spotting handwritten words and regex using a two stage blstm-hmm architecture”, ακολουθούν επίσης την πρώτη στρατηγική όπου για την εξαγωγή των χαρακτηριστικών χρησιμοποιούν Ιστογράμματα Προσανατολισμένων Βαθμίδων (Histograms of Oriented Gradients) ή διαφορετικά (HOG) [56]. Στον αντίποδα, οι T. Bluche, J. Louradour και R. Messina χρησιμοποιούν ως είσοδο στο μοντέλο τους τα raw pixels της εικόνας [8].

Υπάρχουν επίσης δύο βασικές επιλογές για την αποκωδικοποίηση της εξόδου για την μετατροπή της, στην ακολουθία χαρακτήρων που προσδιορίζουν την χειρόγραφη λέξη. Η πρώτη επιλογή είναι με την χρήση Hidden Markov Models και η δεύτερη με την χρήση Connectionist Temporal Classification όπου εισήχθη από τον Alex Graves [2]. Αρκετοί συντάκτες τα τελευταία χρόνια εφαρμόζουν την δεύτερη επιλογή όπως οι T. Bluche, J. Louradour και R. Messina [8] και οι P. Voigtlaender, P. Doetsch και H. Ney στην βιβλιογραφία τους με τίτλο “Handwriting recognition with large multidimensional long short-term memory recurrent neural networks” [7].

Το 2006 οι B. Gatos, I. Pratikakis, A.L Kesidis και S.J Perantonis πρότειναν έναν καινοτόμο συνδυασμό δύο διαφορετικών τρόπων για την ομαλοποίηση της εικόνας και μια υβριδική εξαγωγή χαρακτηριστικών. Εδώ συνδυάζονται δύο τύποι χαρακτηριστικών. Το πρώτο χαρακτηριστικό όπου δημιουργεί ένα σύνολο ζωνών διαιρώντας την εικόνα και υπολογίζοντας την πυκνότητα των pixels κάθε ζώνης και το δεύτερο χαρακτηριστικό όπου υπολογίζει την περιοχή που σχηματίζεται από την λέξη. Έγινε χρήση δύο ταξινομητών, ο ταξινομητής ελάχιστης απόστασης και οι μηχανές διανυσμάτων υποστήριξης. Πέτυχαν ακρίβεια 80.76% χρησιμοποιώντας την IAM database [11].

Στην δημοσιευσή τους με τίτλο “Tandem HMM with Convolutional Neural Network for Handwritten Word Recognition”, οι Theodore Bluche, Hermann Ney και Christopher Kermorvant παρουσίασαν έναν συνδυασμό HMM με συνελικτικά νευρωνικά δίκτυα για την αναγνώριση χειρόγραφων λέξεων. Το συνελικτικό νευρωνικό δίκτυο αποτελούνταν από τρία συνελικτικά επίπεδα με 32, 64, 128 πίνακες χαρακτηριστικών αντίστοιχα και kernels μεγέθους 5x5, ακολουθούμενα από την συνάρτηση max-pooling. Στο τέλος της αρχιτεκτονικής αυτής υπάρχει ένα πλήρως συνδεδεμένο κρυφό επίπεδο αποτελούμενο από 700 μονάδες (units) και ένα softmax επίπεδο εξόδου. Το μοντέλο προπονήθηκε στην IAM database και πέτυχε ακρίβεια 79.50% [10].

Τα τελευταία χρόνια οι κύριες αρχιτεκτονικές μοντέλων που εφαρμόζονται στην αναγνώριση χειρόγραφων χαρακτήρων, λέξεων και κειμένων περιλαμβάνουν τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) και ειδικότερα τα δίκτυα Long Short-Term Memory (LSTM).

Παρόμοια προσέγγιση με την δική μας παρουσιάζουν οι B. Shi, X. Bai και C. Yao στην δημοσίευση τους με τίτλο "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition" το 2017 που ακολούθησαν στο μοντέλο τους για την αναγνώριση χειρόγραφων κειμένων. Πρόκειται για ένα συνελικτικό επαναλαμβανόμενο νευρωνικό (CRNN) ακολουθούμενο από ένα CTC επίπεδο για την μετατροπή της πρόβλεψης σε κείμενο. Το συνελικτικό νευρωνικό δίκτυο το οποίο είναι υπεύθυνο για την εξαγωγή χαρακτηριστικών αποτελείται από εφτά συνελικτικά επίπεδα όπου εξάγουν μια ακολουθία χαρακτηριστικών. Η ακολουθία αυτή τροφοδοτείται στο RNN δίκτυο το οποίο αποτελείται από δύο αμφίδρομα LSTM επίπεδα όπου "σκανάρουν" την ακολουθία για την πρόβλεψη της. Τέλος η έξοδος του επιπέδου αυτού οδηγείται στο CTC επίπεδο για την αποκωδικοποίηση της [13].

Το 2018 οι J. Sueiras, V. Ruiz, A. Sanchez και J. F. Velez πρότειναν μια αρχιτεκτονική νευρωνικών δικτύων που βασίζεται στον συνδυασμό ενός συνελικτικού νευρωνικού δικτύου (CNN) και μίας δομής κωδικοποιητή-αποκωδικοποιητή (encoder-decoder structure). Η εικόνα εισόδου χωρίζεται σε μία ακολουθία από υπο-περιοχές της εικόνας (patches) χρησιμοποιώντας την προσέγγιση Horizontial Sliding Window. Το συνελικτικό νευρωνικό δίκτυο εξάγει χαρακτηριστικά από όλα τα patches της εικόνας. Η ακολουθία των εξαγώμενων χαρακτηριστικών χρησιμοποιείται ως είσοδος σε ένα sequence to sequence δίκτυο (LSTM). Στην συνέχεια το δίκτυο αυτό αναγνωρίζει τους χαρακτήρες της λέξης. Το συνελικτικό δίκτυο που προτείνουν αποτελείται από δύο συνελικτικά επίπεδα ακολουθούμενα από την max-pooling συνάρτηση και ως τελικό επίπεδο ένα dropout layer. Ο κωδικοποιητής είναι ένα LSTM δίκτυο όπου διαβάζει την ακολουθία των χαρακτηριστικών από το συνελικτικό νευρωνικό δίκτυο και εξάγει της σχέσεις μεταξύ αυτών των χαρακτηριστικών. Οδηγείται στον αποκωδικοποιητή ο οποίος είναι και αυτός ένα LSTM δίκτυο όπου περιλαμβάνει και έναν μηχανισμό προσοχής (attention mechanism). Το μοντέλο τους εκπαιδεύτηκε και αξιολογήθηκε πάνω στην IAM database όπου με την χρήση λεξικού κατάφεραν να επιτύχουν ακρίβεια 87% [59]. Παρόμοια λογική ακολούθησαν και οι Ankan Kumar Bhunia, Aishik Konwer, Ayan Kumar Bhunia, Abir Bhownick, Partha P. Roy και Umapada Pal στο μοντέλο τους για την αναγνώριση χειρόγραφων κειμένων [11].

5. Υλοποίηση εργασίας

5.1 Σύνολο Δεδομένων

Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκε η IAM Handwriting Database. Δημοσιεύθηκε για πρώτη φορά στο ICDAR (International Conference on Document Analysis and Recognition) το 1999 από τους U. Marti και H. Bunke [28]. Περιέχει ασπρόμαυρες σαρωμένες εικόνες, με αγγλικού περιεχομένου λέξεις και προτάσεις, σε ανάλυση 300dpi και είναι αποθηκευμένες σε μορφή PNG. Το σύνολο δεδομένων αποτελείται από:

657 συγγραφείς

1539 σελίδες σαρωμένου κειμένου

5685 απομονωμένες και επισημασμένες προτάσεις

13353 απομονωμένες και επισημασμένες γραμμές κειμένου

115320 απομονωμένες και επισημασμένες λέξεις

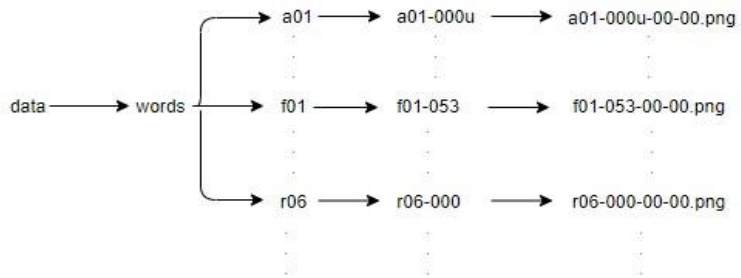
Η πρόσβαση στην database της IAM είναι ελεύθερη για τον οποιονδήποτε αφού πρώτα δημιουργήσει έναν λογαριασμό στο σύνδεσμο:

https://fki.tic.heia-fr.ch/databases/iam-handwriting-database

και στην συνέχεια μπορεί να την κατεβάσει και χρησιμοποιήσει. Είναι η πιο απαιτητική βάση δεδομένων και ίσως η δημοφιλέστερη σε ότι έχει να κάνει με την εκπαίδευση και αξιολόγηση διάφορων μοντέλων αναγνώρισης χειρόγραφων λέξεων.

Στην δική μας εργασία χρησιμοποιήθηκαν δύο αρχεία του συνόλου δεδομένων. Το αρχείο “words.tgz” όπου περιέχει 115320 εικόνες που περιλαμβάνουν λέξεις καθώς και το αρχείο “words.txt” όπου περιέχει τις ετικέτες (labels) για κάθε μια από τις εικόνες του αρχείου “words.tgz”.

Η δομή του αρχείου “words.tgz” είναι η εξής:



Ενώ η δομή του αρχείου“words.txt” είναι:

a01-000u-00-01 ok 154 507 766 213 48 NN MOVE

όπου η πρώτη στήλη είναι το όνομα της εικόνας και η τελευταία στήλη (ένατη) είναι η ετικέτα (label) της αντίστοιχης εικόνας. Οι μεσαίες στήλες μας παρέχουν κάποιες επιπλέον πληροφορίες για την εκάστοτε εικόνα οι οποίες όμως δεν χρησιμοποιούνται για την εκπαίδευση του νευρωνικού δικτύου. Στην παρακάτω εικόνα παρατίθενται δύο δείγματα από το σύνολο δεδομένων της IAM Handwriting Database.

Εικόνα 25: Δείγματα της IAM Handwriting Database

5.2 Υλικό

Η δημιουργία και η εκπαίδευση του νευρωνικού δικτύου καθώς και η ανάπτυξη της διαδικτυακής διεπαφής έγιναν στο Google Colab

5.3 Λογισμικό

5.3.1 Γλώσσα Προγραμματισμού

Ιστορικά, ένα ευρύ φάσμα διαφορετικών γλωσσών προγραμματισμού και περιβαλλόντων έχουν χρησιμοποιηθεί για την ανάπτυξη εφαρμογών και στην έρευνα της μηχανικής μάθησης. Ωστόσο, η Python θεωρείται ευρέως ως η προτιμώμενη γλώσσα για την δημιουργία και εκμάθηση μοντέλων μηχανικής μάθησης. Αυτό οφείλεται στο γεγονός ότι σε σύγκριση με άλλες γλώσσες προγραμματισμού, όπως η C++ και η Java, η σύνταξή της είναι απλούστερη. Αυτό την οδήγησε στο να σημειώσει τεράστια αύξηση της δημοτικότητας της στην επιστημονική κοινότητα με αποτέλεσμα οι πιο πρόσφατες βιβλιοθήκες μηχανικής και βαθιάς μάθησης να βασίζονται στην Python. Με βασικό επίκεντρο την αναγνωσιμότητα, η Python είναι μια ερμηνευμένη γλώσσα υψηλού προγραμματισμού, η οποία είναι εύκολη στην εκμάθηση. Με βάση αυτό, ο κώδικας είναι πιο κατανοητός από τους ανθρώπους, γεγονός που διευκολύνει την κατασκευή μοντέλων μηχανικής μάθησης. Επίσης αποτελείται από πολλές βιβλιοθήκες που την κάνουν εύκολη στην χρήση, ενώ ένα μειονέκτημα της είναι ότι είναι πιο αργή από άλλες γλώσσες προγραμματισμού

5.3.2 Βιβλιοθήκες

Μερικές από τις βιβλιοθήκες που χρησιμοποιούμε στο μοντέλο μας είναι: sys, os, numpy, tensorflow, cv2, random, spellchecker, editdistance και PIL. Πάμε να δούμε λίγο αναλυτικότερα τις σημαντικότερες από αυτές.

NumPy: Είναι μια βιβλιοθήκη που παρέχει μια απλή αλλά ισχυρή δομή δεδομένων, τον πολυδιάστατο πίνακα. Αυτό είναι το θεμέλιο πάνω στο οποίο χτίζεται σχεδόν όλη η δύναμη της εργαλειοθήκης της επιστήμης των δεδομένων της Python. Είναι μια αριθμητική βιβλιοθήκη ανοιχτού κώδικα και μπορεί να χρησιμοποιηθεί για να εκτελέσει έναν αριθμό μαθηματικών λειτουργιών σε συστοιχίες όπως τριγωνομετρικές, στατιστικές και αλγοριθμικές ρουτίνες. Χρησιμοποιείται όταν θέλουμε να εργαστούμε με πίνακες, μην ξεχνάμε ότι τα δεδομένα εισόδου μας έχουν την μορφή δισδιάστατου πίνακα. Επίσης χρησιμοποιούμε αριθμητικές συναρτήσεις της βιβλιοθήκης αυτής όπως η εύρεση μεγίστου-ελαχίστου (max-min).

cv2: Είναι μια τεράστια βιβλιοθήκη ανοιχτού κώδικα για την όραση του υπολογιστή (computer vision), την μηχανική μάθηση και την επεξεργασία εικόνας. Χρησιμοποιώντας την, μπορούμε να επεξεργαστούμε εικόνες για να εντοπίσουμε αντικείμενα, πρόσωπα ή και χαρακτήρες. Όλες οι δομές πίνακα της συγκεκριμένης βιβλιοθήκης μπορούν να μετατραπούν σε συστοιχίες Numpy και αντίστροφα. Αυτό διευκολύνει πολύ την ενσωμάτωση κι άλλων βιβλιοθηκών που χρησιμοποιούν την Numpy. Στην δική μας εργασία χρησιμοποιήσαμε αρκετές συναρτήσεις που προσφέρει η βιβλιοθήκη cv2, όπως είναι η IMREAD_GRAYSCALE, για την μετατροπή της εικόνας σε ασπρόμαυρη, η resize, για την αλλαγή του μεγέθους της εικόνας ώστε να ταιριάζει στην είσοδο του νευρωνικού δικτύου, η erode, για την διάβρωση των γραμμών των λέξεων για να είναι πιο ευδιάκριτες από το μοντέλο, η GaussianBlur, για την ομαλοποίηση της εικόνας όπως επίσης και η meanStdDev, για τον υπολογισμό της μέσης και τυπικής απόκλισης αντίστοιχα της εικόνας στην διαδικασία της κανονικοποίησης (normalization).

Tensorflow: Χρησιμοποιήσαμε την βιβλιοθήκη tensorflow για την δημιουργία και εκπαίδευση του νευρωνικού δικτύου όπου παρέχει πολλούς και προσιτούς τρόπους για αυτούς τους σκοπούς. Είναι μια βιβλιοθήκη η οποία δημιουργήθηκε από την Google με σκοπό να υπολογίζει γρήγορα αριθμητικούς υπολογισμούς. Πρόκειται για μια θεμελιώδη βιβλιοθήκη που μπορεί να χρησιμοποιηθεί για την δημιουργία μοντέλων βαθιάς μηχανικής μάθησης απευθείας ή με την χρήση βιβλιοθηκών περιτυλίγματος (wrapped libraries).

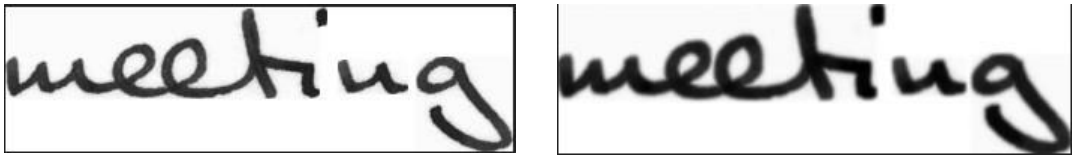
Random: Αυτή η συνάρτηση δημιουργεί ψευδο-τυχαίους αριθμούς σε όποια μορφή θέλουμε. Στην δική μας περίπτωση την χρησιμοποιήσαμε για την δημιουργία τυχαίων stretches κατά την διαδικασία της data augmentation (αύξηση δεδομένων).

Spellchecher: Η συγκεκριμένη συνάρτηση ελέγχει την έξοδο του μοντέλου με γνωστές λέξεις και ελέγχει αν υπάρχει αλλιώς την αντικαθιστά με την πιο ¨κοντινή¨ λέξη στην έξοδο.

5.4 Προεπεξεργασία Δεδομένων

Είναι γνωστό πως όσα περισσότερα δεδομένα έχει ένα νευρωνικό δίκτυο για να εκπαιδευτεί τόσο καλύτερα αποτελέσματα επιφέρει. Η data augmentation είναι μια τεχνική που χρησιμοποιείται για την δημιουργία νέων και διαφορετικών εικόνων από τις υπάρχουσες εικόνες της βάσης δεδομένων με σκοπό την αύξηση αυτής. Αυτό μπορεί να γίνει εφαρμόζοντας διάφορες τεχνικές μετασχηματισμού. Στην δικιά μας περίπτωση για τον σκοπό αυτό δημιουργούμε τυχαία stretches (τεντώματα) στα ήδη υπάρχον δεδομένα για την αύξηση της βάσης δεδομένων. Η τεχνική αυτή εφαρμόζεται μόνο στα δεδομένα εκπαίδευσης (training data) και όχι στα δεδομένα επικύρωσης (validation data). Γνωρίζοντας ότι υπάρχουν δεδομένα-εικόνες στην βάση δεδομένων της IAM Handwriting Dataset που είναι κατεστραμμένα, με την έννοια ότι το αρχείο δεν ανοίγει, αρχικά τσεκάρουμε αν η εικόνα είναι κατεστραμμένη και εφόσον είναι τότε την μετατρέπουμε σε μαύρη εικόνα. Εφαρμόζουμε επίσης μια μορφολογική συνάρτηση, συγκεκριμένα την erode από την βιβλιοθήκη cv2, όπως αναφέραμε και πιο πριν, για την διάβρωση-αύξηση των γραμμών της λέξης που περιέχεται στην εικόνα. Σε συνδυασμό με την μορφολογική συνάρτηση, χρησιμοποιούμε και μια τεχνική για την αύξηση της αντίθεσης της εικόνας. Αυτό έχει ως αποτέλεσμα τα δεδομένα εισόδου, δηλαδή οι εικόνες, να είναι πιο ευδιάκριτες από το νευρωνικό δίκτυο και συνεπώς να τις αναγνωρίζει καλύτερα. Μία άλλη σημαντική τεχνική που εφαρμόζουμε σε αυτή την φάση είναι η περικοπή της εικόνας περιμετρικά της λέξης. Αν και οι εικόνες στην βάση δεδομένων είναι ήδη

περικομμένες, αυτό το βήμα βελτιώνει αρκετά την ακρίβεια του μοντέλου σε ανεξάρτητα δεδομένα. Επιπλέον, δίνουμε στα δεδομένα το μέγεθος που θέλουμε ώστε να είναι συμβατά ως προς την είσοδο του νευρωνικού μας δικτύου και τέλος εφαρμόζουμε την κανονικοποίηση στα δεδομένα. Σκοπός της κανονικοποίησης των δεδομένων είναι να διασφαλίσει ότι κάθε στοιχείο εισόδου, στην δική μας περίπτωση τα pixels της εικόνας, έχει παρόμοια κατανομή εισόδου. Αποτελεί ένα σημαντικό βήμα στην μηχανική μάθηση καθώς κάνει τα μοντέλα να συγκλίνουν γρηγορότερα στην φάση της εκπαίδευσης.



Εικόνα 26: Σύγκριση εικόνας πριν και μετά την εφαρμογή της μορφολογικής συνάρτησης και της αύξησης της αντίθεσης

5.5 Δημιουργία και Εκπαίδευση Μοντέλου

5.5.1 Δημιουργία Νευρωνικού Δικτύου

Για την δημιουργία του νευρωνικού δικτύου γίνεται εισαγωγή της βιβλιοθήκης tensorflow όπου μας βοηθάει να δημιουργήσουμε το δίκτυο επίπεδο-επίπεδο. Αρχικά δημιουργήσαμε ένα συνελικτικό νευρωνικό δίκτυο αποτελούμενο από πέντε στρώματα της ίδιας δομής. Να τονίσουμε σε αυτό το σημείο ότι πρόκειται για ένα συνελικτικό νευρωνικό δίκτυο πρόσθιας τροφοδότησης, δηλαδή δεν υπάρχει κόμβος που να δημιουργεί κύκλο, καθώς και ότι η έξοδος κάθε επιπέδου είναι η είσοδος του επόμενου. Η είσοδος του συνελικτικού νευρωνικού δικτύου είναι μια ασπρόμαυρη εικόνα μεγέθους 128x32, όπου αποτελεί και το επίπεδο εισόδου του νευρωνικού δικτύου μας. Κάθε ένα από τα πέντε στρώματα αποτελείται από: ένα συνελικτικό επίπεδο (convolutional layer), ένα batch normalization επίπεδο, την συνάρτηση ενεργοποίησης ReLU και ένα επίπεδο συγκέντρωσης (pooling layer). Όπως αναφέραμε και στο προηγούμενο κεφάλαιο το σημαντικότερο επίπεδο ενός συνελικτικού νευρωνικού δικτύου είναι το συνελικτικό επίπεδο. Εδώ επιτελείται το έργο της συνέλιξης, ο πολλαπλασιασμός δηλαδή της εισόδου με την δισδιάστατη συστοιχία βαρών, μιας και έχουμε δισδιάστατη είσοδο, το φίλτρο ή kernel. Στο πρώτο συνελικτικό επίπεδο, το μέγεθος του φίλτρου που επιλέξαμε είναι 5x5. Η έξοδος του συνελικτικού επιπέδου τροφοδοτείται ως είσοδος στο επίπεδο batch normalization. Το επίπεδο αυτό βοηθάει ώστε η εκπαίδευση του νευρωνικού δικτύου να γίνεται πιο γρήγορα καθώς μειώνει σε αρκετά καλό βαθμό των αριθμό των εποχών (epochs) που απαιτούνται για την εκπαίδευση του νευρωνικού δικτύου. Επίσης ένα άλλο στοιχείο του επιπέδου αυτού είναι ότι προσθέτει λίγο θόρυβο στις ενεργοποιήσεις του κρυφού επιπέδου με συνέπεια να μειώνει το overfitting. Ως overfitting ορίζεται το γεγονός όπου το νευρωνικό δίκτυο έχει εκπαιδευτεί πολύ πάνω σε συγκεκριμένα δεδομένα με αποτέλεσμα να κάνει καλές προβλέψεις μόνο πάνω σε αυτές [30][31]. Ακολουθεί η συνάρτηση ενεργοποίησης ReLU, η οποία χρησιμοποιείται για να προσθέσει μη γραμμικότητα, κάτι που απαιτείται για τον χειρισμό μη γραμμικών συνόλων δεδομένου. Αν και το τυπικό είναι το επίπεδο batch normalization να τοποθετείται μετά την συνάρτηση ενεργοποίησης, καταφέραμε καλύτερα αποτελέσματα όταν τοποθετήθηκε πριν από την συνάρτηση ενεργοποίησης. Τελευταίο επίπεδο, του πρώτου στρώματος είναι το pooling layer όπου χρησιμοποιήσαμε την max pooling συνάρτηση για την σταδιακή μείωση των διαστάσεων της

εικόνας, και κατά συνέπεια την μείωση της υπολογιστικής πολυπλοκότητας του νευρωνικού δικτύου.

Χρησιμοποιήσαμε pool size (2,2) μειώνοντας τις διαστάσεις της εικόνας στο μισό (σύνηθες επιλογή). Η έξοδος του pooling layer του πρώτου στρώματος τροφοδοτείται ως είσοδος του συνελικτικού επιπέδου του δεύτερου στρώματος. Η δομή του δεύτερου στρώματος είναι ακριβώς η ίδια με του πρώτου. Έτσι σαν έξοδο από αυτό το επίπεδο και σαν είσοδο του συνελικτικού επιπέδου του τρίτου στρώματος παίρνουμε την έξοδο του πρώτου στρώματος μειωμένη στο μισό. Τα επόμενα τρία στρώματα του συνελικτικού νευρωνικού δικτύου έχουν ακριβώς την ίδια δομή με τα δυο προηγούμενα, δηλαδή αποτελούνται από το συνελικτικό επίπεδο, το επίπεδο του batch normalization, την συνάρτηση ενεργοποίησης ReLU και το επίπεδο max pooling με την μόνη διαφορά ότι στα συνελικτικά επίπεδα των στρωμάτων αυτών χρησιμοποιούμε kernel μεγέθους 3x3 αντί για 5x5 που είχαμε στα συνελικτικά επίπεδα των δυο πρώτων στρωμάτων και επίσης το pool size που χρησιμοποιούμε στα pooling layers των στρωμάτων αυτών είναι (1,2) αντί για (2,2) που ήταν το αντίστοιχο στα δυο πρώτα επίπεδα του συνελικτικού νευρωνικού δικτύου. Η τελική έξοδος από τα πέντε στρώματα του συνελικτικού νευρωνικού δικτύου είναι μια ακολουθία χαρακτηριστικών μεγέθους 32x256. Η χρήση του συνελικτικού νευρωνικού δικτύου γίνεται για την εξαγωγή χαρακτηριστικών από τα δεδομένα. Η έξοδος του συνελικτικού νευρωνικού δικτύου οδηγείται στην είσοδο του επαναλαμβανόμενου νευρωνικού δικτύου. Προτού όμως συνεχίσουμε, ένα εύλογο ερώτημα είναι πώς καταλήξαμε σε αυτές τις τιμές για το μέγεθος των φίλτρων ή αλλιώς kernel. Όπως είδαμε χρησιμοποιούμε φίλτρα διαστάσεων 5x5 στα δύο πρώτα συνελικτικά επίπεδα και φίλτρα διαστάσεων 3x3 στα τρία τελευταία συνελικτικά επίπεδα. Για αρχή να πούμε ότι η επιλογή φίλτρων μικρότερων διαστάσεων από το μέγεθος της εικόνας είναι σκόπιμη καθώς επιτρέπει στο φίλτρο να πολλαπλασιαστεί με την εικόνα πολλές φορές σε διαφορετικά σημεία αυτής και επιπλέον μειώνει το υπολογιστικό κόστος και την κατανομή βάρους που τελικά οδηγεί σε μικρότερα βάρη για την αναπαράσταση. Η πιο δημοφιλής επιλογή για το μέγεθος των φίλτρων αυτή την στιγμή στον κόσμο του deep learning είναι τα φίλτρα μεγέθους 3x3. Για αυτό ακριβώς τον λόγο χρησιμοποιήθηκαν στα τρία τελευταία συνελικτικά επίπεδα. Γιατί όμως όχι και στα δύο πρώτα; Σε μεγάλες διαστάσεις δεδομένων προτιμώνται φίλτρα μεγέθους 5x5. Είναι το μεγαλύτερο μέγεθος φίλτρων που προτιμάται και καθώς το μέγεθος της εικόνας μας, δεν έχει μειωθεί αρκετά στα δύο πρώτα συνελικτικά επίπεδα, χρησιμοποιήσαμε το μέγεθος αυτό, ενώ στα τρία τελευταία όπου το μέγεθος της εικόνας έχει μειωθεί μέσα από την διαδικασία των pooling layers χρησιμοποιήσαμε τα δημοφιλέστερα φίλτρα μεγέθους 3x3. Πώς όμως ο κόσμος του deep learning κατέληξε να χρησιμοποιεί φίλτρα αυτών των δύο διαστάσεων; Το 2012, παρουσιάστηκε η αρχιτεκτονική AlexNet CNN [8], όπου χρησιμοποιούσε kernels μεγέθους 11x11 τα οποία κατανάλωναν δύο έως και τρείς βδομάδες για την εκπαίδευση του νευρωνικού δικτύου. Επομένως, λόγω του πολύ μεγάλου χρόνου εκπαίδευσης που καταναλώνεται και της ακρίβειας, δεν χρησιμοποιούνται πλέον kernels τόσο μεγάλου μεγέθους. Γενικά, είναι προτιμότερη η χρήση kernels όπου οι διαστάσεις τους είναι περιττοί αριθμοί (3x3,5x5) διότι είναι συμμετρικά ως προς το κέντρο. Για αυτό τον λόγο αποφεύγεται η χρήση kernels διαστάσεων 2x2 και 4x4 ενώ έχει σταματήσει και η χρήση kernels 1x1 λόγω του ότι δεν προσφέρουν πληροφορίες για τα γειτονικά pixels της εικόνας [49]. Τέλος, η επιλογή των πέντε συνελικτικών επιπέδων έγινε διότι προσθέτοντας κι άλλο ένα συνελικτικό επίπεδο η συμπίεση της εικόνας ήταν αρκετή ώστε το νευρωνικό δίκτυο να έχει χαμηλότερη απόδοση. Είχαμε μείνει στην έξοδο του πέμπτου και τελευταίου στρώματος του συνελικτικού επιπέδου όπου τροφοδοτείται ως

είσοδος στο επαναλαμβανόμενο νευρωνικό δίκτυο. Αυτό αποτελείται από δυο LSTM επίπεδα 256 μονάδων το καθένα, τα οποία δημιουργούν ένα αμφίδρομο RNN. Τα δυο LSTM επίπεδα διαδίδουν πληροφορίες μέσω της ακολουθίας και χαρτογραφούν την ακολουθία σε έναν πίνακα. Παίρνουμε σαν έξοδο δυο ακολουθίες, μια από το κάθε ένα LSTM επίπεδο και τέλος συνενώνουμε αυτές τις δυο εξόδους. Η τελική έξοδος του επαναλαμβανόμενου νευρωνικού δικτύου είναι ένας πίνακας μεγέθους 32x80. Κάθε στοιχείο του πίνακα αντιπροσωπεύει μια βαθμολογία για έναν από τους 80 χαρακτήρες σε κάθε ένα από τα 32 χρονικά βήματα. Τέλος, η έξοδος του RNN οδηγείται στην είσοδο του CTC (Connectionist Temporal Classification) επιπέδου. Το επίπεδο αυτό αποτελείται από την loss fuction όπου υπολογίζει την τιμή απώλειας όταν εκπαιδεύουμε το νευρωνικό δίκτυο και τον decoder (αποκωδικοποιητή) όπου αποκωδικοποιεί τον πίνακα στο τελικό κείμενο. Θα αναφερθούμε αναλυτικότερα στο συγκεκριμένο επίπεδο στην επόμενη παράγραφο που θα μιλήσουμε για την εκπαίδευση του νευρωνικού δικτύου.

Μέρος ΙΙ: Συμβολή διατριβής

Κεφάλαιο 2ο

2 Υλοποίηση του Ταξινομητή Εικόνων 10 Τάξεων

Αρχικώς, στο παρόν το κεφάλαιο περιγράφεται η υλοποίηση TensorFlow 2.x Keras API. Το σύνολο δεδομένων που θα δουλέψουμε είναι το σύνολο δεδομένων Cifar10, ένα σύνολο δεδομένων εικόνων από 10 διαφορετικές κατηγορίες και θα χρησιμοποιήσουμε ένα Sequential CNN για να προσδιορίσουμε την κλάση μιας εικόνας.

Αυτό το μοντέλο φτάνει ~ 80% ακρίβεια.

Για να τρέχει γρήγορα το παράδειγμα θα πρέπει να πάτε στο μενού `Runtime` και να επιλέξετε `Change runtime type`. Στο παράθυρο που εμφανίζεται θα πρέπει να επιλέξετε ως `Hardware Accelerator` το `GPU`.

In [1]:

```
import tensorflow as tf
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import itertools

print(tf.__version__)
2.5.0
```

2.1 Προεπεξεργασία δεδομένων¶

Πριν δημιουργήσουμε οποιοδήποτε μοντέλο μηχανικής μάθησης, είναι σημαντικό να προεπεξεργαστούμε τα δεδομένα. Στην πραγματικότητα, η προεπεξεργασία των δεδομένων είναι το βήμα που διαρκεί περισσότερο από όλα σε οποιονδήποτε πρόβλημα μηχανικής μάθησης. Στη συνέχεια παρουσιάζονται τα βήματα για την προεπεξεργασία του συνόλου δεδομένων CIFAR10 προκειμένου να το χρησιμοποιήσουμε για το σκοπό που θέλουμε.

Λήψη δεδομένων

Το πρώτο βήμα είναι να πάρουμε τα δεδομένα και να τα χωρίσουμε σε ένα σύνολο δεδομένων εκπαίδευσης και σε ένα σύνολο δεδομένων ελέγχου. Το σύνολο δεδομένων CIFAR10 μπορεί να ληφθεί απευθείας από τη βιβλιοθήκη TensorFlow της Python. Σε αυτή του την έκδοση το σύνολο δεδομένων έχει ήδη διαιρεθεί στα σύνολα εκπαίδευσης κι ελέγχου.

Το `x_train` είναι το σύνολο έγχρωμων εικόνων, 32x32, από αντικείμενα στα οποία θα εκπαιδευτεί το μοντέλο.

Το `y_train` είναι το σύνολο ετικετών που αντιστοιχούν στις εικόνες του `x_train`.

Το `x_test` είναι το σύνολο έγχρωμων εικόνων, 32x32, από αντικείμενα στα οποία θα δοκιμαστεί το μοντέλο.

Το `y_test` είναι το σύνολο των ετικετών που αντιστοιχούν στις εικόνες του `x_test`.

Η Tensorflow είναι μια βιβλιοθήκη που χρησιμοποιείται για την ανάπτυξη εφαρμογών βαθιάς μάθησης με χρήση τεχνητών νευρωνικών δικτύων. Στην Tensorflow όλοι οι υπολογισμοί γίνονται με τη βοήθεια των τανυστών (tensors). Ένας τανυστής είναι ένα διάνυσμα ή μήτρα ν-διαστάσεων που αντιπροσωπεύει όλους τους τύπους δεδομένων. Αναλυτική περιγραφή των τανυστών είναι διαθέσιμη στη διεύθυνση https://www.guru99.com/tensor-tensorflow.html. Η Keras είναι ένα API βαθιάς μάθησης που τρέχει πανω από την Tensorflow, το οποίο διατίθεται στους προγραμματιστές για την ανάπτυξη εφαρμογών βαθιάς μάθησης.

In [2]:
```
# Download CIFAR10 dataset
cifar10 = tf.keras.datasets.cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-
python.tar.gz
170500096/170498071 [==============================] - 7s 0us/step
```

In [3]:
```
# Create a copy of y_train, flattened to one dimension
y_train = y_train.flatten()
# Create a copy of y_test, flattened to one dimension
y_test = y_test.flatten()
```

Οι κατηγορίες των εικόνων είναι 10:

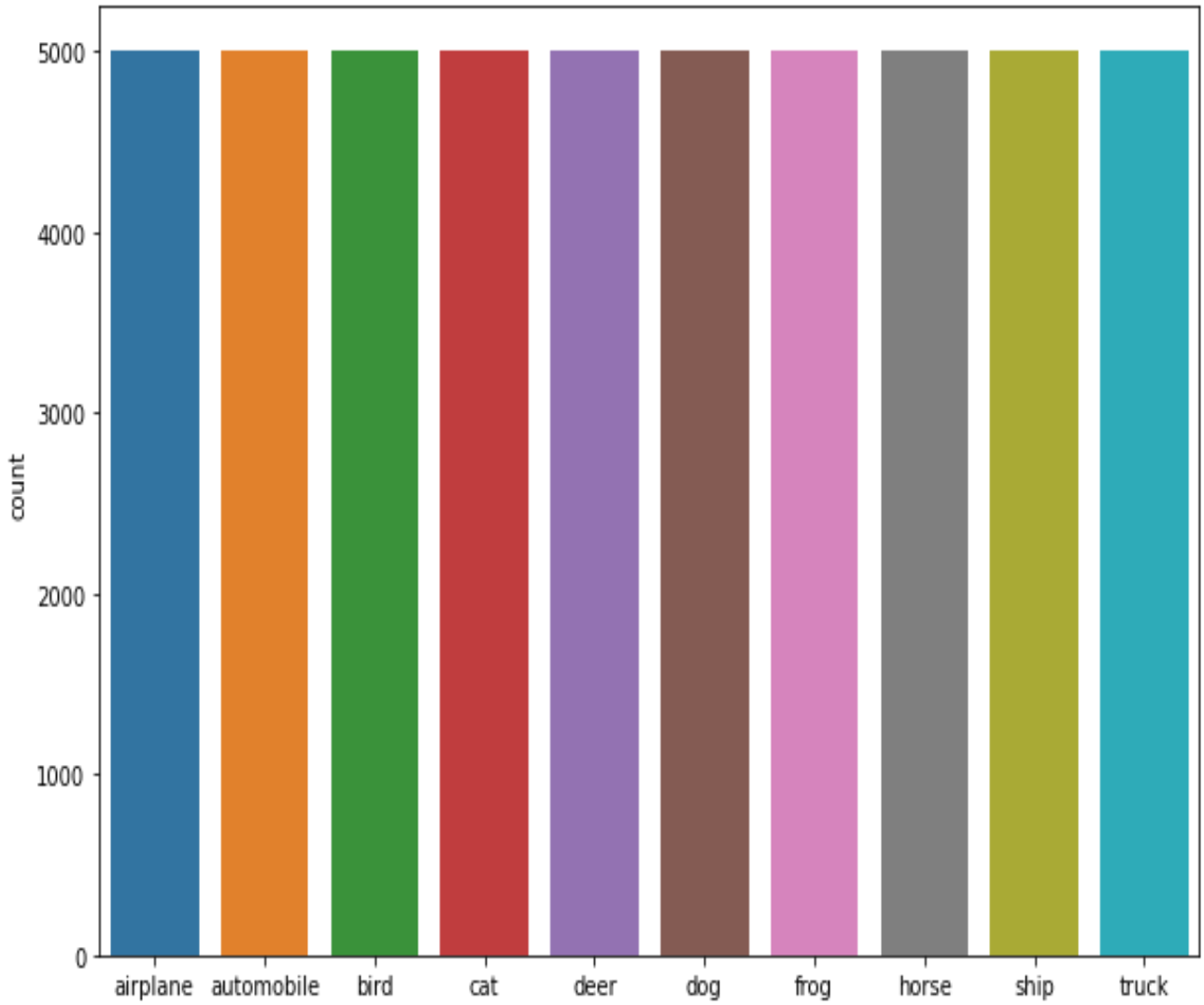
```
0. airplane
1. automobile
2. bird
3. cat
4. deer
5. dog
6. frog
7. horse
8. ship
9. truck
```

Μπορούμε να δούμε πόσες εικόνες υπάρχουν σε κάθε κατηγορία φτιάχνοντας ένα countplot με τις τιμές του συνόλου `y_train`. Βλέπουμε ότι κάθε κατηγορία έχει 5000 εικόνες. Η ομοιόμορφη κατανομή εικόνων είναι χρήσιμη για το μοντέλο μας, καθώς θα έχει αρκετές εικόνες για να μάθει τα χαρακτηριστικά κάθε κατηγορίας.

In [4]:
```
classes = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog',
'horse', 'ship', 'truck']

plt.figure(figsize=(10,7))
#Show the counts of observations in each categorical bin using bars
p = sns.countplot(y_train)
#Set the x axis' labels with list of string labels contained in classes
list
```

```
p.set(xticklabels=classes)
plt.show()
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  FutureWarning
```



Έλεγχος ύπαρξης τιμών NaN

In [5]:

```
#Test x_train element-wise for NaN and return result as a boolean array.
If any of the elements of the array true (i.e. there is a NaN value)
return True, else return False
np.isnan(x_train).any()
```

Out[5]:

```
False
```

In [6]:

```
#Test y_train element-wise for NaN and return result as a boolean array.
If any of the elements of the array true (i.e. there is a NaN value)
return True, else return False
np.isnan(x_test).any()
```

Out[6]:

```
False
```

Δεν υπάρχουν τιμές NaN στο σύνολο δεδομένων μας. Συνεπώς, δεν υπάρχει ανάγκη προεπεξεργασίας των δεδομένων για τη διαχείριση τιμών NaN.

Κανονικοποίηση (normalization)

Δεδομένου ότι το σύνολο δεδομένων `x_train` παριέχει έγχρωμες εικόνες 32x32, το σχήμα εισόδου μας πρέπει να καθοριστεί έτσι ώστε το μοντέλο μας να γνωρίζει τι είσοδο να αναμένει.

Το πρώτο επίπεδο συνελίξεων αναμένει έναν μόνο τανυστή (tensor) διάστασης 50000x32x32x3 αντί για 50000 τανυστές διάστασης 32x32x3.

Όπως έχουμε αναφέρει και σε προηγούμενο εργαστήριο, τα μοντέλα λειτουργούν γενικά καλύτερα όταν τους δίνουμε ως είσοδο κανονικοποιημένες τιμές. Ο καλύτερος τρόπος για την κανονικοποίηση των δεδομένων εξαρτάται από το συγκεκριμένο σύνολο δεδομένων που έχουμε στη διάθεσή μας. Για το σύνολο δεδομένων CIFAR10, θέλουμε κάθε τιμή να κυμαίνεται μεταξύ 0 και 1. Καθώς όλες οι τιμές αρχικά εμπίπτουν στο εύρος 0-255, για να το πετύχουμε αυτό θα πρέπει να διαιρέσουμε όλες τις τιμές με 255.0.

In [7]:

```
input_shape = (32, 32, 3)

x_train=x_train.reshape(x_train.shape[0], x_train.shape[1],
x_train.shape[2], 3)
x_train=x_train / 255.0
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2],
3)
x_test=x_test / 255.0
```

Κωδικοποίηση ετικετών κατηγοριών

Οι ετικέτες των κατηγοριών στα σύνολα εκπαίδευσης κι ελέγχου είναι κατηγορικές και όχι συνεχείς. Για να συμπεριλάβουμε κατηγορημικά δεδομένα στο μοντέλο μας, οι ετικέτες μας θα πρέπει να μετατραπούν σε μια νέα μορφή που είναι γνωστή ως one-hot encoding. Σύμφωνα με αυτή, η κάθε μία από τις ετικέτες μας θα μετατραπούν σε μια σειρά από 10 bits (όσες είναι και οι κατηγορίες μας), στην οποία σειρά όλα τα bits θα είναι 0 εκτός από το bit που βρίσκεται στη θέση `n` όπου `n` είναι ο αριθμός που αντιστοιχεί στην κατηγορία.

Για παράδειγμα, το `2` (bird) γίνεται `[0,0,1,0,0,0,0,0,0,0]` και το `7` (horse) γίνεται `[0,0,0,0,0,0,1,1,0,0]`.

In [8]:

```
y_train = tf.one_hot(y_train.astype(np.int32), depth=10)
y_test = tf.one_hot(y_test.astype(np.int32), depth=10)
```

In [9]:

```
y_train[0]
```

Out[9]:

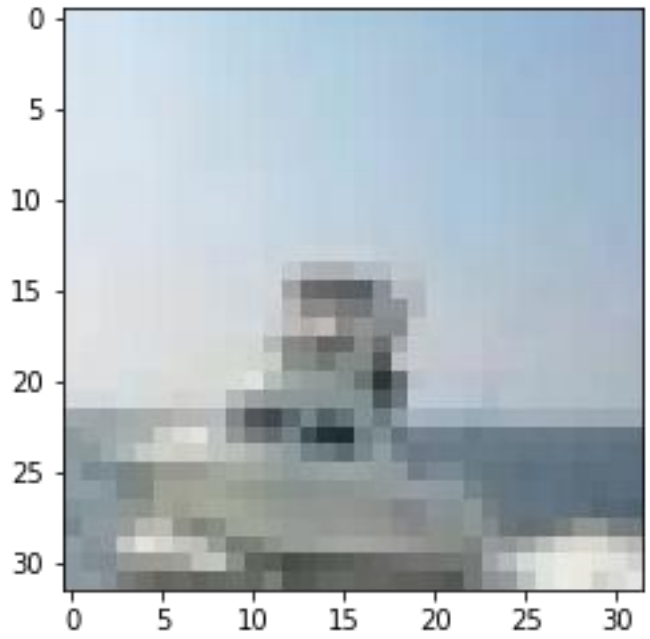
```
<tf.Tensor: shape=(10,), dtype=float32, numpy=array([0., 0., 0., 0., 0.,
0., 1., 0., 0., 0.], dtype=float32)>
```

Οπτικοποίηση δεδομένων

Όποτε χρειαστεί μπορούμε να οπτικοποιήσουμε μια εικόνα που βρίσκεται σε κάποια θέση του συνόλου δεδομένων `x_train`. Π.χ. για να δούμε την 100η εικόνα του συνόλου δεδομένων μας εκτελούμε τον ακόλουθο κώδικα.

In [10]:

```
plt.imshow(x_train[100])
print(y_train[100])
tf.Tensor([0. 0. 0. 0. 0. 0. 0. 0. 1. 0.], shape=(10,), dtype=float32)
```



Η εικόνα αυτή απεικονίζει ένα πλοίο. Το διάνυσμα που αντιστοιχεί στην κατηγορία μετά από την εφαρμογή του one-hot encoding κρατά την τιμή της κατηγορίας "πλοίο" (που είναι `8`).

Κατασκευή CNN

Τώρα είμαστε έτοιμοι να φτιάξουμε το μοντέλο του CNN.

Ορισμός μοντέλου

Το μοντέλο που θα ορίσουμε περιέχει διάφορα επίπεδα, τα οποία στοιβάζονται το ένα πάνω στο άλλο. Η έξοδος ενός επιπέδου τροφοδοτεί την είσοδο του επόμενου επιπέδου.

Τα επίπεδα Conv2D είναι συνελίξεις. Κάθε φίλτρο (32 στα πρώτα δύο στρώματα συνελίξεων και 64 στα επόμενα δύο στρώματα συνελίξεων) μετατρέπει ένα μέρος της εικόνας (5x5 για τα δύο πρώτα στρώματα Conv2D και 3x3 για τα επόμενα δύο επίπεδα Conv2D). Ο μετασχηματισμός εφαρμόζεται σε ολόκληρη την εικόνα.

Το MaxPool2D είναι ένα φίλτρο δειγματοληψίας (max-pooling). Χωρίζει την εικόνα σε μήτρες 2x2. Σε κάθε μια από αυτές τις μήτρες βρίσκει το pixel με τη μέγιστη τιμή και κρτατάει μόνο αυτό. Το φίλτρο στοχεύει στη διατήρηση των κύριων χαρακτηριστικών της εικόνας μειώνοντας παράλληλα το μέγεθός της.

Το Dropout είναι ένα επίπεδο κανονικοποίησης. Στο μοντέλο μας, το 25% των κόμβων του επιπέδου αγνοείται τυχαία, επιτρέποντας στο δίκτυο να μάθει διαφορετικά χαρακτηριστικά. Αυτό αποτρέπει το υπερπροσαρμογή δεδομένων (overfitting).

Το `Relu` είναι μια λειτουργία ενεργοποίησης ανορθωτή, δηλαδή μια συνάρτηση ενεργοποίησης (activation function) που χρησιμοποιείται για την εύρεση μη γραμμικότητας στα δεδομένα. Λειτουργεί επιστρέφοντας την τιμή εισόδου εάν η τιμή εισόδου είναι >=0. Εάν η είσοδος είναι αρνητική, επιστρέφει 0.

Το Flatten μετατρέπει τους τανυστές σε μονοδιάστατα διανύσματα.

Τα πυκνά στρώματα (Dense layers) είναι τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks - ANNs). Το τελευταίο επίπεδο επιστρέφει την πιθανότητα μια εικόνα να ανήκει σε κάθε μία από τις κατηγορίες. Για το σκοπό αυτό χρησιμοποιείται η συνάρτηση softmax.

In [11]:

```
num_classes = 10

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, 5, padding='same',
input_shape=x_train.shape[1:], activation='relu'),
    tf.keras.layers.Conv2D(32, 5, activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Dropout(0.25),

    tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'),
    tf.keras.layers.Conv2D(64, 3, activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Dropout(0.25),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(num_classes, activation='softmax'),
])
```

Μετά τη δημιουργία του μοντέλου το παραμετροποιούμε. Καθώς αυτό το μοντέλο στοχεύει στην κατηγοριοποίηση των εικόνων, θα χρησιμοποιήσουμε ως συνάρτηση απώλειας (loss function) την "categorical_crossentropy " (περισσότερες πληροφορίες γι αυτή τη συνάρτηση απώλειας μπορείτε

να βρείτε στη διεύθυνση https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy). Ενώ ωε μετρική για την αξιολόγηση της απόδοσής του θα ορίσουμε το accuracy, δηλαδή την ακρίβεια με την οποία το μοντέλο κάνει προβλέψεις (μετρά πόσες φορές έχει γίνει σωστή πρόβλεψη).

In [12]:
```
model.compile(optimizer=tf.keras.optimizers.RMSprop(learning_rate=0.0001,
decay=1e-06),
              loss='categorical_crossentropy', metrics=['acc'])
```

Εκπαίδευση του μοντέλου

Πριν εκπαιδεύσουμε το μοντέλο θα πρέπει να ορίσουμε τιμές για κάποιες παραμέτρους. Συγκεκριμένα, θα πρέπει να ορίσουμε το `batch_size`, το οποίο δηλώνει αν θα πάρουμε όλα τα δεδομένα μαζεμένα (`batch_size`= πλήθος δεδομένων εκπαίδευσης) ή θα τα πάρουμε κατά τμήματα μεγέθους `batch_size` το καθένα. Επίσης, θα πρέπει να οριστεί πόσες φορές θα χρησιμοποιηθεί το σύνολο των δεδομένων για την ακπαίδευση (epochs).

In [13]:
```
batch_size = 32
epochs = 50
```

Ο έλεγχος του μοντέλου με χρήση των δεδομένων ελέγχου αποτρέπει την υπερ-προσαρμογή των δεδομένων (overfitting). Στο παράδειγμά μας το 90% του συνόλου δεδομένων χρησιμοποιείται για την εκπαίδευση του μοντέλου, ενώ το υπόλοιπο 10% χρησιμοποιείται για τον έλεγχό του.

Προχωράμε λοιπόν με την εκπαίδευση του μοντέλου.

In [14]:
```
  history = model.fit(x_train, y_train, batch_size=batch_size,
                      epochs=epochs)
Epoch 1/50
1563/1563 [==============================] - 40s 5ms/step - loss: 1.7942 -
acc: 0.3402
Epoch 2/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.4891 -
acc: 0.4588
Epoch 3/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.3526 -
acc: 0.5127
Epoch 4/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.2520 -
acc: 0.5555
Epoch 5/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.1635 -
acc: 0.5890
Epoch 6/50
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0951 -
acc: 0.6108
Epoch 7/50
```

```
1563/1563 [==============================] - 8s 5ms/step - loss: 1.0418 -
acc: 0.6324
Epoch 8/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.9902 -
acc: 0.6512
Epoch 9/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.9509 -
acc: 0.6661
Epoch 10/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.9147 -
acc: 0.6788
Epoch 11/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.8901 -
acc: 0.6877
Epoch 12/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.8591 -
acc: 0.7000
Epoch 13/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.8353 -
acc: 0.7085
Epoch 14/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.8089 -
acc: 0.7204
Epoch 15/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7976 -
acc: 0.7228
Epoch 16/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7753 -
acc: 0.7318
Epoch 17/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7601 -
acc: 0.7366
Epoch 18/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7488 -
acc: 0.7402
Epoch 19/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7388 -
acc: 0.7435
Epoch 20/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7268 -
acc: 0.7484
Epoch 21/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7182 -
acc: 0.7530
Epoch 22/50
```

```
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7120 -
acc: 0.7546
Epoch 23/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.7017 -
acc: 0.7587
Epoch 24/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6974 -
acc: 0.7619
Epoch 25/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6896 -
acc: 0.7645
Epoch 26/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6850 -
acc: 0.7665
Epoch 27/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6811 -
acc: 0.7662
Epoch 28/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6773 -
acc: 0.7710
Epoch 29/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6739 -
acc: 0.7685
Epoch 30/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6689 -
acc: 0.7744
Epoch 31/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6657 -
acc: 0.7727
Epoch 32/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6600 -
acc: 0.7761
Epoch 33/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6612 -
acc: 0.7771
Epoch 34/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6629 -
acc: 0.7779
Epoch 35/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6510 -
acc: 0.7793
Epoch 36/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6504 -
acc: 0.7816
Epoch 37/50
```

```
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6461 -
acc: 0.7827
Epoch 38/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6455 -
acc: 0.7822
Epoch 39/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6442 -
acc: 0.7840
Epoch 40/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6466 -
acc: 0.7823
Epoch 41/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6401 -
acc: 0.7851
Epoch 42/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6406 -
acc: 0.7864
Epoch 43/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6425 -
acc: 0.7847
Epoch 44/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6388 -
acc: 0.7843
Epoch 45/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6379 -
acc: 0.7853
Epoch 46/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6360 -
acc: 0.7883
Epoch 47/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6360 -
acc: 0.7874
Epoch 48/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6274 -
acc: 0.7900
Epoch 49/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6324 -
acc: 0.7892
Epoch 50/50
1563/1563 [==============================] - 8s 5ms/step - loss: 0.6297 -
acc: 0.7906
```

Αξιολόγηση Μοντέλου

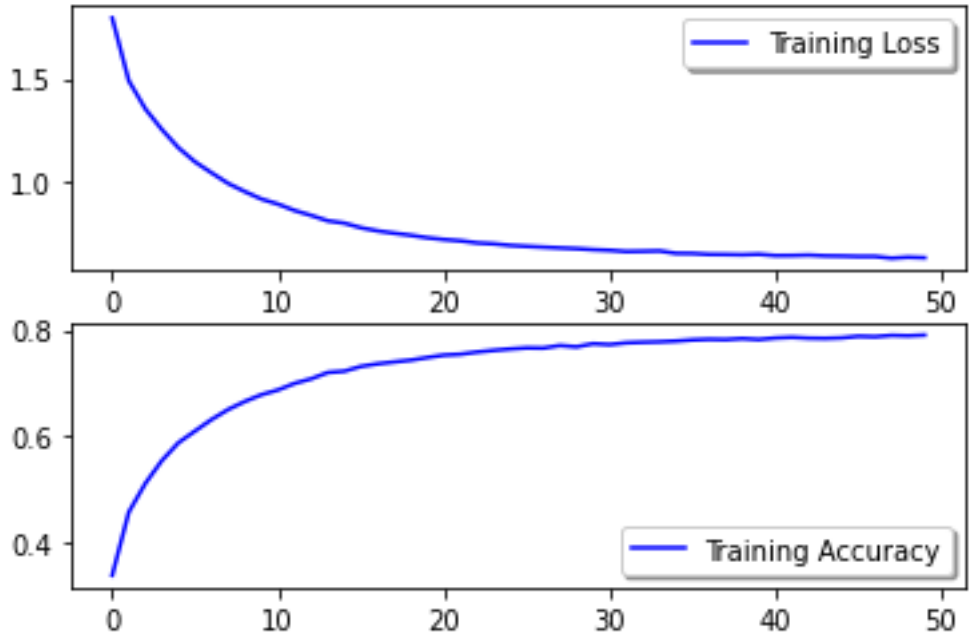
Καμπύλες απώλειας και ακρίβειας

Με βάση τα αποτελέσματα που προέκυψαν παραπάνω σχεδιάζουμε τις καμπύλες που αντιστοιχούν στο loss και στο acc του μοντέλου μας προκειμένου να αξιολογήσουμε την απώλεια και την ακρίβειά του.

In [15]:

```
fig, ax = plt.subplots(2,1)
ax[0].plot(history.history['loss'], color='b', label="Training Loss")
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(history.history['acc'], color='b', label="Training Accuracy")
legend = ax[1].legend(loc='best', shadow=True)
```



Παρατηρούμε ότι η ακρίβεια του μοντέλου αυξάνεται με την πάροδο του χρόνου και η απώλειά του μειώνεται με την πάροδο του χρόνου. Γενικά, θα πρέπει να είμαστε πολύ προσεκτικλη με την επιλογή της τιμής για τα epochs αφού η εκτέλεση του μοντέλου για περισσότερα epochs ενδέχεται να προκαλέσει την υπερβολική προσαρμογή του μοτνέλου στα δεδομένα μας (overfitting).

Χρήση του συνόλου ελέγχου για τη διενέργεια προβλέψεων

Στη συνέχεια χρησιμοποιούμε το μοντέλο μας για να κάνουμε προβλέψεις για τα δεδομένα ελέγχου προκειμένου να αξιολογήσουμε την ακρίβειά του.

In [16]:

```
test_loss, test_acc = model.evaluate(x_test, y_test)
313/313 [==============================] - 1s 3ms/step - loss: 0.7818 -
acc: 0.7629
```

Το μοντέλο μας λειτουργεί αρκετά καλά, με ακρίβεια ~ 80% στα δεδομένα ελέγχου.

Μήτρα Σύγχυσης (Confusion Matrix)

Στο σημείο αυτό θα φτιάξουμε τη μήτρα σύγχυσης (confusion matrix), η οποία δείχνει τον αριθμό των σωστών και λανθασμένων προβλέψεων που γίνονται από το μοντέλο ταξινόμησης (χρησιμοποιώντας τα πραγματικά αποτελέσματα του συνόλου ελέγχου). Περιέχει επομένως πληροφορίες, σχετικά με την πραγματική και την προβλεπόμενη ταξινόμηση. Οι βέλτιστες λύσεις του μοντέλου έχουν μηδενικές λύσεις περιμετρικά από την κύρια διαγώνιο της μήτρας σύγχυσης, ενώ στην κύρια διαγώνιο της μήτρας εμφανίζονται τα ορθά στοιχεία ταξινόμησης. Για να φτιάξουμε τη μήτρα σύγχυαης χρησιμοποιούμε την βιβλιοθήκη TensorFlow της Python.

In [17]:

```
# Predict the values from the validation dataset
y_pred = model.predict(x_test)
# Convert predictions classes to one hot vectors
y_pred_classes = np.argmax(y_pred,axis = 1)
# Convert validation observations to one hot vectors
y_true = np.argmax(y_test,axis = 1)
# compute the confusion matrix
confusion_mtx = tf.math.confusion_matrix(y_true, y_pred_classes)
```

Στη συνέχεια σχεδιάζουμε τη μήτρα σύγχυσης. Βλέπουμε ότι το μοντέλο μας ταξινομεί τους βατράχους αρκετά καλά, με 906 από τις 1000 εικόνες βατράχων να ταξινομούνται σωστά. Μπορούμε επίσης να δούμε ότι υπάρχει σχετικά υψηλή σύγχυση ανάμεσα στις γάτες και τους σκύλους.

In [18]:

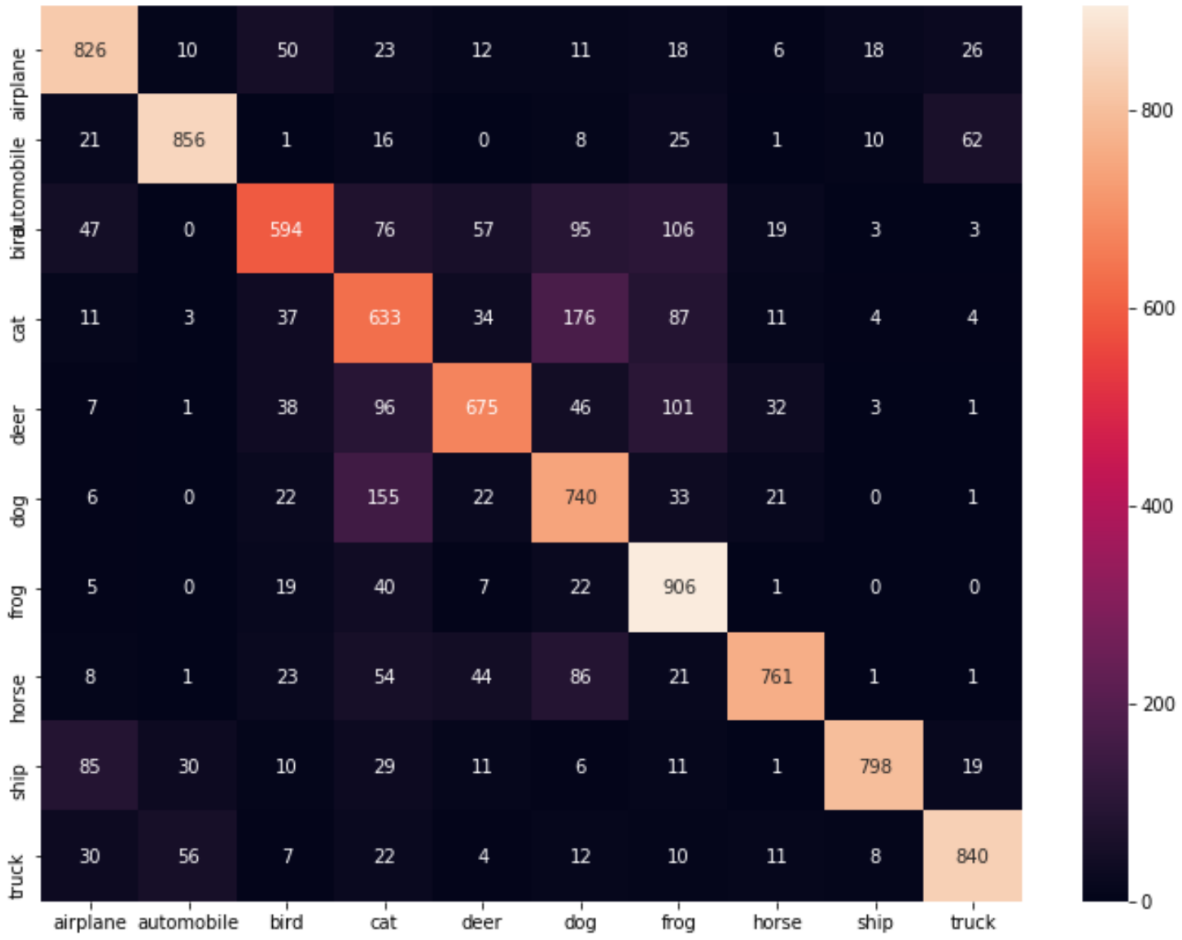
```
plt.figure(figsize=(12, 9))
c = sns.heatmap(confusion_mtx, annot=True, fmt='g')
c.set(xticklabels=classes, yticklabels=classes)
```

Out[18]:

```
[[Text(0, 0.5, 'airplane'),
  Text(0, 1.5, 'automobile'),
  Text(0, 2.5, 'bird'),
  Text(0, 3.5, 'cat'),
  Text(0, 4.5, 'deer'),
  Text(0, 5.5, 'dog'),
  Text(0, 6.5, 'frog'),
  Text(0, 7.5, 'horse'),
  Text(0, 8.5, 'ship'),
  Text(0, 9.5, 'truck')],
 [Text(0.5, 0, 'airplane'),
  Text(1.5, 0, 'automobile'),
  Text(2.5, 0, 'bird'),
  Text(3.5, 0, 'cat'),
  Text(4.5, 0, 'deer'),
  Text(5.5, 0, 'dog'),
  Text(6.5, 0, 'frog'),
  Text(7.5, 0, 'horse'),
```

```
Text(8.5, 0, 'ship'),
Text(9.5, 0, 'truck')]]
```

.



Κεφάλαιο 6ο

3 Συμπεράσματα

4 Βιβλιογραφικές Πηγές

1. **[Ηλεκτρονικό] https://archive.ics.uci.edu/ml/machine-learning-databases/kddcup99-mld/kddcup99.html.**
2. Roiger R.G., Geatz M.W. *"Εξόρυξη Πληροφορίας – Ένας Εισαγωγικός Οδηγός με Παραδείγματα". Εκδόσεις Κλειδάριθμος.* s.l. : Εκδόσεις Κλειδάριθμος, 2008.
3. Lee W., & Stolfo, S. J. *A framework for constructing features and models for intrusion detection systems. ACM transactions on Information and system security (TiSSEC), 3(4).* 2000. σσ. 227-261.
4. Smyth, Stephen D. Bay and Dennis F. Kibler and Michael J. Pazzani and Padhraic. The UCI KDD Archive of Large Data Sets for Data Mining Research and Experimentation. SIGKDD Explorations, 2. 2000.
5. Breiman, L. *Bagging Predictors. Machine Learning.* 1996. σσ. 123-140.
6. Cox, E. Free-Form Text Data Mining Integrating Fuzzy Systems, Self-Organizing Neural Nets and Rule-Based Knowledge Bases. *PC AI.* 2000, September-October, σσ. 22-26.
7. Chester, M. *Neural Networks—A Tutorial.* Upper Saddle River, NJ : PTR Prentice Hall, 1993.
8. Pang-Ning Tan, Michael Steinbach, Vipin Kumar. *Εισαγωγή στην Εξόρυξη Δεδομένων, Εκδόσεις Τζιόλα.* s.l. : Εκδόσεις Τζιόλα, 2018.
9. Foster Provost, Tom Fawcet. *Η Επιστήμη των Δεδομένων για Επιχειρήσεις, .* s.l. : Εκδόσεις Κλειδάριθμος, 2019.
10. Grus, Joel. *Επιστήμη Δεδομένων: Βασικές Αρχές και Εφαρμογές με Python, 2η έκδοση.* s.l. : Εκδόσεις Παπασωτηρίου, 2021.
11. Mohammed J. Zaki, Wagner Meira Jr. *Εξόρυξη και Ανάλυση Δεδομένων: Βασικές Έννοιες και Αλγόριθμοι.* s.l. : Εκδόσεις Κλειδάριθμος., 2017.
12. Anand Rajaraman, Jeffrey David Ullman. *Εξόρυξη από Μεγάλα Σύνολα Δεδομένων.* s.l. : Εκδόσεις Νέων Τεχνολογιών, 2013.
13. Janert, Philipp K. *Data Analysis with Open Source Tools.* s.l. : O'Reilly Press, 2011.
14. Gregory Koch, Richard Zemel, Ruslan Salakhutdinov. Siamese Neural Networks for One-shot Image Recognition. *Department of Computer Science, University of Toronto. Toronto, Ontario, Canada.*