Χαροκόπειο Πανεπιστήμιο

Τμήμα Πληροφορικής & Τηλεματικής

Πτυχιακή εργασία

Τίτλος: Υπηρεσία ιστού για αναγνώριση χαρακτήρων με χρήση μεθόδων μηχανικής μάθησης.

Ηλίας Οουές
Α.Μ.: 20922

Τριμελής Επιτροπή:

κ. Νικολαΐδου Μαρία

κ. Δημητρακόπουλος Δημήτριος

κ. Βαρλάμης Ηρακλής

Αθήνα 2016

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ 6
ABSTRACT 7
1 ΕΙΣΑΓΩΓΗ 8
1.1 Ιστορικά Στοιχεία 8
1.2 Μηχανική Μάθηση 10
1.3 Διαδικτυακές Υπηρεσίες 11
1.4 Δομή Εργασίας 12
2 ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ 13
2.1 Μηχανική Μάθηση 13
2.2 Τρόποι μηχανικής μάθησης 15
2.3 Επιτηρούμενη Μάθηση 15
2.3.1 Ανταλλαγή προκατάληψης-διαφορετικότητας (Bias-variance tradeoff) 17
2.3.2 Πολυπλοκότητας συνάρτησης και ποσό εκπαιδευμένων δεδομένων (Function complexity and amount of training) 17
2.3.3 Διαστατικότητα του διαστήματος του ορίσματος (Dimentionality of the input space) 18
2.3.4 Θόρυβος στις τιμές του παραγώγου (Noise in the output values) 18
2.4 Μη επιτηρούμενη μάθηση (Unsupervised learning) 19
2.5 Δενδροδιαγράμματα (Decision Trees) 21
2.5.1 Κόμβοι (nodes) και Κλάδοι (branches) 22
2.5.2 Είδη προβλημάτων μάθησης 23
2.6 Νευρωνικά Δίκτυα 27
2.7 Learning rules 29
2.7.1 Hebbian Learning Rule 29
2.7.2 The Delta Rule 29
2.7.3 Widrow-Hoff LMS Learning Rule 30
2.7.4 Διαφορές νευρωνικών δικτύων με συμβατικούς υπολογισμούς 30
2.7.5 Χρήση νευρωνικών δικτύων 31
2.8 ΛΟΓΙΣΜΙΚΟ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΜΕ OCR 31
2.8.1 Βιβλιοθήκη Tesseract 32
2.8.2 Βιβλιοθήκη OCRopus 34
2.8.3 Σύγκριση 35

3 Υπηρεσία Διαδικτύου 37
3.1 SOAP 37
3.2 REST 39
3.3 Διαφορές REST - SOAP 41
3.4 Web Application Framework-FLASK 42
4 ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΓΑΣΙΑΣ - ΥΛΟΠΟΙΗΣΗ 44
4.1 Web Service 45
4.2 Διαδικασία επεξεργασίας εικόνας και αναγνώριση χαρακτήρων 46
4.3 Training 51
4.4 Εφαρμογή κανονικών εκφράσεων 56
4.5 Δοκιμή Web Service 57
4.6 Διαδικασία εγκατάστασης Ocropus 59
4.7 Προβλήματα κατά την εγκατάσταση 59
5 ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ 60
5.1 Δυνατότητες εφαρμογής 60
5.2 Μελλοντικές επεκτάσεις 61
6 ΕΠΙΛΟΓΟΣ 61
7 ΒΙΒΛΙΟΓΡΑΦΙΑ 63

Πίνακας Εικόνων και Σχημάτων

ΕΙΚΟΝΑ 2.1: ΈΝΑ ΔΕΝΔΡΟΔΡΙΑΓΡΑΜΜΑ ΓΙΑ ΕΝΑ ΣΥΣΤΗΜΑ ΠΟΥ ΠΡΟΤΕΙΝΕΙ ΜΑΘΗΜΑΤΑ [10] 25
ΕΙΚΌΝΑ 2.2: ΙΣΤΟΓΡΑΜΜΑ ΤΑΜΠΕΛΩΝ. [10] 25
ΠΙΝΑΚΑΣ 2.1: ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ ΜΑΘΗΜΑΤΩΝ. 26
ΕΙΚΟΝΑ 2.3: ΤΕΧΝΗΤΟΣ ΝΕΥΡΩΝΑΣ. 28
ΕΙΚΌΝΑ 2.4: ΤΕΧΝΗΤΌ ΝΕΥΡΩΝΙΚΌ ΔΊΚΤΥΟ 28
ΕΙΚΟΝΑ 2.5: ΑΡΧΙΤΕΚΤΟΝΙΚΗ TESSERACT. 33
ΣΧΉΜΑ 2 1: ΒΗΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ OCROPUS. 34
ΕΙΚΌΝΑ 2.6: ΒΗΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ OCROPUS. 34
ΕΙΚΟΝΑ 2.7: ΣΥΓΚΡΙΣΗ ΜΕΤΑΞΥ TESSERACT ΚΑΙ OCROPUS. 36
ΕΙΚΟΝΑ 3.1: ΔΟΜΗ ΜΗΝΥΜΑΤΟΣ SOAP [8] 38
ΠΙΝΑΚΑΣ 3.1: ΡΗΜΑΤΑ HTTP ΚΑΙ ΧΡΗΣΗ ΤΟΥΣ ΜΕ REST. 41
ΣΧΗΜΑ 4.1: ΣΧΕΔΙΑΓΡΑΜΜΑ ΕΡΓΑΣΙΑΣ. 44
ΕΙΚΟΝΑ 4.1: ΣΚΑΝΑΡΙΣΜΕΝΗ ΦΩΤΟΓΡΑΦΙΑ ΣΤΗΝ ΟΠΟΙΑ ΤΡΕΧΟΥΜΕ OCR. 46
ΣΧΉΜΑ 4.2: ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ OCR. 46
ΣΧΉΜΑ 4.3: SCRIPT ΕΚΤΕΛΕΣΗΣ OCR. 47
ΕΙΚΌΝΑ 4.2: FLATTENED IMAGE 48
ΕΙΚΌΝΑ 4.3: BINARIZED IMAGE 48
ΕΙΚΌΝΑ 4.4: SEGMENTATED IMAGE. 49
ΕΙΚΟΝΑ 4.5: ΞΕΧΩΡΙΣΤΗ ΕΙΚΟΝΑ ΓΙΑ ΚΑΘΕ ΓΡΑΜΜΗ ΤΗΣ ΕΙΚΟΝΑΣ. 49
ΕΙΚΌΝΑ 4.6: ΑΠΟΤΕΛΕΣΜΑΤΑ OCR 50
ΕΙΚΌΝΑ 4.7: ΕΙΣΑΓΩΓΗ ΣΩΣΤΟΥ ΚΕΙΜΕΝΟΥ. 51
ΕΙΚΌΝΑ 4.8: ΠΑΡΑΔΕΙΓΜΑ ΕΚΠΑΙΔΕΥΣΗΣ ΜΟΝΤΕΛΟΥ. 53
ΕΙΚΟΝΑ 4.9: ΑΡΙΣΤΕΡΑ: ΑΡΧΙΚΗ ΕΙΚΟΝΑ, ΔΕΞΙΑ: ΕΙΚΟΝΑ ΜΕΤΑ ΤΗΝ ΕΚΠΑΙΔΕΥΣΗ 54
ΕΙΚΟΝΑ 4.10:ΑΡΙΣΤΕΡΑ: ΑΡΧΙΚΗ ΕΙΚΟΝΑ, ΔΕΞΙΑ: ΑΠΟΤΕΛΕΣΜΑ OCR 55
ΕΙΚΟΝΑ 4.11: ΑΡΙΣΤΕΡΑ: ΑΡΧΙΚΗ ΕΙΚΟΝΑ, ΔΕΞΙΑ: ΑΠΟΤΕΛΕΣΜΑ OCR 55
ΕΙΚΟΝΑ 4.12: ΑΡΙΣΤΕΡΑ: ΑΡΧΙΚΗ ΕΙΚΟΝΑ, ΔΕΞΙΑ: ΑΠΟΤΕΛΕΣΜΑ OCR 56
ΕΙΚΟΝΑ 4.13: ΔΙΕΠΑΦΗ POSTMAN. 57

ΕΙΚΟΝΑ 4.14: ΚΑΡΤΕΛΑ ΓΙΑ ΑΝΕΒΑΣΜΑ ΕΙΚΟΝΑΣ. 58
ΕΙΚΟΝΑ 4.15: ΕΠΙΛΟΓΗ ΕΙΚΟΝΑΣ. 58

ΠΕΡΙΛΗΨΗ

Σκοπός της εργασίας είναι η δημιουργία μιας υπηρεσίας ιστού για την οπτική αναγνώριση χαρακτήρων. Η αναγνώριση υποστηρίζει την Ελληνική γλώσσα. Ακόμα δίνεται η δυνατότητα εκπαίδευσης ενός μοντέλου με την χρήση μεθόδων μηχανικής μάθησης. Η οπτική αναγνώριση χαρακτήρων είναι η διαδικασία με την οποία εξάγεται το κείμενο που περιέχεται σε ένα οπτικό μέσω και συγκεκριμένα μια εικόνα που μπορεί να είναι μια φωτογραφία ή ένα σκαναρισμένο έγγραφο όπως μια σελίδα από βιβλίο. Η εργασία για την δημιουργία της υπηρεσίας ιστού κάνει χρήση του εργαλείου flask, της γλώσσας προγραμματισμού Python και των υπηρεσιών ιστού (REST web services). Για την δημιουργία της εφαρμογής για την οπτική αναγνώριση χαρακτήρων χρησιμοποιείται η βιβλιοθήκη OCRopus η οποία είναι μια βιβλιοθήκη ανοιχτού λογισμικού, υλοποιημένη με Python και δίνει στο χρήστη τα εργαλεία για να δημιουργήσει ένα μοντέλο για εκπαίδευση της εφαρμογής για μελλοντικά καλύτερα αποτελέσματα. Το σενάριο χρήσης της παρούσας εργασίας προσφέρει στο χρήστη μια πλατφόρμα μέσω της οποίας μπορεί να συλλέγει δεδομένα από μια απόδειξη λιανικής πώλησης και να έχει την δυνατότητα να τα επεξεργαστεί.

ABSTRACT

The aim of this application is the creation of a web service for the optical character recognition (OCR). The recognition supports the Greek language and provides the ability to train a model using methods of machine learning. OCR is the process with which text is extracted from an optical data and specifically a picture that can be a photograph or a scanned document like a page from a book. For the creation of the web server, the application uses the Flask framework, the Python programming language and a Rest API. For the creation of the application for the character recognition the OCRopus library is used which is an open-source software implemented with Python and it provides the user the tools to create a training model for better future results. The application offers to user a platform from where he can gather data from a receipt and use the data for his own purposes.

1 ΕΙΣΑΓΩΓΗ

Σκοπός αυτής της εργασίας είναι η παροχή υπηρεσιών OCR χρησιμοποιώντας την βιβλιοθήκη OCRopus και η δημιουργία μιας υπηρεσίας διαδικτύου (web service) για την χρήση της εφαρμογής. Η εφαρμογή χρησιμοποιείται για την εξαγωγή κειμένου από μια εικόνα. Η εικόνα μπορεί να είναι μια φωτογραφία, ένα σκαναρισμένο έγγραφο όπως μια σελίδα από ένα βιβλίο, μια απόδειξη ή ένα δημόσιο έγγραφο. Η εφαρμογή είναι προσαρμοσμένη στην ελληνική γλώσσα και παρέχει την δυνατότητα χρήσης μηχανικής μάθησης για εκπαίδευση. Σαν σενάριο χρήσης, η εργασία αυτή δέχεται φωτογραφίες από αποδείξεις και εξάγει το κείμενο όπως την τιμή, τον ΦΠΑ, την ημερομηνία, τα προϊόντα, το τηλέφωνο και τον αριθμό της ταμειακής μηχανής.

Στα πλαίσια της έρευνας αξιοποιείται η μηχανική μάθηση (machine learning), και συγκεκριμένα ερευνώνται ποιες μέθοδοι μηχανικής μάθησης υπάρχουν και είναι κατάλληλες, τι είναι τα δέντρα απόφασης (decision trees) και τα νευρωνικά δίκτυα (neural networks) καθώς και πως υποστηρίζονται από την γλώσσα προγραμματισμού Python. Ακόμα θα γίνει αναφορά στις δικτυακές υπηρεσίες (web service), ποιες αρχιτεκτονικές χρησιμοποιούνται και ποιες εφαρμογές τις υποστηρίζουν.

1.1 Ιστορικά Στοιχεία

Το **OCR** ή η οπτική αναγνώριση χαρακτήρων είναι μια τεχνολογία η οποία επιτρέπει την μετατροπή φωτογραφιών με περιεχόμενο χειρόγραφο ή τυπωμένο, από πληθώρα τύπων εγγράφων όπως σκαναρισμένο κείμενο, φωτογραφία κάποιου εγγράφου, εικόνα του περιβάλλοντος τραβηγμένη με φωτογραφική μηχανή (π.χ. πινακίδα οδικής κυκλοφορίας) ή κείμενο υποτιτλισμένο σε εικόνα σε κείμενο αναγνωρίσιμο και επεξεργάσιμο από τον υπολογιστή [24].

Ένας σκάνερ μπορεί μόνο να δημιουργήσει μια εικόνα ή ένα στιγμιότυπο του κειμένου το οποίο είναι μια συλλογή από ασπρόμαυρες ή έγχρωμες κουκίδες, γνωστό και ως

ψηφιοποιημένη εικόνα. Η τεχνολογία της οπτικής αναγνώρισης χαρακτήρων χρησιμοποιείται για την εξαγωγή και στην συνέχεια την επεξεργασία του κειμένου που βρίσκεται στην σκαναρισμένη εικόνα [3]. Το OCR χρησιμοποιείται από τον υπολογιστή για εργασίες όπως μετάφραση, κείμενο σε λόγο και εξόρυξη κειμένου και είναι επίσης και τομέας έρευνας στην αναγνώριση μοτίβου (pattern recognition) και στην τεχνητή νοημοσύνη [24]. Το OCR χρησιμοποιείται επίσης και από βιβλιοθήκες για την ψηφιοποίηση των βιβλίων ή για την καταχώρηση κειμένου από δημόσια έγγραφα όπως διαβατήρια, ταυτότητες, τραπεζικές συναλλαγές, αποδείξεις προϊόντων, επαγγελματικές κάρτες και από ηλεκτρονικό ταχυδρομείο [24].

Η αρχή αυτής της τεχνολογίας μπορεί να βρεθεί πίσω στο 1870 όταν ο C. R. Carey εφηύρε τον οπτικό σαρωτή (retina scanner) ο οποίος ήταν ένα σύστημα μετάδοσης εικόνας χρησιμοποιώντας ένα μωσαϊκό από φωτοκύτταρα. Το 1890 ο Πολωνός P. Nipkow εφηύρε τον ακολουθητικό σαρωτή (sequential scanner) ο οποίος ήταν ένα άλμα προς την κατεύθυνση της μοντέρνας τεχνολογίας, όπως η τηλεόραση. Οι πρώτες προσπάθειες για δημιουργία μηχανών υποβοήθησης τυφλών ανθρώπων έγιναν στις αρχές του 19^{ου} αιώνα με πειραματισμό του OCR ωστόσο η σύγχρονη μορφή του βρίσκεται γύρω στο 1940 με την δημιουργία του ηλεκτρονικού υπολογιστή [1].

Η πρώτη μορφή OCR χρησιμοποιήθηκε κατά το 1950 μια περίοδο στην οποίο οι υπολογιστές χρησιμοποιούσαν κάρτες για την εισαγωγή δεδομένων και η χρήση χαμηλού κόστους εφαρμογές ήταν αναγκαίες. Το πρώτο πραγματικό μηχάνημα που διάβαζε με OCR εγκαταστάθηκε στην Reader's Digest το 1954 και λειτουργούσε για να μετατρέπει τυπογραφημένες αναφορές σε κάρτες για εισαγωγή στον υπολογιστή [1].

Βασική αρχή στην αυτόματη αναγνώριση προτύπων είναι η εκμάθηση του υπολογιστή σε συγκεκριμένες κλάσεις προτύπων. Οι κλάσεις χωρίζονται σε γράμματα, νούμερα και ειδικούς χαρακτήρες όπως κόμμα, τελεία, ερωτηματικό κτλ. Για να μάθει ο υπολογιστής να αναγνωρίζει αυτές τις κλάσεις πρέπει να του δείξουμε παραδείγματα χαρακτήρων από κάθε μια κλάση [1].

Βασισμένος σε αυτά τα παραδείγματα το μηχάνημα δημιουργεί κάποιο πρωτότυπο αυτών των χαρακτήρων ώστε κατά τη διάρκεια της αναγνώρισης να συγκρίνει το κάθε δεδομένο χαρακτήρα με τους χαρακτήρες που έχει ήδη καταχωρημένο στη μνήμη με αποτέλεσμα να προσπαθεί να βρει αυτό που ταιριάζει καλύτερα [1].

1.2 Μηχανική Μάθηση

Η χρήση της μηχανικής μάθησης (machine learning) είναι σημαντική διότι καθώς το μοντέλο έρχεται σε επαφή με καινούργια δεδομένα, μπορεί να προσαρμοστεί εύκολα και γρήγορα. Η μηχανική μάθηση είναι ένα πεδίο το οποίο προσαρμόζεται και εξελίσσεται μαζί με την τεχνολογία και καθώς κάποιοι αλγόριθμοι χρησιμοποιούνται σήμερα, η ικανότητα να διαχειρίζεται μεγάλης πολυπλοκότητας δεδομένα όλο και πιο γρήγορα βρίσκεται υπό ανάπτυξη.

Ένα σύστημα μπορούμε να πούμε ότι μαθαίνει, όταν αλλάζει τη δομή του, τον προγραμματισμό του ή τα δεδομένα του με τέτοιο τρόπο ώστε να περιμένουμε μελλοντικές βελτιώσεις. Στα πλαίσια της μηχανικής μάθησης δεν μπαίνει οποιαδήποτε αλλαγή μικρής εμβέλειας όπως η αλλαγή ενός δεδομένου στον πίνακα αλλά αν για παράδειγμα η επίδοση ενός μηχανήματος αναγνώρισης φωνής αυξηθεί εφόσον ακούσει διαφορετικά πρότυπα από τη φωνή ενός ατόμου, τότε μπορούμε να πούμε ότι το μηχάνημα έμαθε [5].

Η μηχανική μάθηση αναφέρεται συνήθως σε αλλαγές συστημάτων που εκτελούν εργασίες με θέμα την τεχνητή νοημοσύνη. Κάποιες από αυτές τις εργασίες σχετίζονται με αναγνώριση, διάγνωση, σχεδίαση κ.α.. Οι αλλαγές που γίνονται μπορεί να σχετίζονται με βελτίωση συστημάτων ή σύνθεσης καινούργιων συστημάτων [5].

Υπάρχουν αρκετοί λόγοι για την χρησιμοποίηση μηχανικής μάθησης και γιατί είναι σημαντικό να υπάρχει παρά να δημιουργούνται μηχανήματα προγραμματισμένα για συγκεκριμένο σκοπό. Κάποιες εργασίες δεν μπορούν να προσδιοριστούν παρά μόνο με συγκεκριμένα παραδείγματα δηλαδή μπορούμε να ορίσουμε συγκεκριμένα ορίσματα και αποτελέσματα αλλά όχι την συγκεκριμένη σχέση μεταξύ του ορίσματος και του παραγώγου. Θα θέλαμε το μηχάνημα να μπορεί προσαρμόζει τη δομή του για να βγάζει σωστά αποτελέσματα από μεγάλο αριθμό ορισμάτων και να μπορεί αποδοτικά να προσδιορίζει τη σχέση μεταξύ του ορίσματος και του παραγώγου. Άλλος λόγος για την χρήση machine learning έγκειται στην πιθανότητα να κρύβονται ανάμεσα σε μεγάλο σωρό από δεδομένα σχέσεις και συσχετίσεις τις οποίες θέλουμε να εξάγουμε [3].

Οι προγραμματιστές κάποιες φορές δημιουργούν μηχανήματα τα οποία δεν λειτουργούν σωστά σε συγκεκριμένα περιβάλλοντα και μερικές φορές τα χαρακτηριστικά του περιβάλλοντος μπορεί να είναι άγνωστα κατά την διαδικασία της παραγωγής οπότε χρησιμοποιούνται μέθοδοι

μηχανικής μάθησης για την βελτίωση αυτών των συστημάτων σε πραγματικό χρόνο. Ακόμα τα περιβάλλοντα αλλάζουν με την πάροδο του χρόνου και δεν χρειάζεται ο συνεχής επανασχεδιασμός του συστήματος από ανθρώπινο δυναμικό [3].

Η μηχανική μάθηση συνήθως χωρίζεται σε δύο βασικούς τύπους. Στην επιτηρούμενη μάθηση (supervised learning) και στην μη επιτηρούμενη μάθηση (unsupervised learning). Στην επιτηρούμενη μάθηση το σύστημα προσπαθεί να μάθει ένα σύνολο καθορισμένων ορισμάτων και παραγώγων χρησιμοποιώντας τη συνάρτηση στόχος (target function). Συγκεκριμένα η συνάρτηση χρησιμοποιείται για την πρόβλεψη της τιμής ενός δεδομένου βάσει των τιμών ενός συνόλου μεταβλητών. Στην μη επιτηρούμενη μάθηση δίνονται παράγωγα δεδομένα χωρίς να έχουν δοθεί ορίσματα. Στόχος του συστήματος είναι να ανακαλύψει σχέσεις και συσχετίσεις από τα δεδομένα βασιζόμενο μόνο στις ιδιότητές τους το οποίο λέγεται ανακάλυψη γνώσης (knowledge discovery). Η μη επιτηρούμενη μάθηση είναι πιο κοντά στον τρόπο μάθησης του ανθρώπου και είναι περισσότερο εφαρμόσιμη από την επιτηρούμενη διότι δεν χρειάζεται κάποιον ειδικό για να σημειώνει χειροκίνητα τα δεδομένα [6].

1.3 Διαδικτυακές Υπηρεσίες

Μέσα στα πλαίσια αυτής της εργασίας είναι η δημιουργία μια υπηρεσίας ιστού ή web service η οποία θα χρησιμοποιείται για την εκτέλεση της υπηρεσίας για οπτική αναγνώριση χαρακτήρων. Η υπηρεσία ιστού είναι μια υπηρεσία ή οποία προσφέρεται από μια ηλεκτρονική συσκευή προς μια άλλη χρησιμοποιώντας τον Παγκόσμιο Ιστό (World Wide Web). Χρησιμοποιούνται πρωτόκολλα όπως http, xml, smtp και μέθοδοι όπως html για τον προγραμματισμό γέφυρας μεταξύ υπολογιστών και χρηστών για την μετακίνηση δεδομένων. Πρακτικά η υπηρεσία ιστού προσφέρει ένα αντικειμενοστραφή περιβάλλον σε ένα server ή μια βάση δεδομένων το οποίο χρησιμοποιείται από άλλον server ο οποίος προσφέρει στο χρήστη ένα περιβάλλον για την χρησιμοποίηση της εφαρμογής. Πλεονέκτημα της χρησιμοποίησης υπηρεσίας ιστού είναι η ευκολία με την οποία γίνονται διαθέσιμα δεδομένα, προϊόντα και υπηρεσίες μέσω του διαδικτύου σε όλους τους ενδιαφερόμενους [7].

Υπάρχουν δύο τεχνολογίες για δημιουργία υπηρεσίας ιστού, το SOAP και το RESTful. Το SOAP είναι ένα πρωτόκολλο για την τυποποίηση των πακέτων που υπάρχουν στα μηνύματα που διαμοιράζονται από της εφαρμογές. Το μόνο που προσδιορίζεται είναι ένας απλός φάκελος

που περιέχει την πληροφορία η οποία διαμοιράζεται καθώς και κανόνες για την μετάφρασή της σε απλή μορφή παρουσίασης με βάση την τεχνολογία XML [8]. Η τεχνολογία REST είναι ιδανική για την δημιουργία εφαρμογών βασισμένων σε υπηρεσία ιστού. Παρέχει σε κάθε πόρο ένα ξεχωριστό κωδικό όπως είναι το URI (uniform resource identifier) και δημιουργεί συσχετίσεις μεταξύ των πόρων. Τα δεδομένα μπορούν να έχουν πολλαπλές αναπαραστάσεις για διαφορετικές καταστάσεις. Χρησιμοποιεί βασικές μεθόδους όπως http, xml και τύπους μέσων όπως ήχος, εικόνα κτλ. [9].

1.4 Δομή Εργασίας

Η εργασία αποτελείται από 5 Κεφάλαια, στα οποία αναλυτικά αναφέρονται:

- Στο Κεφάλαιο 2 περιγράφεται το τεχνολογικό υπόβαθρο της εφαρμογής. Ειδικότερα γίνεται επεξήγηση της μηχανικής μάθησης και των τρόπων μηχανικής μάθησης, γίνεται περιγραφή των decision trees και των νευρωνικών δικτύων (neural networks).
- Στο Κεφάλαιο 3 γίνεται περιγραφή των δικτυακών υπηρεσιών, στις αρχιτεκτονικές που χρησιμοποιούνται και στις μεθόδους και τα εργαλεία όπως το flask framework.
- Στο Κεφάλαιο 4 περιγράφεται λεπτομερώς η μεθοδολογία της εργασίας, πώς δημιουργήθηκε το web service, πληροφορίες για την βιβλιοθήκη ocr και τον τρόπο εκτέλεσής του καθώς και την διαδικασία της εκπαίδευσης.
- Στο Κεφάλαιο 5 γίνεται μια ανάλυση της έρευνας και της εφαρμογής και προκύπτουν κάποια συμπεράσματα καθώς και μελλοντικές επεκτάσεις που μπορούν να γίνουν.

2 ΤΕΧΝΟΛΟΓΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτό το κεφάλαιο θα δούμε την μηχανική μάθηση σαν έννοια, που χρησιμεύει και με ποιόν τρόπο λειτουργεί καθώς και μερικούς τρόπους μηχανικής μάθησης. Ακόμα θα δούμε κάποια πράγματα για τα δενδροδιαγράμματα (decision trees) καθώς και για τα νευρωνικά δίκτυα (neural networks). Τέλος θα δούμε λογισμικά μηχανικής μάθησης για OCR και συγκεκριμένα το Tesseracr και το OCRopus.

2.1 Μηχανική Μάθηση

Η μηχανική μάθηση είναι μια έννοια η οποία είναι εξαιρετικά δύσκολο να οριστεί και ενώ έχουν γίνει προσπάθειες να δοθεί μια επεξήγηση ο καθένας μπορεί να βρει παραδείγματα που βγαίνουν έξω από το πεδίο. Υπάρχει ωστόσο ένας πρακτικός ορισμός ο οποίος αν και ασαφής εμπεριέχει το πεδίο της έρευνας. Οπότε μπορούμε να πούμε πως η μάθηση είναι η βελτίωση της επίδοσης ενός περιβάλλοντος με την λήψη γνώσης μέσω του πειραματισμό στο περιβάλλον αυτό [11].

Η ικανότητα της μάθησης είναι κεντρικό γνώρισμα της νοημοσύνης το οποίο την καθιστά βασική μέριμνα για την τεχνητή νοημοσύνη και την γνωστική ψυχολογία. Το πεδίο της μηχανικής μάθησης ασχολείται με τις προγραμματιστικές εργασίες που σχετίζονται με την μάθηση στους ανθρώπους και τις μηχανές. Παρά την διαφορετικότητα της, η μηχανική μάθηση είναι αναπόσπαστο κομμάτι των δυο πεδίων διότι οι ερευνητές δεν είναι δυνατόν να αγνοούν θέματα όπως παρουσίαση της γνώσης, οργάνωση της μνήμης και επίδοσης τα οποία είναι βασικά κομμάτια της τεχνητής νοημοσύνης. Επίσης η μάθησης χρησιμοποιείται σε οποιοδήποτε τομέα χρησιμοποιεί νοημοσύνη όπως διάγνωση, οργάνωση, φυσική γλώσσα ή έλεγχος κίνησης [11].

Το πεδίο της μηχανικής μάθησης ενώνεται κάτω από την μέριμνα της μάθησης αλλά οι ερευνητές ασχολούνται για διαφορετικούς λόγους. Υπάρχουν τέσσερις βασικοί στόχοι καθένας με την δική του μεθοδολογία, τη δική του προσέγγιση στην αξιολόγηση και τις δικές του επιτυχίες. Ένας στόχος περιέχει την μοντελοποίηση των μηχανισμών που εμπεριέχουν την

ανθρώπινη μάθηση. Σε αυτό το ψυχολογικό κομμάτι οι ερευνητές δημιούργησαν αλγορίθμους οι οποίοι συνεπάγονται με την γνώση της ανθρώπινης γνωστικής αρχιτεκτονικής και είναι σχεδιασμένοι για να εξηγούν συγκεκριμένες συμπεριφορές. Αυτή η προσέγγιση έχει αποφέρει πληθώρα πληροφοριακών μοντέλων τα οποία σχετίζονται με αρκετούς τομείς όπως λύση προβλημάτων, φυσική γλώσσα και αντίληψη γεγονότων. Κάποια από αυτά τα μοντέλα επεξηγούν συμπεριφορές με ίσα δεδομένα και άλλα παίρνουν υπόψη την πιθανότητα λάθους και το χρόνο αντίδρασης του ανθρώπου. Τέτοιες προσομοιώσεις έχουν κυρίως θεωρητικές εφαρμογές στην κατανόηση της συμπεριφοράς στην μάθηση και βοηθάνε στην δημιουργία διδακτικών κειμένων [11].

Διαφορετική ομάδα ερευνητών παίρνει μια πιο εμπειρική προσέγγιση στο θέμα της μηχανικής μάθησης. Ο στόχος είναι να ανακαλύψουν γενικές αρχές που σχετίζονται με τα χαρακτηριστικά των αλγορίθμων και στον τομέα που λειτουργούν. Αυτή η προσέγγιση εμπεριέχει τον πειραματισμό, αλλάζοντας είτε τον αλγόριθμο είτε το πεδίο και παρακολουθώντας στην συνέχεια τις συνέπειες που έχουν αυτές οι αλλαγές στη μάθηση. Κάποιες έρευνες συγκρίνουν διαφορετικές κλάσεις των μεθόδων μάθησης και άλλες ελέγχουν μικρές διαφοροποιήσεις σε συγκεκριμένο αλγόριθμο. Πειραματικές έρευνες της μηχανικής μάθησης έχουν παράγει ένα μεγάλο αριθμό εμπειρικών γενικοποιήσεων όσον αφορά εναλλακτικές μεθόδους οι οποίες παρουσιάζουν περιοχές αδυναμίας και ιδέες για βελτιωμένους αλγορίθμους [11].

Άλλη ομάδα ερευνητών μεταχειρίζεται την μηχανική μάθηση σαν ένα τομέα των μαθηματικών. Ο στόχος αυτών των ομάδων είναι δημιουργήσουν και να αποδείξουν θεωρήματα για την ευπείθεια ολόκληρων κλάσεων των προβλημάτων και τους αλγορίθμους σχεδιασμένους για την λύση αυτών των προβλημάτων. Η τυπική προσέγγιση είναι η προσδιόριση ενός προβλήματος, η υπόθεση εάν μπορεί ή όχι να λυθεί με ένα λογικό αριθμό από εκπαιδευτικών περιπτώσεων και στη συνέχεια η απόδειξη ότι η υπόθεση υπόκειται κάτω από γενικές συνθήκες. Ενώ η εμπειρική προσέγγιση δανείζεται πειραματικές πρακτικές από την φυσική και την ψυχολογία η μαθηματική προσέγγιση δανείζεται από την πληροφορική και την στατιστική [11].

Η τελευταία ομάδα ερευνητών παίρνει μια πιο πρακτική προσέγγιση στη μηχανική μάθηση, με το βασικό στόχο να είναι η χρησιμοποίηση μηχανικής μάθησης για λύση προβλημάτων στον πραγματικό κόσμο. Επειδή η μηχανική μάθηση μπορεί να μετατρέψει

εκπαιδευτικά δεδομένα σε γνώση, έχει την προοπτική να αυτοματοποιήσει την εργασία της λήψης γνώσης. Τα βασικά βήματα εμπεριέχουν τον προγραμματιστή να δημιουργεί ένα πρόβλημα με βάση τη μηχανική μάθηση, να δημιουργεί μια παρουσίαση των εκπαιδευτικών δεδομένων και της γνώσης που έλαβε και τέλος να συνεργάζεται με χρήστες για την ολοκλήρωση του συστήματος εκμάθησης. Αυτός ο τομέας έχει παρουσιάσει πρακτικές εφαρμογές μηχανικής μάθησης στην διάγνωση, στον έλεγχο εργασιών και στην χρονοδρομολόγηση [11].

2.2 Τρόποι μηχανικής μάθησης

Η μηχανική μάθηση χωρίζεται συνήθως σε δυο βασικούς τύπους. Στην επιτηρούμενη μάθηση (supervised learning) και στην μη επιτηρούμενη μάθηση (unsupervised learning).

2.3 Επιτηρούμενη Μάθηση

Η επιτηρούμενη μάθηση είναι η μέθοδος της μηχανικής μάθησης για την εξαγωγή συνάρτησης από επιτηρούμενα εκπαιδευτικά δεδομένα. Τα δεδομένα αυτά εμπεριέχουν σετ με εκπαιδευτικά παραδείγματα. Στην επιτηρούμενη μάθηση κάθε παράδειγμα είναι ένα ζευγάρι συνιστάμενο από ένα όρισμα και ένα επιθυμητό παράγωγο. Ο αλγόριθμος αναλύει τα δεδομένα και παράγει μια συνάρτηση η οποία ονομάζεται ταξινομητική (classifier) εφόσον το αποτέλεσμα είναι διακεκριμένο ή οπισθοδρομική (regression function) εάν τα αποτελέσματα είναι συνεχή. Η συνάρτηση θα πρέπει να προβλέπει σωστά το αποτέλεσμα για οποιοδήποτε έγκυρο όρισμα.

Για να λυθεί ένα πρόβλημα επιτηρούμενης μάθησης χρειάζεται να ακολουθηθούν κάποια βασικά βήματα:

- Προσδιορισμός του τύπου τον εκπαιδευτικών παραδειγμάτων. Η πρώτη ευθύνη του προγραμματιστή είναι να αποφασίσει τι είδους παραδείγματα θα χρησιμοποιήσει. Για παράδειγμα μπορεί να χρησιμοποιήσει ένα γράμμα, μια λέξη ή μια ολόκληρη γραμμή κειμένου.

- Συλλογή εκπαιδευτικού σετ. Το σετ πρέπει να αντιπροσωπεύει την πραγματική χρήση της συνάρτησης. Έτσι μαζεύεται ένα σετ από ορίσματα και τα παράγωγά τους είτε από ειδικούς ερευνητές είτε από μετρήσεις.
- Καθορισμός της αντιπροσώπευσης χαρακτηριστικών γνωρισμάτων της συνάρτησης. Η ακρίβεια της συνάρτησης εξαρτάται έντονα από τον τρόπο με τον οποίο το αντικείμενο εισαγωγής αντιπροσωπεύεται. Χαρακτηριστικά το όρισμα μετασχηματίζεται σε ένα διάνυσμα χαρακτηριστικών γνωρισμάτων το οποίο περιέχει διάφορες πληροφορίες για το αντικείμενο. Ο αριθμός των χαρακτηριστικών δεν πρέπει να είναι μεγάλος λόγω της πληγής της διαστατικότητας αλλά πρέπει να περιέχει αρκετές πληροφορίες ώστε να μπορεί να προβλέψει σωστά το αποτέλεσμα.
- Καθορισμός της δομής της συνάρτησης και του αντίστοιχου αλγορίθμου εκμάθησης. Για παράδειγμα ο προγραμματιστής μπορεί να επιλέξει να χρησιμοποιήσει διανυσματικές μηχανές (vector machines) ή δέντρα απόφασης (decision trees).
- Ολοκλήρωση του σχεδίου. Τρέξιμο του αλγορίθμου εκμάθησης στο μαζεμένο εκπαιδευτικό σύνολο. Κάποιοι αλγόριθμοι επιβλεπόμενης μάθησης απαιτούν από τον χρήση τον καθορισμό ορισμένων παραμέτρων ελέγχου. Αυτές οι παράμετροι μπορούν να ρυθμιστούν με την βελτιστοποίηση της απόδοσης σε ένα υποσύνολο.
- Αξιολόγηση της ακρίβειας της συνάρτησης εκμάθησης. Μετά την ρύθμιση των παραμέτρων, η απόδοση της προκύπτουσας συνάρτησης πρέπει να μετρηθεί σε ένα δοκιμαστικό σετ το οποίο είναι ξεχωριστό από το εκπαιδευτικό σετ.

Υπάρχει ένα ευρύ φάσμα εποπτευόμενων αλγορίθμων εκμάθησης καθένα από αυτά με τα πλεονεκτήματα και τις αδυναμίες του. Δεν υπάρχει ένας συγκεκριμένος αλγόριθμος που να καλύπτει όλα τα προβλήματα επιβλεπόμενης μάθησης. Υπάρχουν τέσσερα βασικά θέματα να ληφθούν υπόψη στην επιβλεπόμενη μάθηση.

2.3.1 Ανταλλαγή προκατάληψης-διαφορετικότητας (Bias-variance tradeoff)

Το πρώτο θέμα είναι η ανταλλαγή μεταξύ της διαφοράς και της προκατάληψης [12]. Ας πούμε ότι έχουμε διαθέσιμα αρκετά διαφορετικά, αλλά εξίσου καλά, εκπαιδευτικά σετ δεδομένων. Ένας αλγόριθμος εκμάθησης είναι προκατηλλειμένος ως προς ένα συγκεκριμένο όρισμα x, εάν κατά την εκπαίδευση σε κάθε σετ δεδομένων είναι συστηματικά λανθασμένος κατά την πρόβλεψη του παραγώγου του x. Ένας αλγόριθμος έχει υψηλή διαφορετικότητα για ένα συγκεκριμένο όρισμα εάν προβλέψει διαφορετικά αποτελέσματα όταν εκπαιδεύεται με διαφορετικά εκπαιδευτικά σετ. Το λάθος της πρόβλεψης συσχετίζεται με το άθροισμα της προκατάληψης και της διαφορετικότητας του αλγορίθμου εκμάθησης [13]. Γενικά υπάρχει μια ανταλλαγή μεταξύ της προκατάληψης και της διαφορετικότητας. Ένας αλγόριθμος με χαμηλή προκατάληψη πρέπει να είναι ευέλικτος ώστε να μπορεί να προσαρμόσει καλά τα δεδομένα. Αν όμως είναι πολύ ευέλικτος θα προσαρμόσει κάθε εκπαιδευτικό σετ διαφορετικά και έτσι θα έχει μεγάλη διαφορετικότητα. Βασικό στοιχείο πολλών μεθόδων επιτηρούμενης μάθησης είναι η δυνατότητά τους να προσαρμόζουν την ανταλλαγή μεταξύ προκατάληψης και διαφορετικότητας είτε αυτόματα είτε δίνοντας στον χρήστη την δυνατότητα να προσαρμόσει μια παράμετρο.

2.3.2 Πολυπλοκότητας συνάρτησης και ποσό εκπαιδευμένων δεδομένων (Function complexity and amount of training)

Το δεύτερο θέμα έχει να κάνει με το ποσό των διαθέσιμων εκπαιδευμένων δεδομένων σχετικά με την πολυπλοκότητα της αληθινής συνάρτησης. Αν η αληθινή συνάρτηση είναι απλή, τότε ένας ευέλικτος αλγόριθμος μάθησης με μεγάλο βαθμό προκατάληψης και μικρό βαθμό διαφορετικότητας θα έχει την δυνατότητα να την μάθει από ένα μικρό δείγμα δεδομένων. Εφόσον όμως η αληθινή συνάρτηση είναι περίπλοκη τότε η συνάρτηση θα μπορεί να μάθει μόνο από ένα μεγάλο ποσοστό εκπαιδευμένων δεδομένων και εφόσον χρησιμοποιεί έναν ευέλικτο αλγόριθμο με χαμηλό βαθμό προκατάληψης και υψηλό βαθμό διαφορετικότητας. Ένας καλός αλγόριθμος λοιπόν προσαρμόζει αυτόματα την ανταλλαγή βασιζόμενος στον αριθμό των δεδομένων αλλά και στην πολυπλοκότητα της συνάρτησης που πρέπει να μάθει.

2.3.3 Διαστατικότητα του διαστήματος του ορίσματος (Dimentionality of the input space)

Το τρίτο θέμα είναι η διαστατικότητα του διαστήματος του ορίσματος. Εάν τα διανύσματα έχουν υψηλή διάσταση το πρόβλημα μπορεί να γίνει πολύ δύσκολο ακόμα και αν η αληθινή συνάρτηση εξαρτάται από ένα μικρό μέρος τους. Αυτό συμβαίνει διότι οι πολλές διαφορετικές διαστάσεις μπορούν να συγχύσουν τον αλγόριθμο και να τον κάνει να έχει υψηλή διαφορετικότητα. Ως εκ τούτου η υψηλή διαστατικότητα απαιτεί την μεταβολή του ταξινομητή σε χαμηλή διαφορετικότητα και υψηλή προκατάληψη. Στην πράξη εάν ο προγραμματιστής καταφέρει να αφαιρέσει τα άσχετα στοιχεία από τα ορίσματα τότε είναι πιθανό να βελτιώσει την ακρίβεια της συνάρτησης μάθησης. Υπάρχουν αλγόριθμοι για μελλοντική χρήση οι οποίοι ψάχνουν και αναγνωρίζουν τα σχετικά στοιχεία και διαγράφουν τα άσχετα. Αυτή είναι μια περίπτωση της γενικότερης στρατηγικής της μείωσης διαστατικότητας, όπου επιδιώκεται η χαρτογράφηση των δεδομένων εισόδου σε χαμηλότερο διαστατικό διάστημα προτού τρέξει ο αλγόριθμος επιβλεπόμενης μάθησης.

2.3.4 Θόρυβος στις τιμές του παραγώγου (Noise in the output values)

Το τέταρτο θέμα είναι ο βαθμός του θορύβου στα επιθυμητά δεδομένα εξόδου. Αν τα επιθυμητά δεδομένα εξόδου είναι συχνά λάθος, λόγω ανθρώπινου λάθους ή προβληματικού αισθητήρα, τότε ο αλγόριθμος μάθησης δεν θα πρέπει να προσπαθήσει να βρει μια συνάρτηση που να ταιριάζει απόλυτα με τα εκπαιδευμένα παραδείγματα. Αυτή είναι άλλη μια περίπτωση που είναι καλύτερο να υιοθετηθεί ένας ταξινομητής με υψηλή προκατάληψη και χαμηλή διαφορετικότητα.

Υπάρχουν αρκετοί τρόποι για να γενικευτεί το βασικό πρόβλημα της επιβλεπόμενης μάθησης. Στην ημιεπιτηρούμενη μάθηση (semi-supervised learning) τα επιθυμητά παράγωγα παρέχονται μόνο για ένα υποσύνολο των εκπαιδευτικών δεδομένων και τα υπόλοιπα δεδομένα είναι χωρίς ετικέτα. Στην ενεργή μάθηση (active learning) αντί να υποθέσουμε ότι έχουμε όλα τα εκπαιδευτικά παραδείγματα από την αρχή, οι αλγόριθμοι μαζεύουν διαδραστικά καινούργια παραδείγματα, συνήθως κάνοντας ερωτήσεις στον χρήστη. Συχνά οι ερωτήσεις έχουν να κάνουν με δεδομένα χωρίς ετικέτα σενάριο το οποίο συνδυάζει την ημιεπιτηρούμενη και την ενεργή

μάθηση. Δομημένη πρόβλεψη (structured prediction) χρησιμοποιείται όταν το επιθυμητό αποτέλεσμα είναι περίπλοκο αντικείμενο , όπως για παράδειγμα parse tree ή labeled graph, τότε οι βασικές μέθοδοι πρέπει να επεκταθούν. Μαθαίνοντας να ταξινομεί (learning to rank) χρησιμοποιείται όταν το όρισμα είναι ένα σετ από αντικείμενα και το επιθυμητό αποτέλεσμα είναι μια ταξινόμηση αυτών των αντικειμένων και τότε πρέπει οι βασικές μέθοδοι να επεκταθούν.

2.4 Μη επιτηρούμενη μάθηση (Unsupervised learning)

Η μη επιτηρούμενη μάθηση μελετά το πώς τα συστήματα μπορούν να μάθουν να αντιπροσωπεύουν συγκεκριμένα σχέδια ορισμάτων έτσι ώστε να απεικονίζουν τη στατιστική δομή της συνολικής συλλογής των ορισμάτων. Σε αντίθεση με την επιβλεπόμενη μάθηση δεν υπάρχουν συγκεκριμένα θεμιτά αποτελέσματα ή περιβαλλοντικές αξιολογήσεις που να συνδέονται με κάθε όρισμα αλλά η μη επιβλεπόμενη μάθηση εφαρμόζει προγενέστερες προκαταλήψεις ως προς ποιες πτυχές της δομής του ορίσματος πρέπει να συλληφθεί από το παράγωγο [14].

Το μοναδικό πράγμα που έχουν να χρησιμοποιήσουν οι μέθοδοι μη επιβλεπόμενης μάθησης είναι τα παρατηρηθέντα σχέδια ορισμάτων x_i, τα οποία συνήθως υποτίθεται ότι είναι ανεξάρτητα δείγματα από μια κατώτερη, άγνωστη διανομή πιθανότητας $P_I[x]$, και μιας ρητής ή υπονοούμενης προηγούμενης πληροφορίας ως προ το τι είναι σημαντικό. Μια βασική έννοια είναι ότι το όρισμα, όπως μια εικόνα, έχει ακραίες ανεξάρτητες αιτίες, όπως κάποια αντικείμενα σε συγκεκριμένη τοποθεσία φωτίζονται από συγκεκριμένη πηγή φωτός. Δεδομένου πως σε αυτές τις ανεξάρτητες αιτίες πρέπει να ενεργήσουμε, η καλύτερη αντιπροσώπευση για ένα όρισμα είναι στο επίπεδό τους. Δύο κλάσεις μεθόδων έχουν προταθεί για την μη επιβλεπόμενη μάθηση. Οι τεχνικές εκτίμησης πυκνότητας χτίζουν ρητά στατιστικά μοντέλα (όπως BAYESIAN NETWORKS) για το πώς υποκείμενες αιτίες μπορούν να δημιουργήσουν το όρισμα. Οι τεχνικές εξαγωγής προσπαθούν να εξάγουν στατιστικές τακτικότητες (ή αταξίες) απευθείας από τα ορίσματα [14].

Η μεγαλύτερη κλάση των μεθόδων της μη επιβλεπόμενης μάθησης αποτελείται από μεθόδους για τη μέγιστη εκτιμώμενη πιθανότητα πυκνότητας (maximum likelihood density estimation). Αυτά είναι βασισμένα στη δημιουργία παραμετροποιημένων μοντέλων P[x;G] (με παραμέτρους G) της διανομής πιθανότητας $P_I[x]$ όπου η φόρμες του μοντέλου περιορίζονται από μια προηγούμενη πληροφορία με τη μορφή των αντιπροσωπευτικών στόχων. Αυτά τα μοντέλα ονομάζονται συνθετικά (synthetic) ή παραγωγικά (generative), αφού δοθούντως μιας συγκεκριμένης τιμής του G, διευκρινίζουν πώς να συνθέσουν ή να παράγουν δείγματα του x από τη συνάρτηση P[x;G], που τα στατιστικά τους να αντιστοιχούν το $P_I[x]$. Ένα τυπικό μοντέλο έχει την δομή [14]:

$$P[x;G] = \sum_y P[y; G]P[x[y; G]]$$

Όπου το y αντιπροσωπεύει όλες τις πιθανές αιτίες του ορίσματος x. Το χαρακτηριστικό μέτρο του βαθμού του κακού συνδυασμού λέγεται the Kullback-Leibler divergence:

$$KL[P_I[x],P[x;G]] = \sum_x PI[x] \log[\frac{PI[x]}{P[x;G]}] >= 0$$

Με ισότητα αν και μόνο αν $P_I[x] = P[x;G]$ [14].

Λαμβάνοντας για όρισμα ένα σχέδιο x, το πιο γενικό αποτέλεσμα του μοντέλου αυτού είναι το οπίσθιο τμήμα, το αναλυτικό ή διανομή αναγνώρισης P[y|x;G], η οποία αναγνωρίζει ποιες συγκεκριμένες αιτίες μπορεί να κρυφτεί από το x. Αυτή η αναλυτική διανομή είναι το στατιστικό ανάποδο της συνθετικής διανομής [14].

Ένα πολύ απλό μοντέλο μπορεί να χρησιμοποιηθεί σαν παράδειγμα ομαδοποίησης [16]. Ας πάρουμε υπόψη την κατάσταση στην οποία έχουμε 2 τιμές για το y (1 και 2), με P[y=1] = π; P[y=2] = 1-π, όπου το 'π' ονομάζεται αναλογική μίξη (mixing proportion) και 2 διαφορετικές Gaussian διανομές για τις δραστηριότητες x των φωτοδεκτών εξαρτώμενο από ποιο y έχει διαλεγεί : $P[x[y=1]] \sim N[\mu^1, \Sigma^1]$ και $P[x[y=2]] \sim N[\mu^2, \Sigma^2]$, όπου μ^* είναι μέσα (means) και Σ^* είναι μήτρες συνδιακύμανσης (covariance matrices). Η μη επιβλεπόμενη μάθηση των μέσων καθορίζει τις ομάδες (clusters). Η μηχανική μάθηση των mixing proportions και covariance matrices χαρακτηρίζει το μέγεθος και το σχήμα των clusters. Η μεταγενέστερη διανομή (posterior distribution) $P[y=1|x;\mu^*\Sigma^*]$ αναφέρει πόσο πιθανό είναι μια νέα εικόνα x να παρήχθει

από την πρώτη ομάδα. Ομαδοποίηση μπορεί να γίνει με ή χωρίς επιβλεπτική πληροφορία για τις διαφορετικές κλάσεις. Αυτό το μοντέλο λέγεται mixture of Gaussians [14].

Η εκτίμηση πυκνότητας μέγιστης πιθανότητας (maximum likelihood density estimation) και οι προσεγγίσεις σε αυτή, καλύπτουν ένα πολύ ευρύ φάσμα των αρχών που έχουν προταθεί για την μη επιτηρούμενη μάθηση [14]. Αυτό περιέχει κομμάτια της έννοιας ότι τα αποτελέσματα πρέπει να μεταβιβάζουν σχεδόν όλες τις πληροφορίες του ορίσματος, ότι πρέπει να μπορούν να αναδημιουργήσουν με επιτυχία τα ορίσματα, να υπόκεινται σε περιορισμούς όπως να είναι ανεξάρτητα ή αραιά, και να δίνουν αναφορά για τις υποκείμενες αιτίες των εισαγόμενων. Διαφορετικοί μηχανισμοί, εκτός της ομαδοποίησης, έχουν προταθεί για κάθε μια από αυτές, συμπεριλαμβανομένων είδους Hebbian learning, the Boltzmann and Helmholtz machines, sparse-coding, άλλα πρότυπα μιγμάτων (mixture models) και ανεξάρτητη ανάλυση υλικών [14].

Η εκτίμηση πυκνότητας (density estimation) είναι απλά μια ευρετική για την εκμάθηση καλών παρουσιάσεων. Μπορεί να είναι πολύ αυστηρό, καθιστώντας απαραίτητο να χτιστεί ένα μοντέλο όλης της άχρηστης αφθονίας σε αισθητήριο όρισμα. Μπορεί επίσης να είναι πολύ χαλαρό, ένας πίνακας ο οποίος αναφέρει $P_I[x]$ για κάθε x, μπορεί να είναι ένας εξαιρετικός τρόπος να μοντελοποιηθεί η διανομή αλλά δεν παρέχει τρόπο να παρουσιάζει συγκεκριμένα παραδείγματα του x [14].

2.5 Δενδροδιαγράμματα (Decision Trees)

Τα δενδροδιαγράμματα (decision trees) είναι ένα κλασσικό και φυσικό μοντέλο της μάθησης. Είναι συνδεδεμένο με την βασική αρχή της πληροφορικής, το «διαίρει και βασίλευε» [10]. Ένα decision tree περιγράφει γραφικά τις αποφάσεις που πρέπει να παρθούν, τα συμβάντα που μπορεί να γίνουν, και το αποτέλεσμα που σχετίζονται με το συνδυασμό αποφάσεων και συμβάντων. Πιθανότητες ορίζονται στα συμβάντα και τιμές καθορίζονται για κάθε αποτέλεσμα. Ο μεγαλύτερος στόχος της ανάλυσης είναι ο προσδιορισμός της καλύτερης δυνατής απόφασης [16].

2.5.1 Κόμβοι (nodes) και Κλάδοι (branches)

Τα decision trees περιέχουν έννοιες όπως κόμβους (nodes), κλάδοι (branches), τερματικές τιμές (terminal values), στρατηγική (strategy) [16]. Υπάρχουν τρία είδη κόμβων και δύο είδη κλάδων.

Ένας κόμβος απόφασης (decision node) είναι ένα σημείο όπου μια επιλογή πρέπει να γίνει και εμφανίζεται ως ένα τετράγωνο. Οι κλάδοι που επεκτείνονται από ένα κόμβο απόφασης ονομάζονται κλάδοι απόφασης και κάθε κλάδος αντιπροσωπεύει μια από τις πιθανές εναλλακτικές λύσεις ή σχέδια δράσης σε εκείνο το χρονικό σημείο. Το σύνολο των εναλλακτικών πρέπει να είναι αμοιβαία αποκλειστικό (εάν επιλεχθεί ένα δεν μπορεί να επιλεχθεί κάποιο άλλο) και συλλογικά εξαντλητικό (πρέπει να περιληφθούν όλες οι πιθανές εναλλακτικές λύσεις στο σύνολο) [25].

Ένας κόμβος γεγονότος (event node) είναι ένα σημείο όπου επιλύεται η αβεβαιότητα (ένα σημείο όπου ο ιθύνον μαθαίνει το περιστατικό ενός γεγονότος). Ο κόμβος γεγονότος, που μερικές φορές ονομάζεται κόμβος πιθανότητας, παρουσιάζεται ως κύκλος. Το σύνολο γεγονότων αποτελείται από τους κλάδους γεγονότων ο οποίος επεκτείνεται από τον κόμβο γεγονότος, κάθε κλάδος αντιπροσωπεύει ένα από τα πιθανά γεγονότα που μπορεί να εμφανιστούν σε εκείνο το σημείο. Όπως και στους κόμβους απόφασης τα σετ των γεγονότων πρέπει να είναι αποκλειστικά και εξαντλητικά. Κάθε γεγονός ορίζεται μιας πιθανότητας και το άθροισμα των πιθανοτήτων πρέπει να ισούται με τη μονάδα [25].

Το τρίτο είδος κόμβου είναι ο τερματικός κόμβος (terminal node), που αντιπροσωπεύει το τελικό αποτέλεσμα ενός συνδυασμού αποφάσεων και γεγονότων. Οι τερματικοί κόμβοι είναι τα σημεία τέλους ενός δέντρου απόφασης και παρουσιάζονται ως το τέλος ενός κλάδου στα χειρόγραφα διαγράμματα και ως μια τρίγωνη ή κάθετη μπάρα στα παραγόμενα από υπολογιστή διαγράμματα [25].

Κάθε τερματικός κόμβος έχει μια σχετική τερματική τιμή, η οποία κάποιες φορές ονομάζεται τιμή εξόφλησης (payoff value), τιμή έμβασης (outcome value) ή τιμή σημείου τέλους (endpoint value). Κάθε τερματική τιμή μετρά το αποτέλεσμα ενός σεναρίου (η ακολουθία αποφάσεων και γεγονότων σε ένα μοναδικό μονοπάτι που οδηγεί από τον αρχικό κόμβο απόφασης σε ένα συγκεκριμένο τερματικό κόμβο). Για να καθοριστεί η τερματική τιμή μια

προσέγγιση ορίζει μια τιμή ταμειακής ροής σε κάθε κόμβο απόφασης και κόμβο γεγονότος και στη συνέχεια αθροίζει τις τιμές ταμειακής ροής στους κλάδους που οδηγούν σε ένα τερματικό κόμβο για να υπολογίσουν την τερματική τιμή. Κάποια προβλήματα απαιτούν ένα πιο επιμελημένο μοντέλο τιμής για να καθορίσουν τις τερματικές τιμές [25].

2.5.2 Είδη προβλημάτων μάθησης

Υπάρχει ένας μεγάλος αριθμός τυπικών επαγωγικών προβλημάτων μάθησης. Η βασική διαφορά τους είναι στο τι προσπαθούνε να προβλέψουν. Παρακάτω παρουσιάζονται μερικά προβλήματα:

Οπισθοδρόμηση (regression): Προσπαθεί να προβλέψει μια πραγματική τιμή. Για παράδειγμα προβλέπει την αξία μιας μετοχής βασιζόμενο σε προηγούμενη απόδοσή της ή να προβλέπει τον βαθμό ενός μαθητή στην τελική εξέταση βασιζόμενο στην βαθμολογία του σε προηγούμενες γραπτές εξετάσεις [10].

Δυαδική ταξινόμηση (binary classification): Προσπάθεια να προβλεφθεί μια απλή απάντηση ναι/όχι. Για παράδειγμα να προβλεφθεί εφόσον ένας μαθητής θα διαλέξει ένα συγκεκριμένο μάθημα ή να προβλεφθεί εάν ένας χρήστης θα αφήσει θετική ή αρνητική κριτική σε ένα προϊόν.

Multiclass ταξινόμηση: Προσπάθεια να μπει ένα παράδειγμα σε έναν αριθμό από κλάσεις. Για παράδειγμα να προβλεφθεί εφόσον μια είδηση είναι για διασκέδαση, αθλητικά, πολιτική, θρησκεία κτλ.

Κατάταξη (ranking): Προσπάθεια να ταξινομηθεί ένα σύνολο αντικειμένων με βάση την σχετικότητα. Για παράδειγμα να προβλεφθεί η σειρά με την οποία πρέπει να εμφανιστούν ιστοσελίδες με βάση την ερώτηση του χρήστη.

Παρακάτω θα δούμε ένα παράδειγμα που χρησιμοποιεί την πιο απλή χρήση δενδροδιαγράμματος, τη δυαδική ταξινόμηση (binary classification).

Ας υποθέσουμε πως στόχος είναι να προβλέψουμε εάν κάποιος άγνωστος χρήστης θα μείνει ευχαριστημένος από κάποιο άγνωστο μάθημα και η απάντηση πρέπει να είναι ένα απλό ναι ή όχι. Για να απαντήσουμε σε αυτό το ερώτημα μπορούμε να κάνουμε δυαδικές ερωτήσεις

για το χρήστη ή το μάθημα. Για παράδειγμα αν υποθέσουμε ότι ένα μάθημα βρίσκεται κάτω από την έννοια των συστημάτων και αν με μια ερώτηση μάθουμε ότι ο χρήστης έχει παρακολουθήσει και άλλα μαθήματα συστημάτων και στη συνέχεια μάθουμε ότι στον χρήστη δεν άρεσαν τα μαθήματα, τότε μπορούμε να προβλέψουμε ότι στον χρήστη δεν θα αρέσει το συγκεκριμένο μάθημα [10].

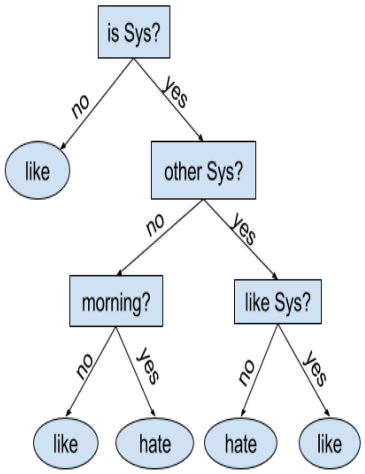
Στόχος στην μάθηση είναι η αναγνώριση των ερωτήσεων που πρέπει να γίνουν, με ποια σειρά πρέπει να ερωτηθούν και ποια απάντηση πρέπει να προβλεφθεί εφόσον έχουν γίνει οι ερωτήσεις. Τα Decision trees παίρνουν το όνομά τους επειδή μπορούμε να γράψουμε τις ερωτήσεις και τις προβλέψεις σε μορφή δέντρου όπως στην Εικόνα 2.1. Σε αυτό το σχήμα οι ερωτήσεις γράφονται σε εσωτερικούς κόμβους (ορθογώνια), και οι προβλέψεις γράφονται στα φύλλα (leaves) που σχηματίζονται με οβάλ σχήμα. Κάθε μη τερματικός κόμβος έχει δύο παιδιά, το αριστερό παιδί προσδιορίζει τι πρέπει να γίνει αν η απάντηση είναι «όχι» και το δεξί αν η απάντηση είναι «ναι» [10].

Για να μάθει το μοντέλο χρειάζεται training data. Ας πάρουμε το παραπάνω παράδειγμα με τα μαθήματα. Στην Εικόνα 2.2 παρέχονται παραδείγματα από σετ χρηστών/μαθημάτων μαζί με την σωστή απάντηση (εάν ο χρήστης είναι ευχαριστημένος από το μάθημα). Από αυτά, πρέπει να δημιουργήσουμε τις ερωτήσεις. Στον πίνακα 2.1 υπάρχουν 20 παραδείγματα αξιολόγησης μαθημάτων τα οποία περιέχουν την αξιολόγηση αλλά και απαντήσεις στα ερωτήματα που μπορούμε να κάνουμε. Από 0-2 θα πούμε ότι τα μαθήματα είναι αρεστά και για -2, -1 ότι δεν είναι. Στο παράδειγμα αυτό θα ονομάζουμε τις ερωτήσεις που μπορούμε να κάνουμε ως χαρακτηριστικά (features) και τις απαντήσεις ως χαρακτηριστικές τιμές (feature values). Η αξιολόγηση είναι η ταμπέλα (label),ένα παράδειγμα είναι απλά ένα σετ από χαρακτηριστικές τιμές και οι εκπαιδευτικές τιμές (training data) είναι ένα σετ παραδειγμάτων μαζί με ταμπέλες [10]. .

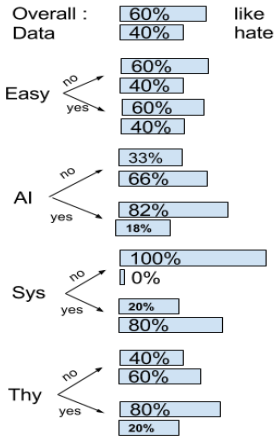
Υπάρχει πολύ μεγάλος αριθμός δέντρων που μπορούν να δημιουργηθούν και είναι προγραμματιστικά αδύνατο να μελετηθούν όλα ώστε να διαλέξουμε το καλύτερο. Σε αυτή την περίπτωση είναι προτιμότερο να δημιουργήσουμε ένα δέντρο άπληστα (greedily).

Το πρώτο πράγμα που πρέπει να βρούμε είναι ένα χαρακτηριστικό το οποίο μας βοηθάει τα μέγιστα για να μαντέψουμε αν ο μαθητής θα ευχαριστηθεί το μάθημα. Σωστός τρόπος σκέψης είναι να κοιτάξουμε το ιστόγραμμα (histogram), Εικόνα 2.2, των ταμπελών για κάθε

χαρακτηριστικό. Κάθε ιστόγραμμα δείχνει τη συχνότητα των «αρέσει», «δεν αρέσει» ταμπελών για κάθε πιθανή τιμή των χαρακτηριστικών. Από το πρώτο χαρακτηριστικό δεν μπορούμε να βγάλουμε χρήσιμα συμπεράσματα καθώς βλέπουμε ότι τα ποσοστά στο «ναι» και στο «όχι» είναι παρόμοια. Στο δεύτερο χαρακτηριστικό ωστόσο μπορούμε να συμπεράνουμε με ακρίβεια και στις δύο περιπτώσεις, δηλαδή είτε είναι ναι είτε όχι, ότι του μαθητή θα του αρέσει ή όχι το μάθημα, ανάλογα με το αποτέλεσμα [10].



Εικόνα 2.1: Ένα δενδροδριάγραμμα για ένα σύστημα που προτείνει μαθήματα [10]



Εικόνα 2.2*: Ιστόγραμμα ταμπελών. [10]*

Ratings	Easy ?	AI?	Sys?	Thy?	Morning?
+2	y	y	n	y	n
+2	y	y	n	y	n
+2	n	y	n	n	n
+2	n	n	n	y	n
+2	n	y	y	n	y
+1	y	y	n	n	n
+1	y	y	n	y	n
+1	n	y	n	y	n
0	n	n	n	n	y
0	y	n	n	y	y
0	n	y	n	y	n
0	y	y	y	y	y
-1	y	y	y	n	y
-1	n	n	y	y	n
-1	n	n	y	n	y
-1	y	n	y	n	y
-2	n	n	y	y	n
-2	n	y	y	n	y
-2	y	n	y	n	n
-2	y	n	y	n	y

Πίνακας 2.1: Παραδείγματα αξιολόγησης μαθημάτων.

2.6 Νευρωνικά Δίκτυα

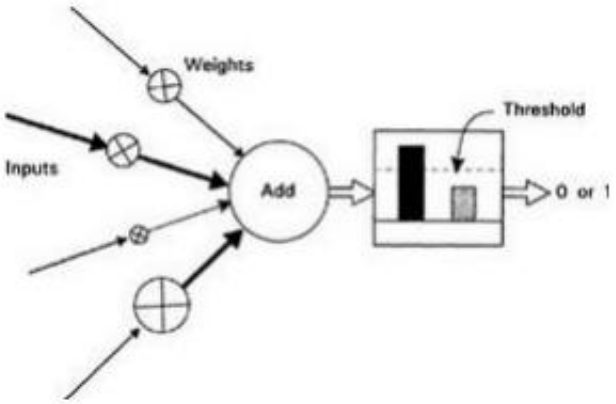
Ο πιο απλός ορισμός ενός νευρωνικού δικτύου, πιο σωστά αναφερόμενο ως τεχνητό νευρωνικό δίκτυο, (artificial neural network ANN), αποδίδεται από τον εφευρέτη ενός από τους πρώτους νευρωνικούς υπολογιστές Dr. Robert Hetch-Nielsen, ο οποίος καθορίζει ένα νευρωνικό δίκτυο ως «ένα υπολογιστικό σύστημα το οποίο απαρτίζεται από ένα αριθμό απλών, ιδιαίτερα διασυνδεδεμένα επεξεργαζόμενα στοιχεία τα οποίο επεξεργάζονται πληροφορίες με την δυναμική απόκρισή τους σε εξωτερικές εισαγωγές» [27].

Στην μηχανική μάθηση τα τεχνητά νευρωνικά δίκτυα είναι μια οικογένεια μοντέλων εμπνευσμένα από βιολογικά νευρωνικά δίκτυα (το κεντρικό νευρωνικό σύστημα των ζώων και συγκεκριμένα τον εγκέφαλο) τα οποία χρησιμοποιούνται για να εκτιμούν ή να προσεγγίζουν συναρτήσεις που εξαρτώνται από μεγάλο αριθμό εισαγωγών και είναι γενικά άγνωστες [28].

Ο ανθρώπινος εγκέφαλος περιέχει περίπου 10^{11} νευρικά κύτταρα ή νευρώνες. Οι νευρώνες επικοινωνούν μέσω ηλεκτρικών σημάτων τα οποία είναι μικρής χρονικής διάρκειας ερεθίσματα ή αλλαγές στην τάση του τοίχους των κυττάρων. Τις εσωνευρωνικές συνδέσεις μεσολαβούν ηλεκτροχημικές συνδέσεις που ονομάζονται συνάψεις (synapses) οι οποίες βρίσκονται σε κλάδους των κυττάρων και αναφέρονται ως δενδρίτες (dendrites) [29].

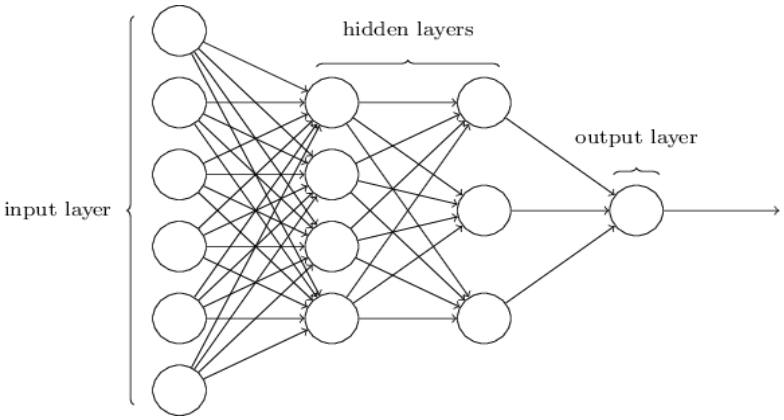
Τα τεχνητά ταυτόσημα των βιολογικών νευρώνων είναι οι κόμβοι. Οι συνάψεις μοντελοποιούνται από ένα μοναδικό αριθμό η βάρος ώστε κάθε εισαγωγικό να πολλαπλασιάζεται με ένα βάρος προτού σταλθεί από το ταυτόσημο του κυτταρικού σώματος. Εδώ τα επιβαρυμένα σήματα αθροίζονται για να παράγουν ένα κόμβο ενεργοποίησης, στην Εικόνα 2.3 αυτό φαίνεται ως threshold logic unit (TLU). Έπειτα η ενεργοποίηση συγκρίνεται με ένα κατώφλι και αν η ενεργοποίηση ξεπεράσει το κατώφλι η μονάδα παράγει ένα υψηλής τιμής αποτέλεσμα αλλιώς βγάζει μηδενικό. Στην Εικόνα 2.4 το μέγεθος των σημάτων αναπαρίσταται με το πλάτος των αντίστοιχων βελών [29].

Σαν δίκτυο λογίζεται κάθε σύστημα τεχνητών νευρώνων είτε πρόκειται για ένα μοναδικό κόμβο είτε μια συλλογή από κόμβους όπου κάθε κόμβος είναι συνδεδεμένος με έναν άλλον στον ιστό [29].



Εικόνα 2.3: Τεχνητός Νευρώνας.

Τα νευρωνικά δίκτυα οργανώνονται σε στρώματα. Αυτά τα στρώματα αποτελούνται από διάφορους διασυνδεδεμένους κόμβους οι οποίοι περιέχουν μια συνάρτηση ενεργοποίησης (activation fanction). Τα σχέδια παρουσιάζονται στο δίκτυο μέσω του στρώματος εισαγωγής (input layer) το οποίο επικοινωνεί με ένα ή περισσότερα κρυμμένα στρώματα (hidden layers) όπου η πραγματική διεργασία γίνεται μέσω ενός συστήματος σταθμισμένων συνδέσεων. Τα κρυμμένα στρώματα συνδέονται με ένα στρώμα παραγωγής (output layer) όπου η απάντηση είναι το παράγωγο όπως φαίνεται στην Εικόνα 2.4 [27].



Εικόνα 2.4: Τεχνητό Νευρωνικό Δίκτυο1

[1] Πηγή: http://neuralnetworksanddeeplearning.com/images/tikz11.png

2.7 Learning rules

Τα περισσότερα τεχνητά νευρωνικά δίκτυα περιέχουν κάποια μορφή μαθητικού κανόνα (learning rule) ο οποίος τροποποιεί τα βάρη των συνδέσεων σύμφωνα με τα σχέδια εισαγωγής τα οποία του παρουσιάζονται. Κατά μια έννοια τα δίκτυα μαθαίνουν από παραδείγματα όπως τα βιολογικά δίκτυα όπως για παράδειγμα ένα παιδί μαθαίνει να αναγνωρίζει τα σκυλιά από παραδείγματα σκύλων [27]. Υπάρχουν διαφορετικά είδη μαθητικών κανόνων και παρουσιάζονται παρακάτω:

2.7.1 Hebbian Learning Rule

Ένας κοινός τρόπος να υπολογιστούν οι αλλαγές στα βάρη των συνδέσεων σε ένα νευρωνικό δίκτυο είναι το Hebbian learning rule όπου μια αλλαγή στη δύναμη μιας σύνδεσης είναι μια συνάρτηση που αποτελείται από presynaptic neural και postsynaptic neural δραστηριότητες [30]. Ο κανόνας αυτός βασίζεται στην υπόθεση ότι αν δύο γειτονικοί νευρώνες πρέπει να ενεργοποιηθούν και να απενεργοποιηθούν την ίδια στιγμή τότε τα βάρη που συνδέουν αυτούς τους νευρώνες πρέπει να αυξηθούν. Στους νευρώνες που λειτουργού σε αντίθετες φάσεις τα βάρη πρέπει να μειωθούν. Αν δεν υπάρχει κάποια σύγκριση μεταξύ των σημάτων τότε τα βάρη πρέπει να παραμείνουν αμετάβλητα [31].

2.7.2 The Delta Rule

Ο κανόνας αυτός λέγεται και μέθοδος least mean square (LMS) και είναι ένας από τους πιο κοινώς χρησιμοποιούμενους κανόνες μάθησης. Για ένα δεδομένο εισαγόμενο πίνακα ο πίνακας εξόδου συγκρίνεται με την σωστή απάντηση. Εάν η διαφορά τους είναι μηδέν δεν γίνεται μάθηση, αλλιώς τα βάρη προσαρμόζονται για να μειώσουν την διαφορά. Η αλλαγή του βάρους από **ui** σε **uj** δίνεται από την συνάρτηση: ***Dwij = r * ai * ej***, όπου **r** είναι η αναλογία μάθησης, **ai** αντιπροσωπεύει την ενεργοποίηση του **ui**, και **ej** είναι η διαφορά μεταξύ του προσδοκώμενου αποτελέσματος και του πραγματικού αποτελέσματος **uj** [30].

2.7.3 Widrow-Hoff LMS Learning Rule

Ο κανόνας αυτός είναι εφαρμόσιμος για επιτηρούμενη εκπαίδευση νευρωνικών δικτύων. Είναι ανεξάρτητος από την συνάρτηση ενεργοποίησης των νευρώνων που χρησιμοποιούνται εφόσον ελαχιστοποιεί το τετραγωνικό λάθος μεταξύ του θεμιτού αποτελέσματος και την τιμής ενεργοποίησης του νευρώνα. Αυτός ο κανόνας μπορεί να θεωρηθεί ως μια ειδική περίπτωση του delta rule [30].

2.7.4 Διαφορές νευρωνικών δικτύων με συμβατικούς υπολογισμούς

Ένας συμβατικός σειριακός υπολογιστής έχει έναν κεντρικό επεξεργαστή που μπορεί να εξετάσει μια σειρά θέσεων μνήμης όπου είναι αποθηκευμένα δεδομένα και πληροφορία. Οι υπολογισμοί γίνονται με τον επεξεργαστή να διαβάζει οδηγίες καθώς και δεδομένα που παρέχονται από τις διευθύνσεις της μνήμης και, εκτελεί τις οδηγίες και τα αποτελέσματα σώζονται σε μια συγκεκριμένη θέση στη μνήμη. Σε ένα σειριακό σύστημα ο τα υπολογιστικά βήματα είναι αιτιοκρατικά, διαδοχικά και λογικά και η κατάσταση μιας δεδομένης μεταβλητής μπορεί να ανιχνευθεί από τη μια ακολουθία στην άλλη [27].

Σε αντίθεση τα νευρωνικά δίκτυα δεν είναι διαδοχικά ή απαραίτητα αιτιοκρατικά. Δεν υπάρχουν σύνθετοι κεντρικοί επεξεργαστές, αλλά υπάρχουν πολλοί απλοί που δεν κάνουν τίποτα παραπάνω από το να παίρνουν το σταθμισμένο άθροισμα των εισαγόμενών τους από τους άλλους επεξεργαστές. Τα νευρωνικά δίκτυα δεν εκτελούν προγραμματισμένες οδηγίες αλλά αποκρίνονται παράλληλα στο σχέδιο των εισαγόμενων που του παρουσιάζονται. Επίσης δεν υπάρχουν ξεχωριστές διευθύνσεις στη μνήμη για αποθήκευση δεδομένων ανταυτού η πληροφορία περιέχεται στη γενική ενεργοποιημένη κατάσταση του δικτύου. Έτσι η γνώση αντιπροσωπεύεται από το ίδιο το δίκτυο το οποίο είναι πραγματικά περισσότερο από το άθροισμα των επιμερους συστατικών του [27].

2.7.5 Χρήση νευρωνικών δικτύων

Τα νευρωνικά δίκτυα λειτουργούν κατά προσέγγιση και δεν πρέπει να χρησιμοποιούνται από συστήματα που δεν επιτρέπουν το λάθος. Για παράδειγμα δεν είναι συνετό να χρησιμοποιηθούν νευρωνικά δίκτυα για τον υπολογισμό του υπολοίπου χρηματικών επιταγών. Ωστόσο μπορούν χρησιμοποιηθούν στα παρακάτω συστήματα: [27]

- Σύλληψη ενώσεων ή ανακάλυψη τακτικοτήτων μέσα σε ένα σύνολο σχεδίων.
- Όπου ο όγκος, ο αριθμός των μεταβλητών ή η ποικιλομορφία των δεδομένων είναι πολύ μεγάλη.
- Οι σχέσεις μεταξύ των μεταβλητών είναι αόριστα κατανοητές.
- Οι σχέσεις είναι δύσκολο να περιγράφουν επαρκώς χρησιμοποιώντας συμβατικές προσεγγίσεις.

2.8 ΛΟΓΙΣΜΙΚΟ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΜΕ OCR

Η γλώσσα προγραμματισμού python είναι για γλώσσα γενικού χαρακτήρα. Είναι μια υψηλού βαθμού γλώσσα,δυναμική, αντικειμενοστραφής και λειτουργεί σε διαφορετικές πλατφόρμες (cross-platform) χαρακτηριστικά τα οποία την κάνουν πολύ ελκυστική στους προγραμματιστές. Τρέχει σε όλες τις πλατφόρμες και σε όλα τα λειτουργικά συστήματα [18].

Προσφέρει μεγάλη παραγωγικότητα σε όλους τους κύκλους ενός λογισμικού όπως την ανάλυση,το σχεδιασμό, τη δημιουργία πρωτοτύπου τον προγραμματισμό, τη δοκιμή, τη διόρθωση, τον συντονισμό, την τεκμηρίωση, την ανάπτυξη και την συντήρηση. Η python προσφέρει μια μοναδική μίξη κομψότητας, απλότητας, πρακτικότητας και δύναμης. Λόγω της μεγάλης συνέπειας και τακτικότητας όπως και της πλούσιας βιβλιοθήκης της και τα πολλά βοηθήματα έτοιμα προς χρήση, βοηθάει στην παραγωγικότητα. Είναι εύκολη στην εκμάθηση και στη χρήση ωστόσο δυνατή ώστε να χρησιμοποιείται από τους ειδικούς [18].

Είναι μια γλώσσα απλή αλλά σύνθετη και ακολουθεί την ιδέα πως αν συμπεριφέρεται με κάποιο τρόπο σε ένα τομέα, τότε ιδεολογικά θα πρέπει να δουλεύει εξίσου σε όλους. Επίσης ακολουθεί τον κανόνα που λέει πως μια γλώσσα δεν πρέπει να έχει συντομεύσεις ή ειδικές

καταστάσεις αλλά μια καλή γλώσσα πρέπει να έχει μέτρο ανάμεσα στο πρακτικό και στη γεύση [18].

```
#!/usr/bin/python
print "Hello, Python!"
```

Εμφανίζει σαν αποτέλεσμα το κείμενο «Hello, Python!

Υπάρχουν αρκετές βιβλιοθήκες που χρησιμοποιούν μεθόδους μηχανικής μάθησης για την αναγνώρισης χαρακτήρων, φτιαγμένες με διαφορετικές γλώσσες προγραμματισμού και μεθόδους. Δύο από τις πιο γνωστές είναι η βιβλιοθήκη OCRopus και η βιβλιοθήκη tesseract.

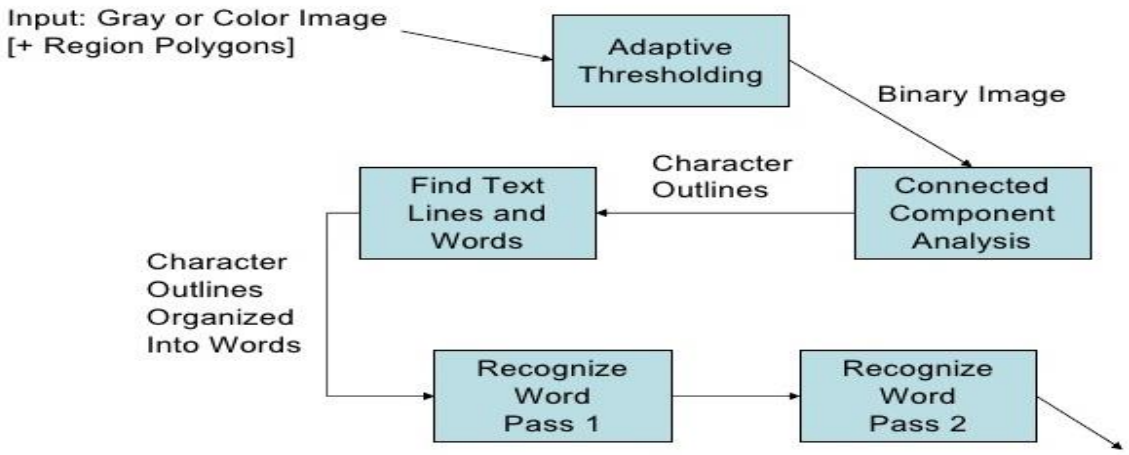
2.8.1 Βιβλιοθήκη Tesseract

Το σύστημα αναγνώρισης γραμματοσειράς Tesseract είναι βασισμένο σε ένα σύστημα οπτικής αναγνώρισης χαρακτήρων της Hewllet Packard και είναι ανοιχτού κώδικα. Εσωτερικά χρησιμοποιεί μια δύο σταδίων σύγκριση σχημάτων μεταξύ των πρωτότυπων χαρακτήρων και των χαρακτήρων της εικόνας εισαγωγής. Το tesseract ενσωματώνει την κατάτμηση των της εισαγόμενης εικόνας σε μεμονωμένους χαρακτήρες μαζί με την αναγνώρισή τους χρησιμοποιώντας οπισθοδρόμηση για την περίπτωση που το επόμενο βήμα καθορίσει ότι η κατάτμηση είναι γεωμετρικά ή γλωσσολογικά αδύνατη [17].

Τα ποσοστά λάθους του tesseract δεν επιτυγχάνουν ακόμα την ίδια απόδοση με τα τρέχοντα εμπορικά συστήματα, αλλά βρίσκεται πολύ κοντά μιας και είναι ακόμα υπό ανάπτυξη. Στην παρούσα φάση το tesseract μπορεί να αναγνωρίσει πολλές διαφορετικές μορφές λατινικών χαρακτήρων και είναι πολύ πιθανό σύντομα να υποστηρίζει και άλλους γλωσσικούς χαρακτήρες [17].

Το tesseract δεν προσπαθεί να προβλέψει τις πιθανές ομοιότητες των χαρακτήρων, τις μεταγενέστερες πιθανότητες ή τις πιθανολογικές δαπάνες κατάτμησης αλλά επιστρέφει

«αποτελέσματα αντιστοιχιών». Επίσης δεν υπολογίζει μια πλήρη γραφική παράσταση κατάτμησης για το εισαγόμενο. Τα παραπάνω περιορίζουν την δυνατότητα να χρησιμοποιηθεί ο tesseract με τα στατιστικά πρότυπα φυσικής γλώσσας, όπως συμβαίνει με το OCRopus [17].



Εικόνα 2.5: Αρχιτεκτονική Tesseract.

Στην Εικόνα 2.5 φαίνονται τα βήματα που πραγματοποιούνται κατά την αναγνώριση χαρακτήρων από μια εικόνα, έγχρωμη ή γκρι. Το πρώτο βήμα είναι μια συνδεδεμένη ανάλυση τμημάτων στην οποία οι περιλήψεις των τμημάτων μαζεύονται όλες μαζί, φωλιάζοντας τες, σε blobs [23].

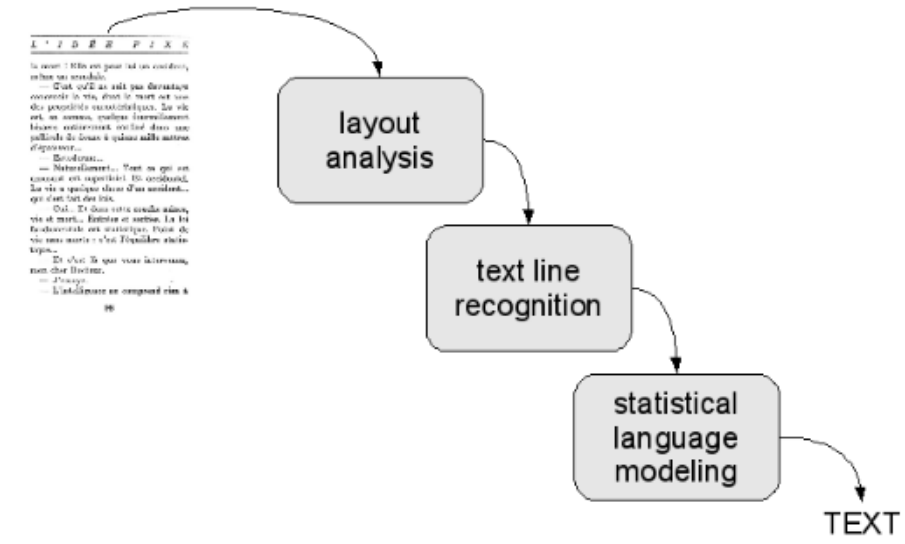
Στη συνέχεια τα blobs οργανώνονται στις γραμμές κειμένων, και οι γραμμές και οι περιοχές αναλύονται σε σταθερό βαθμό ή ανάλογο κείμενο. Οι γραμμές κειμένων διαχωρίζονται σε λέξεις διαφορετικά σύμφωνα με τα διάστημα ανάμεσα στους χαρακτήρες [23].

Τέλος η διαδικασίας της αναγνώρισης γίνεται σε δύο περάσματα. Στο πρώτο πέρασμα γίνεται μια προσπάθεια να αναγνωριστεί κάθε λέξη ξεχωριστά. Κάθε λέξη που είναι ικανοποιητική περνάει σε ένα προσαρμοστικό ταξινομητή (adaptive classifier) ως εκπαιδευτικά δεδομένα. Έτσι στη συνέχεια ο ταξινομητής μπορεί να αναγνωρίσει το κείμενο ακριβέστερα πιο κάτω στην εικόνα. Ωστόσο επειδή ο ταξινομητής μπορεί να εκπαιδεύτηκε πολύ αργά γίνεται ένα δεύτερο πέρασμα ώστε να αναγνωριστούν ακριβέστερα οι γραμμές που βρίσκονται στην αρχή της εικόνας [23].

2.8.2 Βιβλιοθήκη OCRopus

Η βιβλιοθήκη OCRopus είναι ένα καινούργιο, ανοιχτού λογισμικού πρόγραμμα OCR το οποίο δίνει έμφαση στην διαμόρφωση, στην επεκτασιμότητα και στην επαναχρησιμοποίηση και απευθύνεται συγχρόνως στην ερευνητική κοινότητα και σε μετατροπές μεγάλου αριθμού εμπορικών εγγράφων [17].

Οι εμπορικές βιβλιοθήκες OCR είναι παραδοσιακά φτιαγμένες για συγκεκριμένες γραφές και γλώσσες. Αυτό περιορίζει αυτές τις βιβλιοθήκες στο να χρησιμοποιούνται σε μεγάλες ψηφιακές βιβλιοθήκες. Σκοπός του OCRopus είναι να ξεπεράσει αυτούς τους περιορισμούς. Αυτό το καταφέρνει χρησιμοποιώντας πολυγλωσσική αναγνώριση. Για παράδειγμα χρησιμοποιεί unicode κωδικοποίηση και βασίζεται σε HTML και CSS πρότυπα για την παρουσίαση των χαρακτήρων σε μεγάλο αριθμό γλωσσών και γραφών [17].



Εικόνα 2.6: Βήματα λειτουργίας OCRopus.

Στην Εικόνα 2.6 φαίνεται ένα διάγραμμα της δομής του OCRopus. Είναι αυστηρά feed-forward και αποτελείται από τρία βασικά μέρη: ανάλυση σχεδιαγράμματος (layout analysis), αναγνώριση γραμματοσειράς (test line recognition) και στατιστική μοντελοποίηση γλώσσας (statisticla language modeling) [17].

Η ανάλυση σχεδιαγράμματος είναι αρμόδια για τον προσδιορισμό των στηλών κειμένων (text columns), των φραγμών κειμένων (text blocks), των γραμμών κειμένων (text lines), και της σειρά ανάγνωσης (reading order) [17].

Η αναγνώριση γραμματοσειράς είναι αρμόδια για την αναγνώριση του κειμένου που περιλαμβάνεται μέσα σε κάθε γραμμή και την παρουσίαση πιθανών εναλλακτικών αναγνωρίσεων, ως γραφική παράσταση υπόθεσης [17].

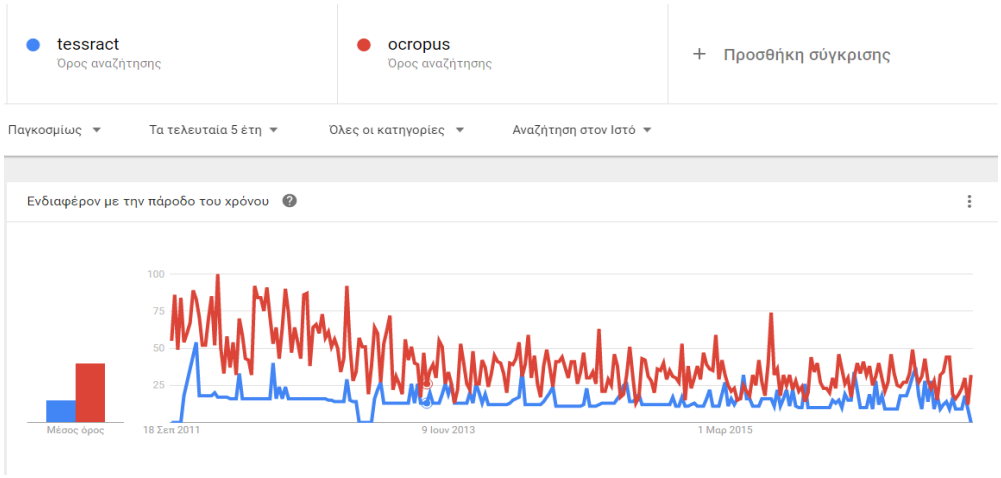
Η στατιστική μοντελοποίηση γλώσσας ενσωματώνει τις εναλλακτικές υποθέσεις αναγνώρισης με την προγενέστερη γνώση για τη γλώσσα, το λεξιλόγιο, τη γραμματική, και την περιοχή του εγγράφου [17].

Η αναγνώριση γραμματοσειράς είτε στηρίζεται στα συστήματα αναγνώρισης γραμμών κειμένων μαύρων κουτιών, συμπεριλαμβανομένων προϋπαρχόντων, είτε σε αναγνώριση που χρησιμοποιεί την υπερ-κατάτμηση (over-segmentation) και την κατασκευή μιας γραφικής παράστασης υπόθεσης. Μια σημαντική πτυχή του συστήματος OCRopus είναι ότι προσπαθεί να προσεγγίσει τα καθορισμένα με σαφήνεια στατιστικά μέτρα σε κάθε διαδικαστικό βήμα [17]. Παραδείγματος χάριν, το προεπιλεγμένο τμήμα ανάλυσης σχεδιαγράμματος προεπιλογής είναι βασισμένο στο γεωμετρικό ταίριασμα μέγιστης πιθανότητας κάτω από ένα γερό γκαουσσιανό (Gaussian) πρότυπο λάθους. Και η MLP-βασισμένη αναγνώριση χαρακτήρων, που ακολουθείται από τη μοντελοποίηση της στατιστικής γλώσσας, προσεγγίζει τις μεταγενέστερες πιθανότητες και τις πιθανότητες κατάτμησης βασισμένες στα στοιχεία κατάρτισης και έπειτα προσπαθεί να βρει τη bayes-βέλτιστη ερμηνεία της εικόνας εισαγωγής ως γραμματοσειρά, λαμβάνοντας υπόψη την προγενέστερη γνώση καταχωρημένη στα στατιστικά γλωσσικά πρότυπα [17].

2.8.3 Σύγκριση

Στην εργασία χρησιμοποιήθηκε το λογισμικό OCRopus διότι παρέχει μεγαλύτερη ευκολία και ευελιξία, με τις βιβλιοθήκες που παρέχει, για να γίνει καλύτερα η εκπαίδευση του μοντέλου.

Σύμφωνα με μια αναζήτηση στην υπηρεσία Google Trends και όπως φαίνεται στην Εικόνα 2.7 διαπιστώνουμε ότι το λογισμικό ocropus είναι πιο δημοφιλές από το tesseract.



Εικόνα 2.7: Σύγκριση μεταξύ Tesseract και Ocropus.

3 Υπηρεσία Διαδικτύου

Σε αυτό το κεφάλαια θα δούμε λίγα λόγια για την υπηρεσία διαδικτύου (web service) και που χρησιμοποιούνται. Ακόμα θα δούμε για τις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση όπως το SOAP και το REST καθώς και μια σύγκριση μεταξύ τους και τέλος θα δούμε λίγα στοιχεία για τα web application frameworks και το flask.

Η υπηρεσία διαδικτύου είναι μια διεπαφή η οποία είναι προσβάσιμη από ένα λογισμικό μέσω ενός δικτύου και δημιουργείται χρησιμοποιώντας βασικές τεχνολογίες διαδικτύου. Γενικά αν μια εφαρμογή μπορεί να προσεγγιστεί μέσω δικτύου χρησιμοποιώντας ένα συνδυασμό από πρωτόκολλα όπως http, xml, smtp τότε είναι υπηρεσία διαδικτύου [8].

Εξαιτίας του τρόπου που φτιάχνονται τα web services μπορούν λειτουργούν σε όλες τις πλατφόρμες ανεξαρτήτου γλώσσας ή λειτουργικού συστήματος. Για παράδειγμα δεν έχει σημασία αν ο server είναι εγκατεστημένος σε linux και η υπηρεσία τρέχει σε windows [8].

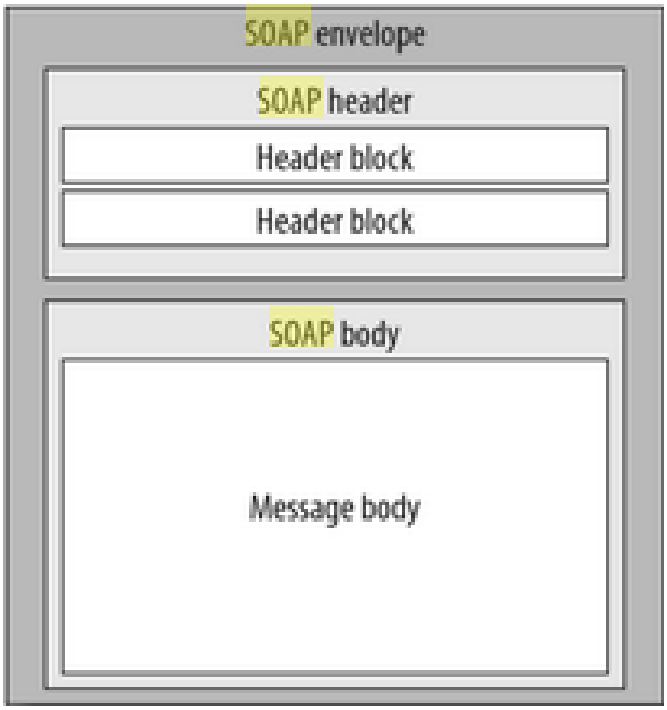
Οι υπηρεσίες διαδικτύου είναι πλαίσια μηνυμάτων. Η μόνη απαίτηση που ζητείται από την υπηρεσία είναι ότι πρέπει να είναι δυνατόν να στέλνει και να λαμβάνει μηνύματα χρησιμοποιώντας στα βασικά πρωτόκολλα του Διαδικτύου [8].

3.1 SOAP

Η θέση του SOAP στην τεχνολογία του web service είναι ως ένα πρωτόκολλο για την τυποποίηση των πακέτων των μηνυμάτων που στέλνονται από το λογισμικό. Η προδιαγραφή καθορίζει ένα απλό xml φάκελο για την πληροφορία που διακινείται και ένα σετ κανόνων για την μετάφραση την εφαρμογής και τα δεδομένα σε xml παρουσίαση. Ο σχεδιασμός του SOAP το καθιστά κατάλληλο για μεγάλη ποικιλία εφαρμογών για μηνύματα [8].

Το SOAP είναι XML, δηλαδή είναι μια εφαρμογή των προδιαγραφών του xml. Βασίζεται σε μεγάλο βαθμό σε πρότυπα xml όπως xml schema και xml Namespaces για τον ορισμό και την λειτουργία του [8].

Ένα μήνυμα SOAP αποτελείται από ένα φάκελο ο οποίος περιέχει μια προαιρετική κεφαλίδα (header) και ένα υποχρεωτικό σώμα (body) όπως φαίνεται στην Εικόνα 3.1. Η κεφαλίδα περιέχει πληροφορία η οποία σχετίζεται με τον τρόπο με το οποίο το μήνυμα επεξεργάζεται. Αυτό συμπεριλαμβάνει ρυθμίσεις δρομολόγησης και παράδοσης, πιστοποίηση ή εξουσιοδότηση και πλαίσιο συναλλαγής. Το σώμα περιέχει το πηγαίο μήνυμα προς παράδοση και επεξεργασία. Οτιδήποτε μπορεί να παρουσιαστεί σε μορφή xml μπορεί να συμπεριληφθεί στο σώμα [8].



Εικόνα 3.1: Δομή μηνύματος SOAP [8]

Η κεφαλίδα, ένα προαιρετικό μέρος ενός μηνύματος SOAP μπορεί να εμφανιστεί πολλαπλάσιοι χρόνοι, εάν εμφανίζεται καθόλου. Η σημασιολογία ή η έννοια κάθε επιγραφής καθορίζεται χαρακτηριστικά μέσα σε ένα σχετικό σχήμα XML. Το όνομα του κορυφαίου στοιχείου στο φραγμό επιγραφών μπορεί να χρησιμοποιηθεί για να χαρτογραφήσει το φραγμό

στη σημασιολογία που καθορίζεται στο σχετικό σχήμα. Το στοιχείο επιγραφών είναι ένα παιδί του φακέλου [20].

Οι κεφαλίδες είναι ο κύριος μηχανισμός από τον οποίο το SOAP μπορεί να επεκταθεί ώστε να περιλαμβάνει πρόσθετα χαρακτηριστικά γνωρίσματα και λειτουργίες όπως η ασφάλεια, οι συναλλαγές, και άλλες ιδιότητες που σχετίζονται με την καλύτερη ποιότητα της υπηρεσίας [20].

Το body περιέχει xml δεδομένα προκαθορισμένα από την εφαρμογή τα οποία ανταλλάσσονται στο μήνυμα SOAP. Το σώμα πρέπει να περιλαμβάνεται μέσα στο φάκελο και πρέπει να ακολουθεί οποιεσδήποτε οδηγίες περιέχει η κεφαλίδα του μηνύματος. Το σώμα ορίζεται ως ένα παιδί του φακέλου, και η σημασιολογία για το σώμα καθορίζεται από το σχετικό σχήμα SOAP [20].

3.2 REST

Το REST είναι μια αρχιτεκτονική που χρησιμοποιείται για την δημιουργία εφαρμογών διαδικτύου. Τις αρχές του REST τις περιέγραψε πρώτος ο Roy Fielding στην διπλωματική για το διδακτορικό του [9]. Παρακάτω παρουσιάζονται εν συντομία οι βασικές αρχές:

- Κάθε πόρος εφοδιάζεται με ένα μοναδικό κλειδί όπως το URI (uniform resource identifier).
- Ένωση μεταξύ των πόρων, δημιουργία σχέσεων μεταξύ τους.
- Χρήση βασικών μεθόδων (HTTP, XML, πολυμέσα).
- Οι πόροι μπορεί να έχουν πολλαπλές παρουσιάσεις οι οποίες καθρεφτίζουν διαφορετικές καταστάσεις.
- Οι επικοινωνίες πρέπει να είναι χωρίς κατάσταση με χρήση HTTP [9].

Η αρχιτεκτονική REST συχνά αναφέρεται ως η αρχιτεκτονική του διαδικτύου. Οι περισσότερες εφαρμογές παρουσιάζουν τα χαρακτηριστικά του REST όταν δημιουργούν εφαρμογές διαδικτύου. Κατά τη χρήση της REST χρησιμοποιείται μια προσέγγιση

πελάτη/δρομολογητή για να διαχωρίσει τον χρήστη από το χώρο αποθήκευσης των δεδομένων [9].

Ένα βασικό στοιχείο της αρχιτεκτονικής είναι η έννοια του πόρου. Οι δρομολογητές περιέχουν τους πόρους και οι χρήστες τους καταναλώνουν. Κάθε πληροφορία που μπορεί να ονομαστεί μπορεί να είναι πόρος όπως για παράδειγμα ένα έγγραφο, μια εικόνα ή ακόμα και ο σημερινός καιρός [9].

Κάθε πόρος έχει ένα αναγνωριστικό, το URI, και μπορεί επίσης να έχει μια σχετική παρουσίαση όπως για παράδειγμα ένα έγγραφο μπορεί να είναι HTML έγγραφο ή μια εικόνα να είναι JPEG ή ο καιρός να παρουσιάζεται με ένα xml αρχείο. Επίσης μπορεί να περιλαμβάνει περεταίρω πληροφορίες (metadata),όπως το τύπο του αρχείου, την τελευταία ημερομηνία επεξεργασίας του αρχείου [9].

Η αρχιτεκτονική REST επωφελείται από την χρήση του πρωτοκόλλου http με την εφαρμογή εννοιών REST σε εφαρμογές που τρέχουν σε περιβάλλον διαδικτύου. Τα πρωτόκολλο http ευθύνεται σε μεγάλο βαθμό για την επιτυχία του διαδικτύου και θεωρείται το βασικότερο πρωτόκολλο σήμερα. Χρησιμεύει για την πρόσβαση σε δεδομένα όχι μόνο σελίδων html αλλά κάθε είδους πολυμέσου [9].

Όταν γίνεται επεξεργασία πόρων με την http, καθορίζεται ένα αναγνωριστικό για τον πόρο μαζί με την ενέργεια που θα γίνει πάνω στον πόρο. Το αναγνωριστικό ονομάζεται με την χρήση uri και η ενέργεια με ρήμα http. Τα ρήματα Http βοηθούν ώστε να υπάρχει μια ομοιόμορφη διεπαφή για την αλληλεπίδραση με τον πόρο το οποίο είναι βασική αρχή της REST αρχιτεκτονικής. Στον Πίνακα 3.1 φαίνονται περιληπτικά μερικά τέτοια ρήματα και πως χρησιμοποιούνται με την χρήση REST [9].

Verb	**Περιγραφή**
GET	Ανακτά ένα πόρο με αναγνωριστικό URI.
POST	Στέλνει ένα πόρο στον σέρβερ. Ενημερώνει τον πόρο στην τοποθεσία που είναι δηλωμένη από το URI.

PUT	Στέλνει ένα πόρο στον σέρβερ ώστε να αποθηκευτεί στην τοποθεσία που είναι δηλωμένη από το URI.
DELETE	Διαγράφει ένα πόρο με αναγνωριστικό URI.
HEAD	Ανακτά τα περαιτέρω δεδομένα (metadata) ενός πόρου με αναγνωριστικό URI.

Πίνακας 3.1: Ρήματα http και χρήση τους με REST.

3.3 Διαφορές REST - SOAP

Και οι δύο αρχιτεκτονικές έχουν πλεονεκτήματα και ξεχωριστά και το ένα με το άλλο.

Το SOAP είναι η πιο βαριά επιλογή για την δημιουργία εφαρμογών. Έχει τα ακόλουθα πλεονεκτήματα [21].

- Είναι ανεξάρτητο από την γλώσσα την πλατφόρμα και την μεταφορά (το REST χρειάζεται την http).
- Λειτουργεί καλά σε διανεμημένα επιχειρησιακά περιβάλλοντα.
- Είναι τυποποιημένο.
- Παρέχει σημαντική προ-κατασκευής επεκτασιμότητα με τη μορφή του WS προτύπου .
- Ενσωματωμένες μέθοδοι για χειρισμό λαθών.
- Αυτοματοποίηση όταν χρησιμοποιείται με συγκεκριμένες γλωσσικά προϊόντα.

Το REST είναι πιο εύκολο στη χρήση και είναι και πιο ευέλικτο. Έχει τα παρακάτω πλεονεκτήματα [21].

- Μικρότερος χρόνος εκμάθησης.
- Αποδοτικό (το SOAP χρησιμοποιεί xml για όλα τα μηνύματα, το REST μπορεί αν χρησιμοποιήσει μικρότερη μορφή μηνύματος).
- Ταχύτατο (δεν χρειάζεται εκτενής επεξεργασία).

- Κοντινός σχεδιασμός με άλλες τεχνολογίες διαδικτύου.

Σύμφωνα με τις διαφορές των δυο τύπων web services επιλέξαμε να υλοποιήσουμε την εφαρμογή χρησιμοποιώντας το REST web service διότι έχει ευρύτερη αποδοχή και είναι πιο αποδοτικό σε τεχνολογίες κινητών συσκευών [23].

3.4 Web Application Framework-FLASK

Ένα πλαίσιο ιστού (web framework) ή πλαίσιο εφαρμογής ιστού (web application framework) είναι ένα πλαίσιο λογισμικού που είναι σχεδιασμένο για να υποστηρίζει την ανάπτυξη μια εφαρμογής ιστού (web application) συμπεριλαμβανομένων των υπηρεσιών ιστού (web service), πόρων ιστού (web resources) και web API. Τα πλαίσια ιστού στοχεύουν να διευκολύνουν τα γενικά έξοδα που συνδέονται με τις κοινές δραστηριότητες που εκτελούνται στην ανάπτυξη ιστού. Για παράδειγμα πολλά πλαίσια ιστού παρέχουν βιβλιοθήκες για πρόσβαση σε βάση δεδομένων, δημιουργία προτύπων, διαχείριση πλαισίων και συνόδων και συχνά προωθούν την επαναχρησιμοποίηση κώδικα. Παρότι συχνά στοχεύουν σε ανάπτυξη δυναμικών ιστοσελίδων μπορούν να χρησιμοποιηθούν και στην ανάπτυξη στατικών ιστοσελίδων [32].

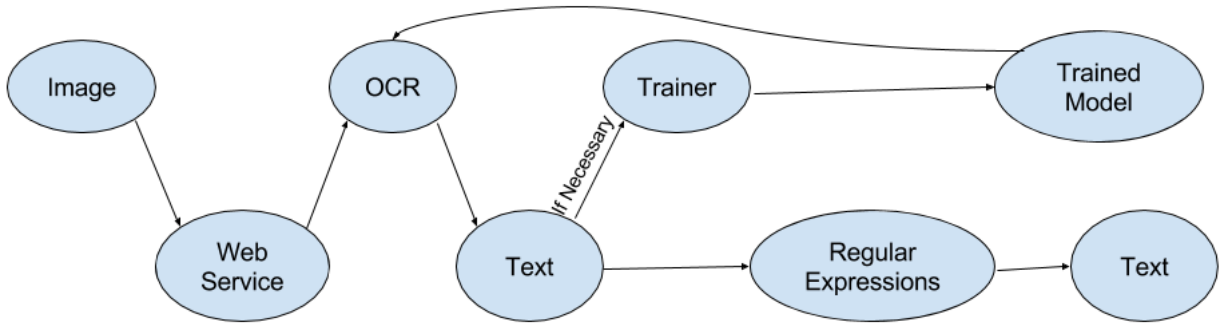
Το Flask είναι ένα μικροπλαίσιο ιστού γραμμένο στην Python και είναι βασισμένο στο εργαλείο Werzeug και στην πρότυπη μηχανή Jinja2. Λέγεται μικροπλαίσιο διότι δεν απαιτεί ιδιαίτερα εργαλεία ή βιβλιοθήκες. Δεν περιέχει κανένα στρώμα αφαίρεσης βάσεων δεδομένων, φόρμα επικύρωσης ή οποιαδήποτε άλλα συστατικά όπου προϋπάρχουσες βιβλιοθήκες τρίτων παρέχουν κοινές λειτουργίες. Ωστόσο το flask υποστηρίζει επεκτάσεις οι οποίες μπορούν να προσθέσουν γνωρίσματα εφαρμογών σαν να ήταν εφαρμοσμένα από το ίδιο. Οι επεκτάσεις υπάρχουν για αντικείμενο-συγγενικούς χαρτογράφους (object-oriented mappers), φόρμα επικύρωσης (form validation), χειρισμό μεταφόρτωσης (upload handling), διάφορες ανοιχτού λογισμικού τεχνολογίες επικύρωσης (open validation technologies), και διάφορα κοινά σχετικα με το πλαίσιο εργαλεία (common framework related tools). Οι επεκτάσεις ενημερώνονται πιο τακτικά από τον πυρήνα του flask [33].

Παρακάτω παρουσιάζονται μερικά χαρακτηριστικά του flask:

- Ενσωματωμένος server ανάπτυξης (development server) και διορθωτής λάθους (debugger).
- Ενσωματωμένη υποστήριξη εξέτασης μονάδας (unit testing).
- Αποστολή αιτήματος RESTful.
- Υποστήριξη για ασφαλή cookies (σύνοδοι πελατών).
- Βασισμένο σε Unicode.

4 ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΓΑΣΙΑΣ - ΥΛΟΠΟΙΗΣΗ

Σκοπός της εργασίας είναι να δημιουργηθεί μια υπηρεσία διαδικτύου ή οποία θα δέχεται σαν όρισμα μια εικόνα και χρησιμοποιώντας οπτική αναγνωρίσει χαρακτήρων θα εξάγει το κείμενο και εφαρμόζοντας πάνω στο κείμενο κανονικές εκφράσεις θα εμφανίζω το συγκεκριμένο κείμενο που επιθυμώ.



Σχήμα 4.1: Σχεδιάγραμμα εργασίας.

Συγκεκριμένα και με βάση το Σχήμα 4.1 η εργασία έχει τις εξής προδιαγραφές:

- Αρχικά έχουμε μια εικόνα η οποία έχει κάποιο κείμενο το οποίο θέλουμε να εξάγουμε για επεξεργασία.
- Στη συνέχεια θέλουμε μια υπηρεσία διαδικτύου που να μας προσφέρει τη δυνατότητα να ανεβάζουμε μια εικόνα μέσω μιας διεπαφής.
- Αφού ανεβάσουμε την εικόνα στην υπηρεσία, θέλουμε να εξάγουμε το κείμενο. Για να γίνει αυτό πρέπει να εφαρμόσουμε μεθόδους οπτικής αναγνώρισης χαρακτήρων.

- Εφόσον λειτούργησε σωστά η εφαρμογή έχουμε κάποιο κείμενο προς επεξεργασία από την εικόνα. Αυτό το κείμενο μπορεί να το θέλουμε οπως είναι ή μπορεί να θέλουμε συγκεκριμένα στοιχεία.

- Αν θέλουμε συγκεκριμένα στοιχεία χρησιμοποιούμε κανονικές εκφράσεις (regular expressions) και έχουμε σαν αποτέλεσμα ακριβώς το κείμενο που θέλουμε.

- Υπάρχει περίπτωση η αναγνώριση χαρακτήρων να μην λειτούργησε ιδανικά, οπότε το κείμενο να μην είναι αυτό που θέλουμε. Σε αυτή την περίπτωση θέλουμε να χρησιμοποιήσουμε μεθόδους μηχανικής μάθησης πάνω στην εφαρμογή ώστε να εκπαιδευτεί για να μπορεί να βγάζει στο μέλλον σωστά αποτελέσματα.

4.1 Web Service

Αρχικά πρέπει να δημιουργηθεί η υπηρεσία διαδικτύου που θα δίνει την δυνατότητα στον χρήστη να ανεβάζει μια εικόνα. Χρησιμοποιήθηκε python και βιβλιοθήκες flask και rest api.

Ο παραπάνω κώδικας χρησιμοποιεί python και βιβλιοθήκες του flask ώστε να ανεβάσει μια φωτογραφία. Αρχικά ορίζει τις υποστηριζόμενες επεκτάσεις για την φωτογραφία, και τον φάκελο όπου θα την αποθηκεύσει. Με το που δεχθεί την εικόνα την αποθηκεύει στην διεύθυνση που είναι ορισμένη και στη συνέχεια καλεί, με όρισμα την εικόνα, το πρόγραμμα που είναι υπεύθυνο για την αναγνώριση χαρακτήρων.

Το επόμενο κομμάτι του προγράμματος είναι η εφαρμογή οπτικής αναγνώρισης χαρακτήρων πάνω στην εικόνα ώστε να εξάγουμε το κείμενο. Χρησιμοποιείται η open source βιβλιοθήκη ocropy (βλέπε κεφάλαιο 2.5.2) η οποία είναι φτιαγμένη σε γλώσσα python.

Εικόνα 4.1: Σκαναρισμένη Φωτογραφία στην οποία τρέχουμε OCR.

4.2 Διαδικασία επεξεργασίας εικόνας και αναγνώριση χαρακτήρων

```
INFO:  #/home/elias/Documents/ocropy-master/apodixeis/menites2.jpg
INFO:  === /home/elias/Documents/ocropy-master/apodixeis/menites2.jpg 1
INFO:  flattening
INFO:  estimating skew angle
INFO:  estimating thresholds
INFO:  rescaling
INFO:  /home/elias/Documents/ocropy-master/apodixeis/menites2.jpg lo-hi (0.23 0.
88) angle  0.0
INFO:  writing
INFO:
INFO:  ########## /usr/local/bin/ocropus-gpageseg /home/elias/Documents/ocropy
INFO:
INFO:  /home/elias/Documents/ocropy-master/temp/0001.bin.png
INFO:  scale 41.3279566395
INFO:  computing segmentation
INFO:  computing column separators
INFO:  computing lines
INFO:  propagating labels
INFO:  spreading labels
INFO:  number of lines40
INFO:  finding reading order
INFO:  writing lines
```

Σχήμα 4.2: Παράδειγμα εκτέλεσης OCR.

Στο Σχήμα 4.2 φαίνεται ένα παράδειγμα εκτέλεσης ocr πάνω στην εικόνα. Το πρόγραμμα παίρνει σαν όρισμα την εικόνα που στο συγκεκριμένο παράδειγμα είναι η “menites2.jpg” και στη συνέχεια την επεξεργάζεται έτσι ώστε να την φέρει σε μια μορφή που να μπορεί να εξάγει το κείμενο.

Συγκεκριμένα τρέχει το παρακάτω script

```
ocropus-nlbin path/menites2.jpg -o temp

ocropus-gpageseg 'path/????.bin.png'

ocropus-rpred -n 'path/????/??????.bin.png' -m    'path/uw3-700-
model-00100000.pyrnn.gz'

ocropus-hocr 'path/????.bin.png' -o temp.html

ocropus-visualize-results temp
```

Σχήμα 4.3: Script εκτέλεσης OCR.

Το πρώτο βήμα είναι το **binarization** δηλαδή την μετατροπή της εικόνας από χρωματιστή (greyscale) σε ασπρόμαυρη. Το OCRopus για να το κάνει χρησιμοποιεί μια μορφή adaptive thersholding όπου η σχέση μεταξύ λευκού και μαύρου διαφοροποιείται σε όλη την εικόνα πράγμα το οποίο είναι χρήσιμο εάν υπάρχει διαφορά στον φωτισμό της φωτογραφίας. Ακόμα μαζί σε αυτό βήμα βρίσκεται μια διαδικασία skew estimation η οποίο περιλαμβάνει την περιστροφή της εικόνας ώστε οι γραμμές να βρίσκονται σε ευθεία γραμμή.[18]

Για να γίνει το binarization τρέχει η εντολή ocropus-nlbin path/menites2.jpg -o temp όπου ως έξοδο στέλνει στη διεύθυνση προορισμού δύο αρχεία εικόνας:

- `*.bin.png`: binarized εκδοχή της Εικόνας 4.1. (Εικόνα 4.2)
- `*.nrm.png`: μια "επίπεδη" εκδοχή της Εικόνας 4.1, πριν το binarization. Αυτό δεν είναι πολύ χρήσιμο. (Εικόνα 4.3)

Εικόνα 4.3: Binarized image

Εικόνα 4.2: Flattened Image

Το επόμενο βήμα είναι το **segmentation** το οποίο είναι η εξαγωγή κάθε μια γραμμής ξεχωριστά από την εικόνα. Αρχικά υπολογίζει την "κλίμακα" του κειμένου βρίσκοντας συνδεδεμένες συνιστώσες στην binarized εικόνα (αυτά συνήθως είναι μεμονωμένα γράμματα) και στη συνέχεια υπολογίζει τη διάμεσο των διαστάσεών τους.

Στη συνέχεια προσπαθεί να βρει κάθε μια γραμμή ξεχωριστά. Η ακολουθία είναι η εξής:

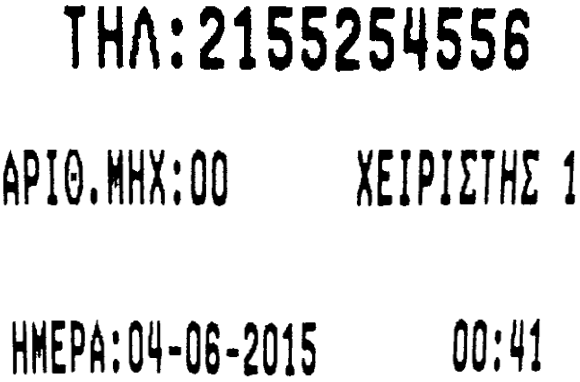
- Αφαιρεί τις συνιστώσες με πολύ μεγάλο ή πολύ μικρό μέγεθος (με βάση την κλίμακα). Αυτά είναι απίθανο να είναι γράμματα.
- Εφαρμόζει το y-derivative ενός Gaussian kernel για να βρει τα πάνω και τα κάτω άκρα των εναπομείνοντων στοιχείων. Στη συνέχεια το θολώνει οριζοντίως για να συνδυάσει τις κορυφές των γραμμάτων που βρίσκονται στην ίδια γραμμή.

- Ότι βρίσκεται μεταξύ της κορυφής και του κάτω μέρους είναι οι γραμμές

Για να γίνει το segmentation τρέχει η εντολή ocropus-gpageseg 'path/????.bin.png' όπου ως έξοδο στέλνει στη διεύθυνση προορισμού μερικά αρχεία εικόνας:

- `*.pseg.png`: κωδικοποιεί το segmentation. Το χρώμα κάθε pixel υποδηλώνει σε ποια γραμμή και στήλη της αρχική εικόνας ανήκει. (Εικόνα 4.4)
- `*.*.bin.png`: Μία εικόνα για κάθε γραμμή που εξήγαγε το πρόγραμμα. (Εικόνα 4.5)

Εικόνα 4.4: Segmentated image.



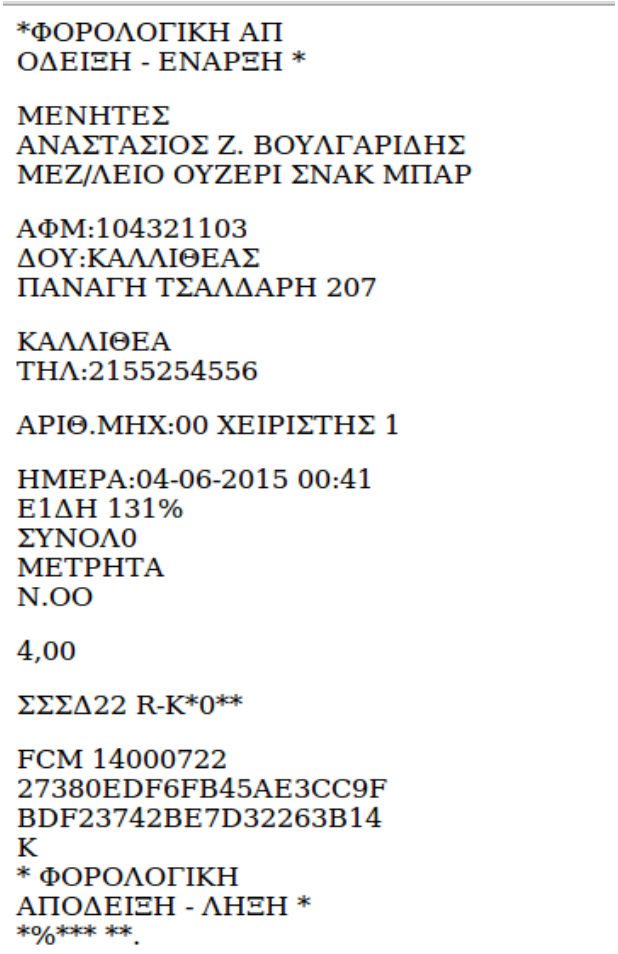
Εικόνα 4.5: Ξεχωριστή εικόνα για κάθε γραμμή της εικόνας.

Αφού ολοκληρωθεί η προετοιμασία της εικόνας περνάμε στην διαδικασίας της αναγνώρισης των γραμμών. Αυτό το κομμάτι είναι δύσκολο διότι κάθε γραμμή μπορεί να είναι διαφορετική. Αυτό συμβαίνει διότι μπορεί κατά το binarization οι γραμμές να βγήκαν πιο σκοτεινές ή πιο φωτεινές ή το skew estimation να μην λειτούργησε σωστά.

Η εντολή για το βήμα αυτό είναι η εξής:

ocropus-rpred -n 'path/????/??????.bin.png' -m 'path/uw3-700-model-00100000.pyrnn.gz'

Αφού αναγνωρίσει κάθε γραμμή ξεχωριστά χρησιμοποιεί το μοντέλο που του έχουμε δώσει και μαντεύει το κείμενο. Σαν έξοδο βγάζει μια εικόνα για την κάθε γραμμή που διαβάζει και ένα αρχείο που περιέχει το κείμενο που έχει μαντέψει.

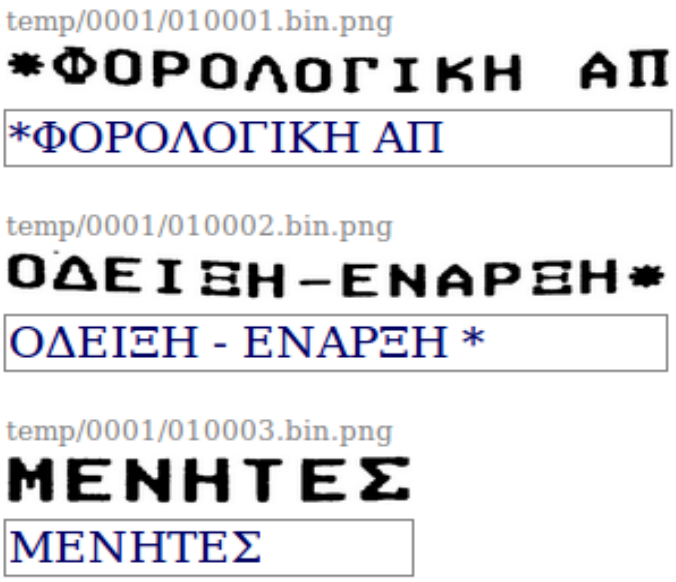


Εικόνα 4.6: Αποτελέσματα OCR

Στην Εικόνα 4.6 βλέπουμε τα αποτελέσματα, δηλαδή το κείμενο που πήρε το πρόγραμμα από την εικόνα που του δώσαμε ως είσοδο.

4.3 Training

Αν το αποτέλεσμα που παίρνουμε δεν είναι ικανοποιητικό για εμάς μπορούμε χρησιμοποιήσουμε μηχανική μάθηση. Μας δύνεται λοιπόν η δυνατότητα να εκπαιδεύσουμε το μοντέλο ώστε μελλοντικά να βγάζει καλύτερα αποτελέσματα. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας επιτηρούμενη μάθηση και χρειάζεται την εικόνα για κάθε γραμμή και το κείμενο από το οποίο θα μάθει.



Εικόνα 4.7: Εισαγωγή σωστού κειμένου.

Για να κάνουμε την εκπαίδευση αρχικά παίρνουμε ένα html αρχείο το οποίο όπως παρουσιάζεται στην Εικόνα 4.7 δίνεται η δυνατότητα να διορθωθούν όποια αποτελέσματα θεωρούνται λανθασμένα. Στη συνέχεια σώζεται το αρχείο με τα διορθωμένα αποτελέσματα και εκτελείται ο κώδικας **ocropus-gtedit extract temp-correction.html** ο οποίος δημιουργεί δύο αρχεία :

- *.png, το οποίο είναι μια εικόνα της γραμμής.
- *.gt.txt είναι ένα αρχείο που περιέχει το διορθωμένο κείμενο

Για να εκπαιδεύσουμε το μοντέλο τρέχουμε το script **ocropus-rtrain 'apod/*/*.bin.png' -d 100000 -o uw3-700-model** το οποίο παίρνει το κείμενο που έχει διαβάσει και το κείμενο που του έχουμε δώσει ως διορθωμένο. Το πρόγραμμα διαβάζει το αρχικό κείμενο και το κείμενο που του έχουμε δώσει και αντιστοιχεί κάθε γράμμα. Με αυτό τον τρόπο μπορεί να προσδιορίσει ακριβώς, τι είναι αυτό που διαβάζει. Για παράδειγμα έχει διαβάσει από την εικόνα το γράμμα "α" το οποίο για το πρόγραμμα είναι εικόνα και όχι κείμενο. Έχοντας το διορθωμένο κείμενο το πρόγραμμα αντιστοιχεί την εικόνα με τον χαρακτήρα "α". Κάνει αυτή την διαδικασία πάρα πολλές φορές, στο συγκεκριμένο παράδειγμα εκατό χιλιάδες φορές, το πρόγραμμα μαθαίνει να αντιστοιχεί κατευθείαν το γράμμα με την εικόνα. Όταν τελειώσει αυτή η διαδικασία παράγεται ένα μοντέλο «**uw3-700-model**» το οποίο περιέχει όλες τις αντιστοιχίσεις που έχει κάνει η εφαρμογή.

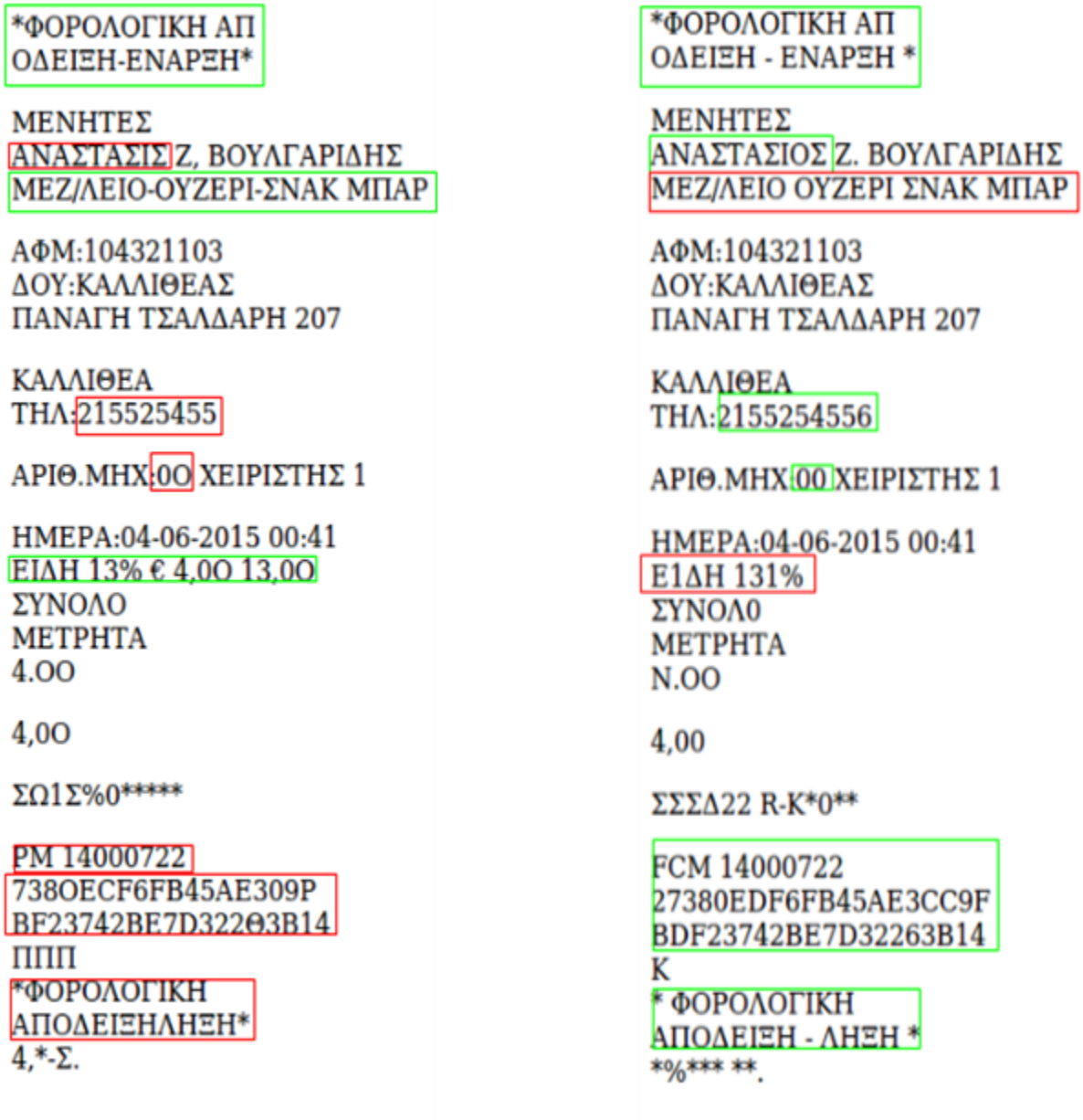
Στην Εικόνα 4.8 φαίνεται ένα παράδειγμα της διαδικασίας εκπαίδευσης του μοντέλου. Το **tru** είναι τα αληθινά δεδομένα, δηλαδή το κείμενο που έχουμε διορθώσει. Το **aln** είναι το κείμενο που αντιστοιχεί το πρόγραμμα στα αληθινά δεδομένα και το **out** είναι το αποτέλεσμα της μάθησης το οποίο γίνεται καλύτερο όσο επαναλαμβάνεται η διαδικασία.

```
   TRU: ΔΟΥ Α'ΚΑΛΛΙΘΕΑΣ
   ALN: ΔΣ
   OUT:
26 21.90 (289, 48) apod/0011/01000b.bin.png
   TRU: 409336000034/1083056
   ALN:
   OUT:
27 75.81 (805, 48) apod/0010/010017.bin.png
   TRU: 9F0513FDD6059379B65C93819FCD4151D13D9AE0
   ALN: 90
   OUT:
28 22.43 (249, 48) apod/0006/01001f.bin.png
   TRU: 26,50 ΕΥΡΩ
   ALN: 2Ω
   OUT:
29 21.22 (211, 48) apod/0010/010013.bin.png
   TRU: Μετρητά
   ALN: M~
   OUT:
30 38.53 (415, 48) apod/0016/010004.bin.png
   TRU: ΑΝΑΣΤΑΣΙΟΣ Ζ. ΒΟΥΛΓΑΡΙΔΗΣ
   ALN: Σ
   OUT:
```

Εικόνα 4.8: Παράδειγμα Εκπαίδευσης μοντέλου.

Όταν τελειώσει η διαδικασία επαναλαμβάνουμε την εκτέλεση ocr πάνω στην εικόνα χρησιμοποιώντας το μοντέλο που παράχθηκε από τη διαδικασία της εκπαίδευσης. Εάν έχει γίνει επαρκής αριθμός επαναλήψεων κατά τη διαδικασία της μάθησης τότε τα αποτελέσματα της αναγνώρισης χαρακτήρων πρέπει να είναι σωστότερα από την προηγούμενη εκτέλεση.

Παρακάτω βλέπουμε δύο παραδείγματα με αποτελέσματα πριν και μετά τη διαδικασία της εκπαίδευσης. Στην εικόνα 4 9 βλέπουμε τα αποτελέσματα πριν την εκπαίδευση και σε κόκκινο πλαίσιο βλέπουμε το κείμενο το οποίο είναι λανθασμένο. Στην εικόνα 4 10 βλέπουμε τα καινούργια αποτελέσματα χρησιμοποιώντας το μοντέλο που φτιάξαμε με την εκπαίδευση και σε πράσινο πλαίσιο βλέπουμε ότι το κείμενο έχει διορθωθεί σύμφωνα με το κείμενο που το δώσαμε.



Εικόνα 4.9: Αριστερά: αρχική εικόνα, Δεξιά: εικόνα μετά την εκπαίδευση

Παρακάτω παρουσιάζονται μερικά αποτελέσματα. Αριστερά είναι η αρχική εικόνα και δεξιά το αποτέλεσμα που έβγαλε η αναγνώριση χαρακτήρων.

ΝΟΜΙΜΗ ΑΠΟΔΕΙΞΗ - ΕΝΑΡΞΗ

ΑΝΕΔΗΚ ΚΡΗΤΙΚΟΣ Α.Ε SUPER MARKET

ΕΔΡΑ: ΚΥΨΕΛΗ ΑΙΓΙΝΑΣ

ΑΦ.Μ:094247924
Υ/Α ΡΙΔΟΥ 102 ΚΑΛΛΙΘΕ
2 /0002
:210-9599600

ΑΠΟΔΕΙΞΗ ΛΙΑΝ Σ ΠΩΛΗΣΗΣ

ΔΕΛΤΑ ΓΑΛΑ ΦΡΕΣΚΟ IGHT 50 0,76 13,00%
ΜΠΡΟΚΟΛΑ
0,715 x 2,14 1,53 13,00%
ΚΡΕΜΜΥΔΙΑ ΞΕΡΑ
0,895 x 0,7 0,69 13,00%
ΣΥΝΟΛΟ
ΜΕΤΡΗΤΑ
ΡΕΣΤΑ
ΠΩΛΗΘΕΝΤΑ ΕΙΔΗ :
2,98ΕΥΡΩ
5,00
2,02

ΑΡ.ΝΟΜ.ΑΠΟΔ.– 215 19/05/2015 18:27:43
Α/Α 348
/ΜΑΑΖ12002042
ΝΟΜΙΜΗ ΑΠΟΔΕΙΞΗ - ΛΗΞΗ
ΑΝ ΕΙΧΑΤΕ ΚΑΡΤΑ ΜΕΛΟΥΣ
ΘΑ ΚΕΡΔΙΖΑΤΕ 1 ΠΟΝΤΟΥΣ

Ε Π** –ΦΒΜΦ *ΚΡΡΝ*
* - ΜΛ Φ* ρ Κ9Μ

Εικόνα 4.10:Αριστερά: Αρχική εικόνα, Δεξιά: Αποτέλεσμα OCR

*ΦΟΡΟΛΟΓΙΚΗ ΑΠ
ΟΔΕΙΞΗ-ΕΝΑΡΞΗ*

ΜΕΝΗΤΕΣ
ΑΝΑΣΤΑΣΙΟΣ Ζ. ΒΟΥΛΓΑΡΙΔΗΣ
ΜΕΖ/ΛΕΙΟ-ΟΥΖΕΡΙ-ΣΝΑΚ ΜΠΑΡ
ΑΦΜ:104321103
ΔΟΥ:ΚΑΛΛΙΘΕΑΣ
ΠΑΝΑΓΗ ΤΣΑΛΔΑΡΗ 207
ΚΑΛΛΙΘΕΑ
ΤΗΛ:2155254556

ΑΡΙΘ.ΜΗΧ:00 ΧΕΙΡΙΣΤΗΣ 1
ΗΜΕΡΑ:03-06-2015 22:43
- - - - - - - - - - - - - - -
7 x 0,40
ΕΙΔΗ 13% € 2,80 13,00%
ΕΙΔΗ 23% € 3,50 23,00%
ΕΙΔΗ 23% € 7,00 23,00%
ΕΙΔΗ 13% € 4,00 13,00%
- - - - - - - - - - - - - - -
ΣΥΝΟΛΟ € 17,30
ΜΕΤΡΗΤΑ 17,30
- - - - - - - - - - - - - - -
ΠΟΣ/ΤΑ:10,000
ΗΜ.ΑΠΟΔ:0004 ΠΡ.ΑΠΟΔ:0000590

FCM 14000722
2D001E088CF11D313113
374A49AF993D5B5F8DE3

*ΦΟΡΟΛΟΓΙΚΗ
ΑΠΟΔΕΙΞΗ-ΛΗΞΗ*

*ΦΟΡΟΛΟΓΙΚΗ ΑΠ
ΟΔΕΙΞΗ - ΕΝΑΡΞΗ *

ΜΕΝΗΤΕΣ
ΑΝΑΣΤΑΣΙΟΣ Ζ. ΒΟΥΛΓΑΡΙΔΗΣ
ΜΕΖ/ΛΕΙΟ ΟΥΖΕΡΙ ΣΝΑΚ ΜΠΑΡ

ΑΦΜ:104321103
ΔΟΥ:ΑΛΛΙΘΕΑΣ
ΠΑΝΑΓΗ ΤΣΑΛΔΑΡΗ 207

ΚΑΛΛΙΘΕΑ
ΤΗΛ:2155254556
ΑΡΙΘ.ΜΗΧ:00 ΧΕΙΡΙΣΤΗΣ 1
ΗΜΕΡΑ:03-06-2015 22:43
7 x
040

ΕΙΔΗ 13%
ΕΙΔΗ 23%
ΕΙΔΗ 23%
ΕΙΔΗ 1313.00
ΣΥΝΟΛΟ ΕΟ
ΜΕΤΡΗΤΑ
ΠΟΣ/ΤΑ:10,000
17,30

ΗΜ.ΑΠΟΔ:0004 ΠΡ.ΑΠΟΔ:0000590

FCM 14000722
2D001E088CF11D313113
374A49AF993D5B5F8DE3
ΙΕ ΥΤΝ ΩΓ
ΒΟΟΣ.ΩΑ
*ΦΟΡΟΔΜΟΙΚΗ
ΦΔ/ΜΕΙΞΗ - ΛΗΞΗ *
ΟΜα***/-Μ

Εικόνα 4.11: Αριστερά: Αρχική εικόνα, Δεξιά: Αποτέλεσμα OCR

ΣΕΙΡΑ
Α/Β ΒΑΣΙΛΟΠΟΥΛΟΣ Α.Ε
ΑΦΜ:094025817 ΔΕΥ ΦΑΕ ΑΘΗΝΩΝ
Κατ-στημο:Καλλιθ- (ΜΕΤΡΟ
Πατρι-ρχου Γρορο Ε 75
ΤΛ.2109510025

1374
ΑΡΙΘΜΟΣ ΠΑΡΑΣΤΑΤΙΚΟΥ 54661
ΤΑΜΙΑΣ

ΑΠΟΔΕΙΞΗ ΛΙΑΝΙΚΗΣ ΠΩΛΗΣΗΣ
0,700 KG X 0,62
ΚΡΕΜΜΥΔΙΑ ΞΕΡΑ ΕΛΛ
ΚΟΥΚΟΥΝΑΡΙ 50ΓΡ
ΣΟΚΟΛΑΤΑ ΑΜΥΓΔΑΛΟΥ
0,675 KG X 1,20
ΤΟΜΑΤΕΣ ΧΥΜΑ ΕΛΛ.
1,355 KG 1,39
ΜΕΛΙΤΖ.ΦΛΑΣΚΕΣ ΕΛΛ
ΣΥΝΟΛΟ

ΣΥΝΟΛΟ ΕΙΔΩΝ *5*
ΜΕΤΡΗΤΑ
ΡΕΣΤΑ

ΑΝΑΛΥΣΗ ΦΠΑ
0,43 13,00%
1,96 13,00%
1,30 13,00%
0,81 13,00%
1,88 13,00%
6ΘΕΥΡΩ

6,50
0,12

ΣΥΝΤ. ΦΠΑ ΚΑΘΑΡΗ ΑΞΙΑ ΑΞΙΑ ΦΠΑ
13,00%
5,65
0,7X

ΗΜΕΡ: 04/05/2015 ΩΡΑ: 20:24:24 POS: 1
ΑΡ. ΣΥΝΑΛ. - 00652

Η ΧΑΡΑ ΝΑ ΚΕΡΔΙΖΕΙΣ
ΚΑΘΕ ΦΟΡΑ ΠΟΥ ΨΩΝΙΖΕΙΣ

Εικόνα 4.12: Αριστερά: Αρχική εικόνα, Δεξιά: Αποτέλεσμα OCR

4.4 Εφαρμογή κανονικών εκφράσεων

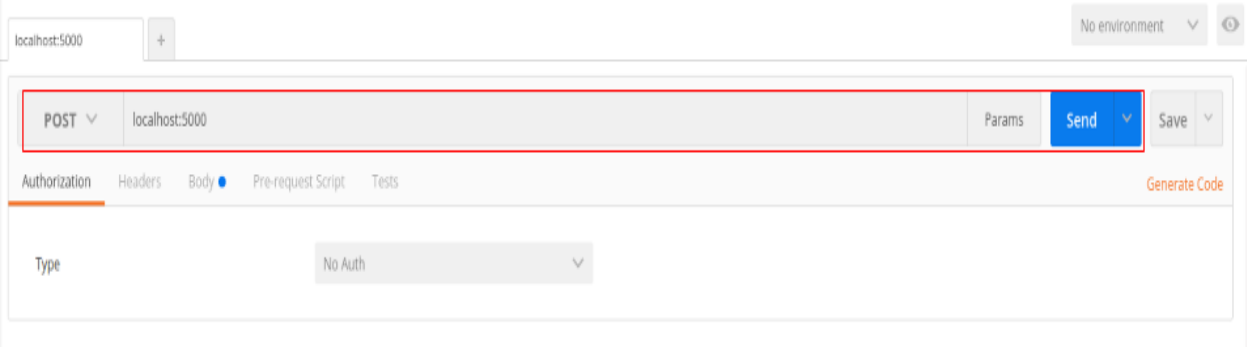
Εφόσον έχει γίνει η εκπαίδευση και έχει από την αρχή η αναγνώριση χαρακτήρων χρειάζεται να εφαρμόσουμε κανονικές εκφράσεις στην εικόνα ώστε να πάρουμε συγκεκριμένα αποτελέσματα. Μια απόδειξη λιανικής πώλησης αναγράφει πολλά δεδομένα όπως τον ΦΠΑ, το ΑΦΜ της εταιρίας , τα προϊόντα και τις τιμές τους ωστόσο αναγράφει και δεδομένα που είναι συνήθως ασήμαντα για χρήση όπως τον κωδικό της ταμειακής μηχανή καθώς και το όνομα του ταμεία του καταστήματος. Παρακάτω φαίνονται οι κανονικές εκφράσεις που χρησιμοποιήθηκαν:

- Εμφανίζει τον ΦΠΑ : `[0-9]*[\.,][0-9]*%`
- Εμφανίζει το ΑΦΜ : `ΑΦΜ *[:.]* *\d{9}`

- Εμφανίζει την ημερομηνία : `\d{1,2} +[A-Ωα-ωάίέόύ.]+ \d{1,4}` αν είναι ολογράφως και `\d{1,2}[\/-] *\d{1,2}[\/-] *\d{1,4} *` αν είναι αριθμητική.
- Εμφανίζει την ώρα : `\d{2}:\d{2}:*\d{0,2}`
- Εμφανίζει την τιμή των προϊόντων : `/€\d{1,}[.,]*\d{2}/`
- Εμφανίζει την συνολική τιμή : `ΣΥΝΟΛΟ [Α-Ω]* *[€ ]*\d{1,}[.,]*\d{2}`

4.5 Δοκιμή Web Service

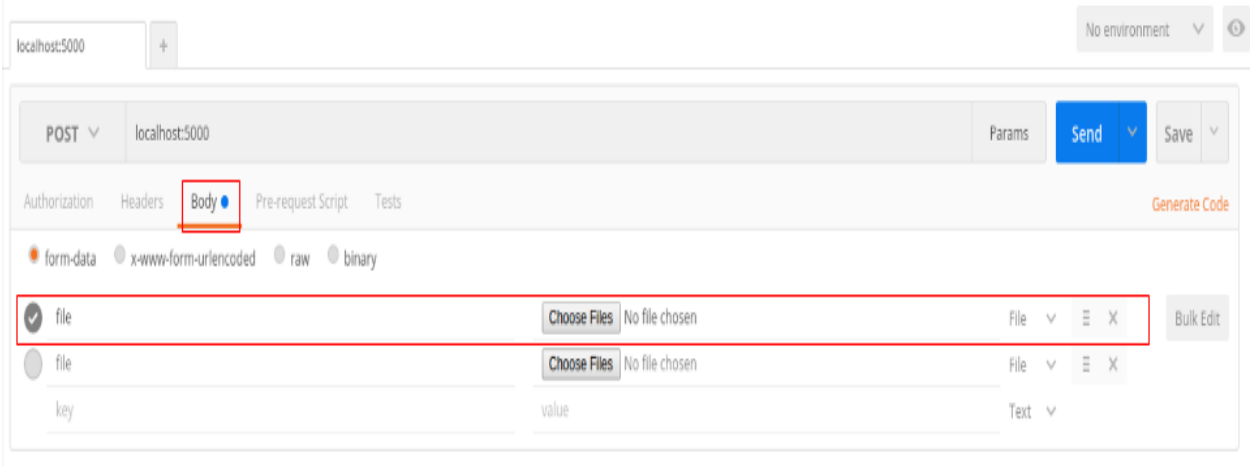
Για τη δοκιμή του rest web service χρησιμοποιήθηκε το εργαλείο**Postman**. Παρακάτω παρουσιάζεται ένα παράδειγμα χρησιμοποίησης του εργαλείου αυτού.



Εικόνα 4.13: Διεπαφή Postman.

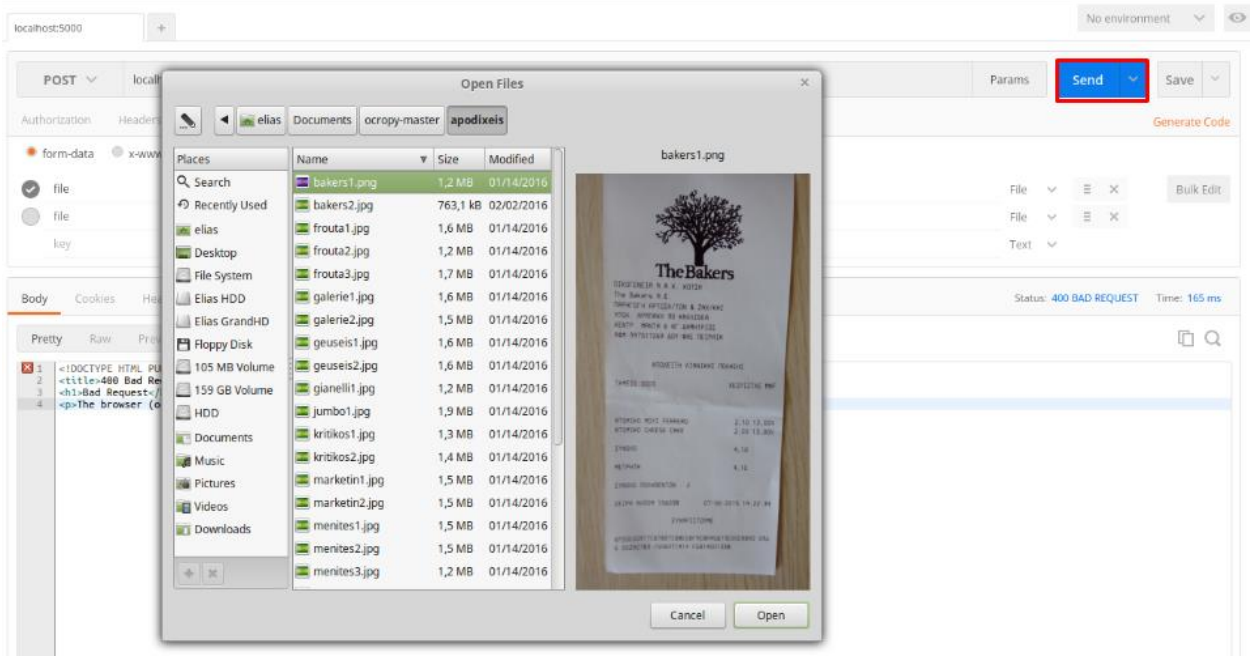
Στην Εικόνα 4.13 βλέπουμε το κομμάτι του εργαλείου που χρησιμοποιείται για να στέλνει μηνύματα στην εφαρμογή ιστού. Όπως φαίνεται μέσα στο πλαίσιο υπάρχουν τρείς παράμετροι που πρέπει να επεξεργαστούν. Αρχικά επιλέγουμε το ρήμα που θέλουμε να χρησιμοποιήσουμε. Για την συγκεκριμένη εφαρμογή πρέπει να στείλουμε ένα μήνυμα και γι' αυτό επιλέγουμε το **POST**.

Στη συνέχεια πρέπει να δώσουμε την διεύθυνση που τρέχει το web service. Στο συγκεκριμένο παράδειγμα το service τρέχει στον τοπικό υπολογιστή οπότε δίνουμε **localhost** και τη πόρτα που εδώ είναι 5000.



Εικόνα 4.14*: Καρτέλα για ανέβασμα εικόνας.*

Στο συγκεκριμένο παράδειγμα θέλουμε να ανεβάσουμε μια εικόνα από τον υπολογιστή για να την επεξεργαστούμε. Όπως φαίνεται στην Εικόνα 4.14 επιλέγουμε την καρτέλα **Body** όπως φαίνεται στο πλαίσιο. Στη συνέχεια πρέπει να επιλέγουμε την εικόνα που θέλουμε να ανεβάσουμε. Το όνομα **file** είναι το όνομα της μεταβλητής που έχουμε ορίσει στο web service. Πατώντας choose files μπορούμε να επιλέξουμε την εικόνα που θέλουμε να χρησιμοποιήσουμε όπως φαίνεται στην Εικόνα 4.15.



Εικόνα 4.15: Επιλογή εικόνας.

4.6 Διαδικασία εγκατάστασης Ocropus

Για να γίνει εγκατάσταση του λογισμικού χρειάζεται να τρέξουν κάποιες εντολές σε περιβάλλον linux.

- `sudo apt-get install $(cat PACKAGES)`
- `wget -nd http://www.tmbdev.net/en-default.pyrnn.gz`
- `mv en-default.pyrnn.gz models/`
- `sudo python setup.py install`

4.7 Προβλήματα κατά την εγκατάσταση

Κατά την εγκατάσταση των εργαλείων και των βιβλιοθηκών δημιουργήθηκαν κάποια προβλήματα. Αρχικά η εφαρμογή έτρεξε στο λειτουργικό σύστημα Ubuntu στο οποίο κατά την εκτέλεση του προγράμματος γινόταν κλήση μιας μεθόδου από μια συγκεκριμένη βιβλιοθήκη της python, τη matplotlib. Παρουσιαζόταν λάθος κατά την κλήση καθώς ο πρόγραμμα δεν έβρισκε την βιβλιοθήκη στο σύστημα. Ωστόσο παρότι έγινε η εγκατάσταση της βιβλιοθήκης αυτής και χωρίς να παρουσιαστεί κάποιο πρόβλημα το πρόγραμμα συνέχιζε να ζητάει την βιβλιοθήκη. Το πρόβλημα αυτό λύθηκε με την χρησιμοποίηση διαφορετικού λειτουργικού συστήματος, του Linux Mint. Κατά την εκτέλεση του προγράμματος στο λειτουργικό αυτό ζητήθηκαν κάποιες βιβλιοθήκες οι οποίες εγκαταστάθηκαν και το πρόγραμμα έτρεξε επιτυχώς.

5 ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

5.1 Δυνατότητες εφαρμογής

Στο πλαίσιο της εργασίας δημιουργήθηκε μια εφαρμογή για οπτική αναγνώριση χαρακτήρων και μια εφαρμογή διαδικτύου για την χρησιμοποίηση της εφαρμογής αυτής. Η υπηρεσία ιστού δίνει την δυνατότητα στον χρήστη να ανεβάσει μια φωτογραφία η οποία αποθηκεύεται και δίνεται σαν όρισμα στην εφαρμογή αναγνώρισης χαρακτήρων. Η εφαρμογή αυτή λαμβάνει φωτογραφίες αποδείξεων και εξάγει δεδομένα όπως την τιμή, τα προϊόντα το ΦΠΑ και άλλα.

Ακόμα η εφαρμογή δίνει την δυνατότητα στην δημιουργία ενός μοντέλου που χρησιμοποιείται για την εκπαίδευση της εφαρμογής. Το μοντέλο δημιουργείται από τον χρήστη ο οποίος εισάγει διορθωμένα δεδομένα στην εφαρμογή. Αυτά τα δεδομένα μπορεί να είναι μια απλή διόρθωση του αποτελέσματος που έβγαλε η εφαρμογή. Ωστόσο υπάρχει και η δυνατότητα εισαγωγής κειμένου από το οποίο το μοντέλο μπορεί όχι μόνο να μάθει να αναγνωρίζει καλύτερα τους χαρακτήρες αλλά και να αναγνωρίζει τη γραμματοσειρά. Στη συνέχεια το μοντέλο χρησιμοποιείται για να γίνει καλύτερη πρόβλεψη των χαρακτήρων με βάση τις διορθώσεις του χρήστη. Η διαδικασία της δημιουργίας του μοντέλου εξαρτάται από τους πόρους που μπορεί να προσφέρει ο υπολογιστής και ο χρόνος περαίωσης διαφέρει και μπορεί να είναι και πολύ μεγάλος.

Η εφαρμογή δημιουργείται με εργαλεία και βιβλιοθήκες ανοιχτού λογισμικού, όπως το OCRopus και το flask. Αυτό δίνει την δυνατότητα περαιτέρω επεξεργασίας όπως η εισαγωγή ή η αλλαγή μιας ή περισσότερων βιβλιοθηκών και δυνατοτήτων. Τέτοια επεξεργασία μπορεί να γίνει για την αλλαγή του τρόπου σάρωσης της εικόνας. Για παράδειγμα στην εργασία έγινε επεξεργασία ώστε να υποστηρίζονται Ελληνικοί χαρακτήρες ώστε να μπορεί να γίνει και εκπαίδευση με εισαγωγή διορθωμένου Ελληνικού κειμένου.

Εφόσον χρησιμοποιείται ανοιχτό λογισμικό δίνεται η δυνατότητα δημιουργίας open data εφαρμογής η οποία θα είναι προσβάσιμη από όλους για χρήση και περαιτέρω επέκταση.

5.2 Μελλοντικές επεκτάσεις

Η εφαρμογή δέχεται σαν όρισμα μια εικόνα από μια απόδειξη, κάνει οπτική αναγνώριση χαρακτήρων πάνω της και εξάγει σαν αποτέλεσμα το κείμενο. Θα μπορούσε μελλοντικά να γίνει κάποια επέκταση στη λειτουργία ή στη χρήση της εφαρμογής.

Μια λειτουργία που θα μπορούσε να δημιουργηθεί μελλοντικά είναι μια web εφαρμογή για την τήρηση των οικονομικών στοιχείων των καταναλωτών. Αυτό θα έδινε την δυνατότητα συλλογής δεδομένων από λογαριασμούς, από εφοριακές υποχρεώσεις, περιουσιακά στοιχεία και να βγάζει κάποια συμπεράσματα. Για παράδειγμα μπορεί υπολογίζει τις εκφορές και να βγάζει σαν αποτέλεσμα την επιστροφή που δικαιούται ο καταναλωτής από το κράτος.

Για την περεταίρω διευκόλυνση των παραπάνω δυνατοτήτων μπορεί μελλοντικά να δημιουργηθεί μια εφαρμογή για κινητές συσκευές που να παρέχει την ίδια υπηρεσία μέσω mobile περιβάλλοντος.

Ακόμα η εφαρμογή θα μπορούσε να επεκταθεί και σε άλλα πεδία ώστε να μπορεί να επεξεργάζεται πληθώρα τύπων δεδομένων. Για παράδειγμα να μην διαβάζει μόνο αποδείξεις αλλά να μπορεί να επεξεργάζεται κείμενο από βιβλία και άλλα έγγραφα.

6 ΕΠΙΛΟΓΟΣ

Στην εργασία αυτή δημιουργήθηκε μια εφαρμογή διαδικτύου η οποία λαμβάνει σαν είσοδο μια εικόνα και εφαρμόζει οπτική αναγνώριση χαρακτήρων για να εξάγει το κείμενο από την εικόνα αυτή. Ακόμα έχει τη δυνατότητα χρησιμοποιώντας μεθόδους μηχανικής μάθησης να εκπαιδεύσει ένα μοντέλο ώστε να γίνεται καλύτερη πρόβλεψη των χαρακτήρων στο μέλλον. Για

να δημιουργηθεί η εφαρμογή έγινε έρευνα στη μηχανική μάθηση και στους τρόπους που χρησιμοποιούνται, στα δενδροδιαγράμματα (decision trees) και στα νευρωνικά δίκτυα (neural networks). Όσον αφορά την εφαρμογή διαδικτύου έγινε έρευνα πάνω στο web service και στις τεχνολογίες SOAP και REST. Για την αναγνώριση χαρακτήρων χρησιμοποιήθηκε η βιβλιοθήκη OCRopus που είναι υλοποιημένη σε γλώσσα προγραμματισμού python.

7 ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Line Eikvil, “Optical character Recognition”, December 1993
2. S. Mori C.Y. Suen & K. Yamamoto. Historical Review of OCR research and Development. IEEE Proceedings, special issue on OCR, p. 1029-1057, July 1992.
3. What is OCR and OCR Technology, https://www.abbyy.com/finereader/about-ocr/what-is-ocr/
4. Alex Smola, S.V.N Vishwanathan, Introduction to Machine Learning, 2008.
5. Nils J. Nilsson, Intoduction to Machine learning, November 3, 1998
6. Kevin P. Murphy, Machine Learning A Probabilistic Perspective, May 2012
7. B.V. Kumar, S.V. Subrahmanya, Web Services An Introduction, 2008
8. James Snell, Doug Tidwell, Pavel Kulchenko, :Programming Web Services with SOAP, January 2002
9. Samisa Abeysinghe, RESTful PHP Web Services
10. Hal Daume III, A Course In Machine learing, Chapter 01 Decision Trees, September 2015
11. Pat Langley, Elements Of Machine Learning, 1996
12. S. German, E. Bienestock, and R. Doursat, Neural networks and the bias/variance dilemma. Neural computation 4, 1-58, 1992
13. G. James, Variance and Bias for General Loss Functions, Machine Learning 51, 115-135, 2003
14. Peter Dayan, Unsupervised Learning,
15. Intrator, N & Cooper, LN, Objective function formulation of the BCM theory of visual cortical plasticity statistical connections, stability conditions, Neural networks, 5, 3-17, 1992
16. Nowlan, Maximum likelihood competitive learning, 1990
17. Thomas M. Breuel, The OCRopus Open Source OCR System, 2008
18. http://www.danvk.org/2015/01/09/extracting-text-from-an-image-using-ocropus.html
19. http://www.danvk.org/2015/01/11/training-an-ocropus-ocr-model.html
20. Eric Newcomer, Understanding Web Services, 2002.
21. http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/
22. Performance Study – REST vs SOAP for Mobile Applications, http://www.ateam-oracle.com/performance-study-rest-vs-soap-for-mobile-applications/
23. Ray Smith, An Overview of the Tesseract OCR Engine
24. Optical character recognition, https://en.wikipedia.org/wiki/Optical_character_recognition
25. Introduction to decision trees, http://www.treeplan.com/chapters/introduction-to-decision-trees.pdf

26. Hal Daume III, A Course In Machine learing, Chapter 08 Neural networks, September 2015
27. A basic Introduction To Neural Networks, http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html
28. Artificial neural networks, https://en.wikipedia.org/wiki/Artificial_neural_network
29. Kevin Gurney, An introduction to Neural Networks, 1997
30. Vikas Wariyal, Shreyank Gupta, Vaibhav Krumar, Neural Network Learning Rules, 2013
31. Bogdan M. Wilamowski, Neural Networks Learning, 2010
32. Web Framework, https://en.wikipedia.org/wiki/Web_framework
33. Flask (web framework), https://en.wikipedia.org/wiki/Flask_(web_framework)