

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη διαδικτυακής εφαρμογής οπτικής αναγνώρισης
χαρακτήρων κειμένου**

Ιωάννα Γκούρα

**Εισηγητές:
Δρ. -Μηχ. Γεώργιος Διλιντάς, Καθηγητής
Στυλιανός Βουτσινάς, Πανεπιστημιακός Υπότροφος**

**ΑΘΗΝΑ
ΣΕΠΤΕΜΒΡΙΟΣ 2016**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη διαδικτυακής εφαρμογής οπτικής αναγνώρισης χαρακτήρων
κειμένου**

**Ιωάννα Γκούρα
Α.Μ. 39357**

Εισηγητής:

**Δρ. –Μηχ. Γεώργιος Διλιντάς, Καθηγητής
Στυλιανός Βουτσινάς, Πανεπιστημιακός Υπότροφος**

Εξεταστική Επιτροπή:

.....
.....
.....

Ημερομηνία εξέτασης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη Πτυχιακή Εργασία,
της Γκούρα Ιωάννας, με αριθμό μητρώου 39357 φοιτήτρια του Τμήματος
Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα
παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια
πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα
πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο
συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και
άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το
Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης
του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από
αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο
θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε.
πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την
ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο
άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της πτυχιακής μου εργασίας, η οποία υλοποιήθηκε στο Εργαστήριο Λειτουργικών Συστημάτων του Ανώτατου Τεχνολογικού Ιδρύματος Πειραιά Τ.Τ, θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέλαβαν στη διεκπεραίωση της.

Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου κ. Βουτσινάς Στυλιανός, για τον οποίο οφείλω να εκφράσω τις θερμές μου ευχαριστίες, για το ενδιαφέρον θέμα που προσέφερε σε συνεργασία με τον Δρ Γεώργιο Διλιντά, καθώς και την εμπιστοσύνη που μου έδειξαν δίνοντάς μου τη δυνατότητα να εκπονήσω την πτυχιακή μου εργασία στο συγκεκριμένο επιστημονικό τομέα. Τους ευχαριστώ επίσης, για τις πολύτιμες γνώσεις, τις συμβουλές, την υποστήριξη και την καθοδήγηση που μου παρείχαν καθόλη τη διάρκεια της εργασίας, όπως και καθ' όλη τη διάρκεια των σπουδών μου.

Ακόμα θα ήθελα να ευχαριστήσω τους δικούς μου ανθρώπους για την αμέριστη στήριξη, συμπαράσταση και κατανόηση που έδειξαν όλο αυτό το διάστημα.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη διαδικτυακής εφαρμογής αυτόματης αναγνώρισης χαρακτήρων κειμένου. Στα πρώτα κεφάλαια αναφερθήκαμε σε βασικές έννοιες και χρήσεις της επεξεργασίας ψηφιακής εικόνας και της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων. Άφου κάναμε μια ιστορική αναδρομή, έγινε περαιτέρω ανάλυση στην βιβλιοθήκη tesseract-ocr, στην οποία και βασίστηκε η παρούσα εργασία. Μετά την ολοκλήρωση της βιβλιογραφικής ανασκόπησης, παρουσιάζεται μια πλήρη περιγραφή του τρόπου χρήσης και λειτουργίας της εφαρμογής, όπως επίσης και μελλοντικές βελτιστοποιήσεις και επεκτάσεις αυτής.

ABSTRACT

This present thesis concerns the development of a web application for automatic recognition of text characters. The first chapters touched upon basic concepts and uses of digital image processing and Optical Character Recognition (OCR) technology. After we made a throwback, became further analysis to tesseract-ocr library, to which this work based. After completion of the literature review, presented a full description of how to use and operation the application, as well as future optimizations and extensions thereof.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Ψηφιακή επεξεργασία έντυπων εγγράφων
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: επεξεργασία εικόνας, αρχείο, συμπύεση, OCR, tesseract, responsive, web, PHP

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ.....	14
2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....	16
2.1 Επεξεργασία Εικόνας	16
2.1.1 Ψηφιακή Εικόνα.....	17
2.1.2 Χρωματικά Μοντέλα.....	18
2.1.3 Τύποι Εικόνων.....	23
2.1.4 Ανάλυση Εικόνων.....	25
2.1.5 Απωλεστική και Μη Απωλεστική Συμπύεση.....	26
2.2 Οπτική Αναγνώριση Χαρακτήρων.....	29
2.2.1 Ιστορική Αναδρομή.....	30
2.2.2 Πλεονεκτήματα OCR.....	34
2.2.3 Βιβλιοθήκη Tesseract.....	36
2.3 Τεχνολογία αυτόματης προσαρογής.....	40
3. ΠΕΙΡΑΜΑΤΙΚΗ ΥΛΟΠΟΙΗΣΗ.....	42
3.1 Περιγραφή Εφαρμογής.....	42
3.2 Διαγράμματα Ροής.....	52
3.3 Πειραματική Προσέγγιση.....	58
3.4 Μελέτη Συστήματος Ψηφιοποίησης.....	72
4. ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	74
4.1 Σχόλια – Συμπεράσματα.....	74
4.2 Μελλοντική Επέκταση.....	74
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	82

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2.1: Μετατρέποντας την πραγματικότητα σε ένα ψηφιακό αρχείο.....	16
Εικόνα 2.2: Πινακοποιημένη εικόνα.....	17
Εικόνα 2.3: Μοντέλο RGB.....	19
Εικόνα 2.4: Μοντέλο CMY(K).....	20
Εικόνα 2.5: Μοντέλο HSI.....	21
Εικόνα 2.6: Μετατροπή RGB σε YcrCb.....	22
Εικόνα 2.7: Μοντέλο HSV.....	22
Εικόνα 2.8: Ενδεικτικοί τύποι αρχείων εικόνας.....	23
Εικόνα 2.9: Παράδειγμα έλλειψης χρωματικού υπόβαθρου.....	24
Εικόνα 2.10: Απεικόνιση σημείων ανά ίντσα.....	25
Εικόνα 2.11: Αναπαράσταση εικονοστοιχείων μονοχρωματικής εικόνας.....	26
Εικόνα 2.12: Αναπαράσταση εικονοστοιχείων έγχρωμης εικόνας.....	26
Εικόνα 2.13: Αναπαράσταση ενδεικτικών παραδειγμάτων συμπίεσης.....	27
Εικόνα 2.14: Αναπαράσταση απωλεστικής και μη συμπίεσης.....	28
Εικόνα 2.15: Οπτική αναγνώριση χαρακτήρων.....	29
Εικόνα 2.16: Αριστερά ο Goldberg και δεξιά ο Edmund Fournier d' Albe.....	30
Εικόνα 2.17: Gustav Tauschek.....	31
Εικόνα 2.18: Ραδιοφωνική Εταιρεία Αμερικής.....	31
Εικόνα 2.19: David H.Shepard.....	32
Εικόνα 2.20: Jacob Rabinow.....	33
Εικόνα 2.21: Εισαγωγή εγγράφων σε ηλεκτρονικό σύστημα.....	34
Εικόνα 2.22: Από το φυσικό αρχείο στο ηλεκτρονικό.....	35
Εικόνα 2.23: Δημιουργία αντιγράφων ασφαλείας.....	36
Εικόνα 2.24: Αρχιτεκτονική Tesseract.....	37
Εικόνα 2.25: Leptonica.....	38
Εικόνα 2.26: Μηχανή οπτικής αναγνώρισης αρχείων.....	39
Εικόνα 2.27: Προβολή ιστοσελίδας με χρήση της τεχνολογίας του Responsive..	40
Εικόνα 3.1: Γλώσσες προγραμματισμού HTML5, CSS3, JS, PHP.....	42
Εικόνα 3.2: Οι γνωστότεροι φυλλομετρητές.....	43
Εικόνα 3.3: Προβολή της εφαρμογής σε διάφορες αναλύσεις.....	44
Εικόνα 3.4: Αρχική σελίδα εφαρμογής.....	46
Εικόνα 3.5: Σελίδα σύνδεσης χρήστη.....	47

Εικόνα 3.6: Σελίδα εγγραφής χρήστη.....	47
Εικόνα 3.7: Μήνυμα επιβεβαίωσης της διαδικασίας.....	48
Εικόνα 3.8: Μήνυμα ηλεκτρονικού ταχυδρομείου με σύνδεσμο επιβεβαίωσης...	48
Εικόνα 3.9: Σελίδα σύνδεσης χρήστη.....	48
Εικόνα 3.10: Σελίδα εφαρμογής OCR.....	49
Εικόνα 3.11: Αριστερά ενδεικτικό κείμενο αγγλικού κειμένου, δεξιά ελληνικού....	49
Εικόνα 3.12: Περιοχή επιλογής γλώσσας.....	50
Εικόνα 3.13: Προεπισκόπηση περιβάλλοντος επιλογής αρχείου.....	50
Εικόνα 3.14: Αποτέλεσμα OCR σε μορφή προεπισκόπησης και επιλογή αποθήκευσης του .txt αρχείου.....	51
Εικόνα 3.15: Αποτέλεσμα OCR σε .zip αρχείο και επιλογή αποθήκευσης.....	51
Εικόνα 3.16.1: Διάγραμμα ροής του κώδικα ocr μέρος 1.....	52
Εικόνα 3.16.2: Διάγραμμα ροής του κώδικα ocr μέρος 2.....	53
Εικόνα 3.17: Διάγραμμα ροής του κώδικα login.....	54
Εικόνα 3.18: Διάγραμμα ροής του κώδικα register.....	55
Εικόνα 3.19: Διάγραμμα ροής του κώδικα logout.....	56
Εικόνα 3.20: Διάγραμμα ροής του κώδικα password-reset.....	57
Εικόνα 3.21: Διάγραμμα ροής του κώδικα δημιουργίας πίνακα χρηστών.....	58
Εικόνα 3.22: Παράδειγμα 1.....	59
Εικόνα 3.23: Παράδειγμα 2.....	60
Εικόνα 3.24: Παράδειγμα 3.....	61
Εικόνα 3.25: Παράδειγμα 4.....	62
Εικόνα 3.26: Παράδειγμα 5.....	63
Εικόνα 3.27: Παράδειγμα 6.....	64
Εικόνα 3.28: Παράδειγμα 7.....	65
Εικόνα 3.29: Παράδειγμα 8.....	66
Εικόνα 3.30: Παράδειγμα 9.....	67
Εικόνα 3.31: Παράδειγμα 10.....	68
Εικόνα 3.32: Παράδειγμα 11.....	69
Εικόνα 3.33: Παράδειγμα 12.....	70
Εικόνα 3.34: Παράδειγμα 13.....	70
Εικόνα 3.35: Παράδειγμα 14.....	71
Εικόνα 3.36: Γράφημα Αποτελεσμάτων.....	72
Εικόνα 3.37: Αναπαράσταση Συστήματος Ψηφιοποίησης.....	73

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

DIP Digital Image Processing

UPDIG Universal Photographic Digital Imaging Guidelines

RGB Red Green Blue

CMYK Cyan Magenta Yellow

HIS Hue Saturation Intensity

HSV Hue Saturation Value

PNG Portable Network Graphics

PC Personal Computer

PPI Points Per Inch

DPI Dots Per Inch

RLE Run Length Encoding

LZW Lempel Zip Welch

OCR Optical Character Recognition

RCA Radio Corporation of America

IMR Intelligent Machines Research

GUI Graphical User Interface

UML Unified Modeling Language

PHP Hypertext Preprocessor

HTML Hyper Text Markup Language

CSS Cascading Style Sheets

JS Java Script

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Ο απώτερος σκοπός της παρούσας πτυχιακής εργασίας είναι η κατανόηση της τεχνολογίας OCR και η δημιουργία διαδικτυακής εφαρμογής που αφορά την οπτική αναγνώριση χαρακτήρων κειμένου από έντυπα έγγραφα, όπως επίσης η συνειδητοποίηση των λύσεων που μπορεί να προσφέρει ο προγραμματισμός αν τον χρησιμοποιήσουμε σωστά.

Η συνεχής ανάγκη εύρεσης και επεξεργασίας πληροφοριών, καθιστά αναγκαία μια εφαρμογή μετατροπής έντυπου εγγράφου σε επεξεργάσιμο ψηφιακό κείμενο. Η δυσκολία παρουσιάζεται στην εκπαίδευση απαιτούμενων βιβλιοθηκών και η παραμετροποίησή τους, ώστε να έχουμε το επιθυμητό αποτέλεσμα.

Έχει αναπτυχθεί πλήθος παρόμοιων εφαρμογών είτε σε μορφή desktop είτε διαδικτυακής μορφής, πολλές από αυτές παρουσιάζουν σφάλματα στην ελληνική γλώσσα, καθώς και είναι η δυσκολότερη, λόγω του ότι δεν ακολουθεί το λατινικό αλφάβητο όπως οι περισσότερες γλώσσες και οι χαρακτήρες εμπεριέχουν τόνους. Άλλες πάλι λειτουργούν άψογα, δίνοντας αποτελέσματα υψηλών ποσοστών επιτυχίας με πολύ μικρά έως και μηδαμινά σφάλματα, όμως το κόστος τους είναι αρκετά μεγάλο με αποτέλεσμα να αποτρέψει πολλούς χρήστες.

Η παρούσα εφαρμογή είναι Open Source και έχει χτιστεί σε ένα φιλικό περιβάλλον ως προς το χρήστη καθώς προσαρμόζεται αυτόματα σε όλες τις συσκευές χωρίς να αλλοιώνεται η προβαλλόμενη πληροφορία.

Οι θεωρητικές έννοιες για την κατανόηση λειτουργίας της εφαρμογής, αναπτύσσονται με στόχο την πρόκληση ενδιαφέροντος του αναγνώστη για το γνωστικό αντικείμενο της εργασίας, και την εξοικείωσή του με την τεχνολογία OCR.

Η παρούσα εργασία, λόγω του ότι είναι ανοικτού κώδικα, δίνει τη δυνατότητα στους ενδιαφερόμενους να διευρύνουν τις γνώσεις τους και με την κατάλληλη ενασχόληση να προβούν σε πιθανές βελτιστοποιήσεις, δημιουργώντας παρόμοιες εφαρμογές για προσωπική τους χρήση ή ακόμα και εμπορικού σκοπού.

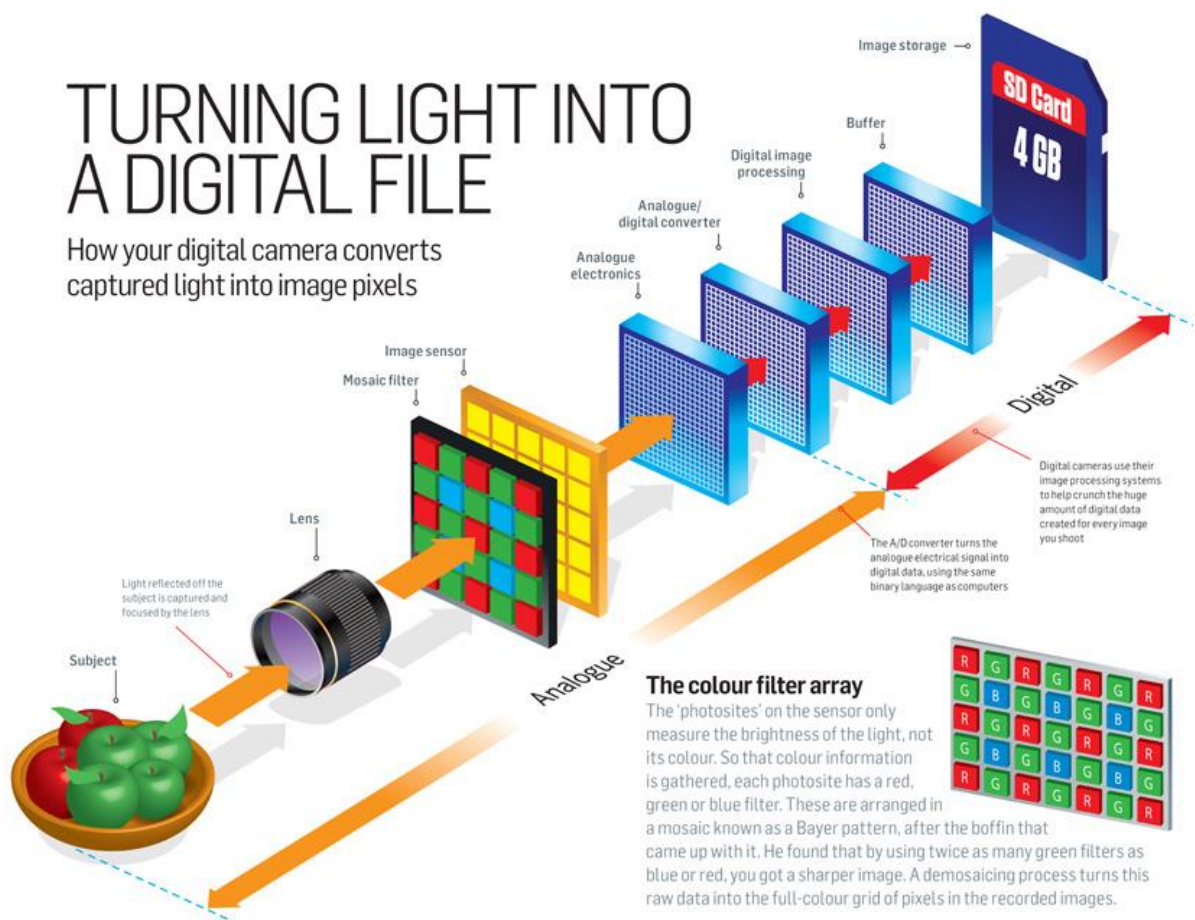
Ακολουθούν κεφάλαια στα οποία αναπτύσσεται η λογική που οδήγησε στην υλοποίηση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 2

ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

2.1 Επεξεργασία Εικόνας

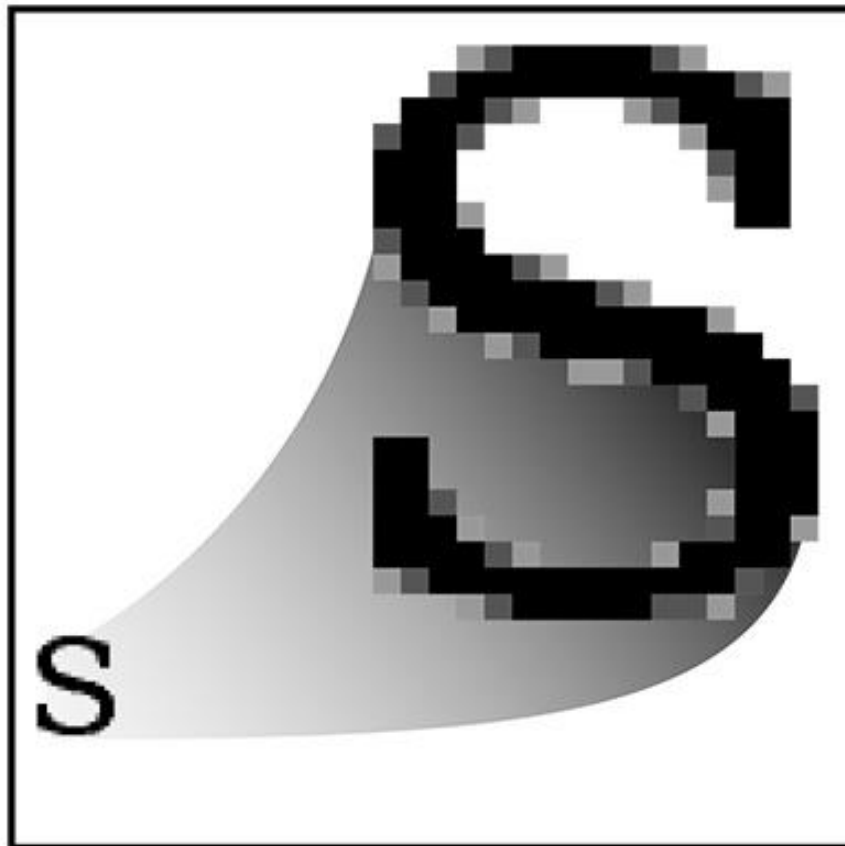
Η Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) ή Digital Image Processing (DIP) ασχολείται με τον τρόπο χρήσης μιας εικόνας που έχει ληφθεί, δηλαδή με τεχνικές ανάκτησης, ψηφιοποίησης, ανάλυσης, αποθήκευσης, συμπίεσης και βελτίωσης μιας εικόνας. [1]



Εικόνα 2.1: Μετατρέποντας την πραγματικότητα σε ένα ψηφιακό αρχείο

2.1.1 Ψηφιακή Εικόνα

Ψηφιακή εικόνα είναι μια αριθμητική αναπαράσταση (συνήθως δυαδική) μιας δισδιάστατης εικόνας. Ανάλογα με το αν η ανάλυση της εικόνας είναι σταθερή, μπορεί να είναι διανυσματική ή πινακοποιημένη. Από μόνος του, ο όρος «ψηφιακή εικόνα» αναφέρεται συνήθως σε πινακοποιημένες εικόνες (raster images) ή εικόνες bitmap.[1]



Raster

.jpeg .gif .png

Εικόνα 2.2: Πινακοποιημένη εικόνα

Οι πινακοποιημένες εικόνες έχουν ένα πεπερασμένο σύνολο από ψηφιακές τιμές, που ονομάζονται στοιχεία εικόνας ή εικονοστοιχεία. Η ψηφιακή εικόνα περιέχει ένα σταθερό αριθμό γραμμών και στηλών από εικονοστοιχεία (pixels).

Pixels είναι τα μικρότερα επιμέρους στοιχεία μιας εικόνας, κρατώντας πληροφορίες που αντιπροσωπεύουν την φωτεινότητα ενός δεδομένου χρώματος σε οποιοδήποτε συγκεκριμένο σημείο.[2]

Τυπικά, τα εικονοστοιχεία αποθηκεύονται στη μνήμη του υπολογιστή ως μια εικόνα raster ή χάρτη, σε μια δισδιάστατη διάταξη μικρών ακεραίων. Αυτές οι τιμές συχνά μεταδίδονται ή αποθηκεύονται σε συμπιεσμένη μορφή.[1]

Οι Raster εικόνες μπορούν να δημιουργηθούν από μια ποικιλία συσκευών εισόδου και τεχνικών, όπως ψηφιακές φωτογραφικές μηχανές, σαρωτές, μηχανήματα μέτρησης συντεταγμένων, εναέρια ραντάρ, και πολλά άλλα. Μπορούν επίσης να συντεθούν από αυθαίρετα δεδομένα, όπως μαθηματικές συναρτήσεις ή τρισδιάστατα γεωμετρικά μοντέλα τα οποία αποτελούν μια σημαντική υπο-περιοχή γραφικών των ηλεκτρονικών υπολογιστών. Ο τομέας της ψηφιακής επεξεργασίας εικόνας είναι η μελέτη των αλγορίθμων για τον μετασχηματισμό τους.[2][1]

Οι περισσότεροι χρήστες έρχονται σε επαφή με εικόνες raster μέσα από ψηφιακές φωτογραφικές μηχανές, οι οποίες χρησιμοποιούν αρκετές μορφές αρχείων εικόνας. Ορισμένες ψηφιακές φωτογραφικές μηχανές δίνουν πρόσβαση σε σχεδόν όλα τα δεδομένα που λαμβάνονται από την κάμερα. Οι παγκόσμιες κατευθυντήριες γραμμές φωτογραφικών εικόνων (UPDIG) προτείνουν να χρησιμοποιούνται οι αρχικές μορφές, όταν είναι δυνατόν, δεδομένου ότι τα μη επεξεργασμένα αρχεία παράγουν την καλύτερη ποιότητα εικόνων. [2]

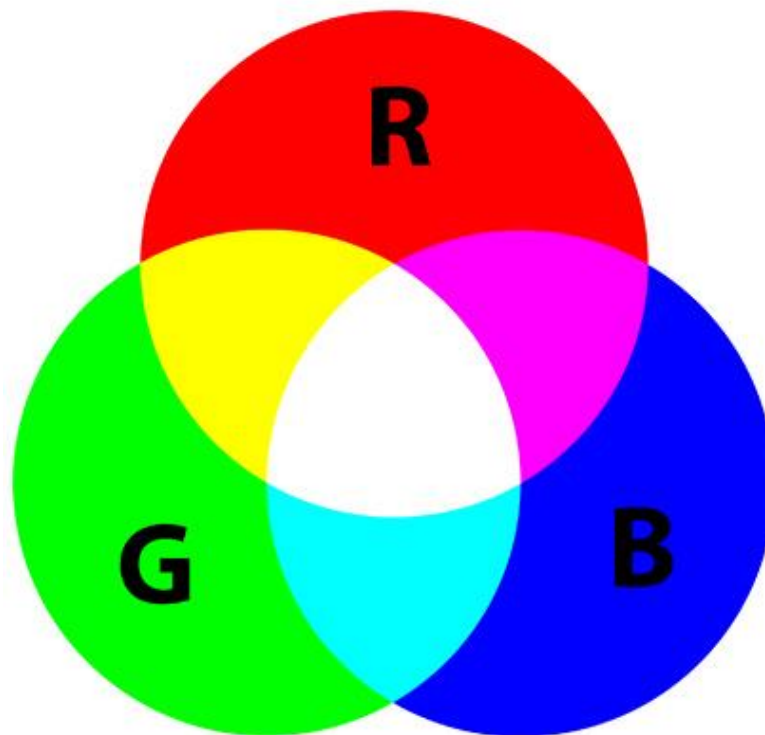
Αυτές οι μορφές αρχείων επιτρέπουν τον φωτογράφο και το μέσο επεξεργασίας το μεγαλύτερο επίπεδο ελέγχου και ακρίβειας για έξοδο. Η χρήση τους παρεμποδίζεται από την επικράτηση των αποκλειστικών πληροφοριών (εμπορικών μυστικών) για ορισμένους κατασκευαστές φωτογραφικών μηχανών.[2]

2.1.2 Χρωματικά Μοντέλα

Η λεπτομερειακή απόδοση των χρωμάτων με μαθηματική μορφή, είναι απαραίτητη για την επεξεργασία τους από τα ψηφιακά μέσα. Η ανάγκη αυτή συνέβαλε στην ανάπτυξη των χρωματικών μοντέλων. Καθένα από τα εικονοστοιχεία, στα οποία αναλύεται η εικόνα, είναι χρωματικά συναφή. Υπάρχουν διάφορα χρωματικά συστήματα, όπως το RGB, το οποίο χρησιμοποιείται στα

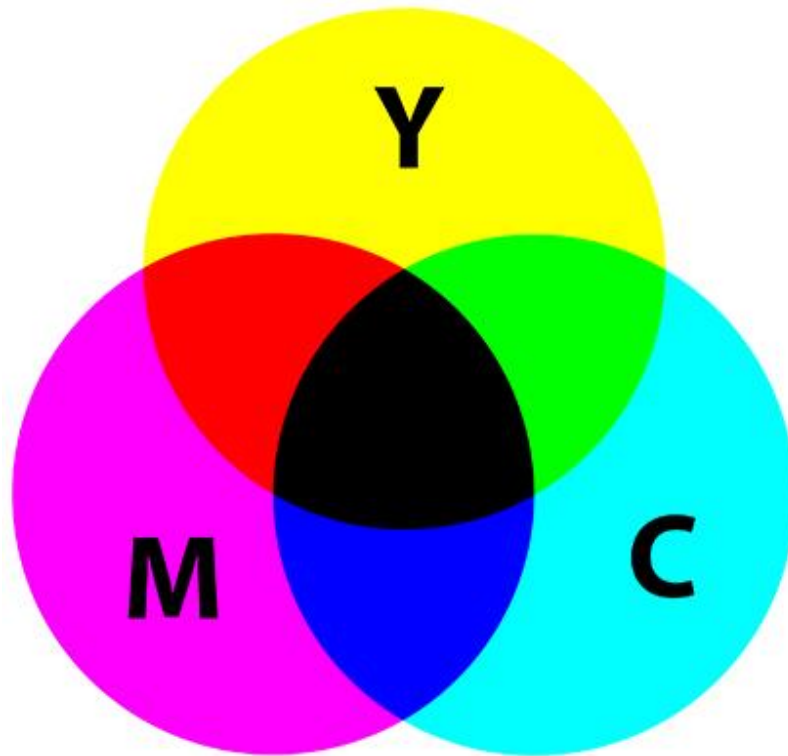
τερματικά, το CMYK το οποίο χρησιμοποιείται στην εκτύπωση έγχρωμων εικόνων, τα HSI, YCbCr, YIQ και YUV. [1]

Το μοντέλο RGB απλοποιεί αρκετά τη σχεδίαση γραφικών συστημάτων, αλλά δημιουργεί δυσκολίες στην ανάπτυξη αλγορίθμων λόγω της συσχέτισης των χρωματικών συνιστώσων του. [9]



Εικόνα 2.3: Μοντέλο RGB

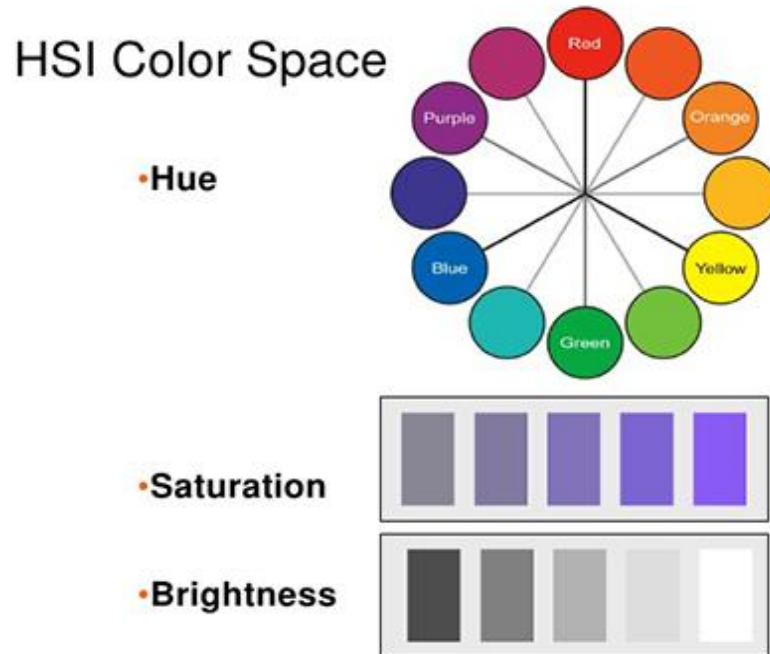
Το σύστημα CMY ή CMYK ορίζεται από τα χρώματα κυανό (Cyan), ματζέντα (Magenta), κίτρινο (Yellow), τα οποία είναι συμπληρωματικά των χρωμάτων RGB. Εάν προστεθεί και το μαύρο προκύπτει το CMYK, το οποίο παράγει στις εκτυπώσεις καλύτερο μαύρο από την ανάμειξη των τριών προαναφερθέντων χρωμάτων. [9]



Εικόνα 2.4: Μοντέλο CMY(K)

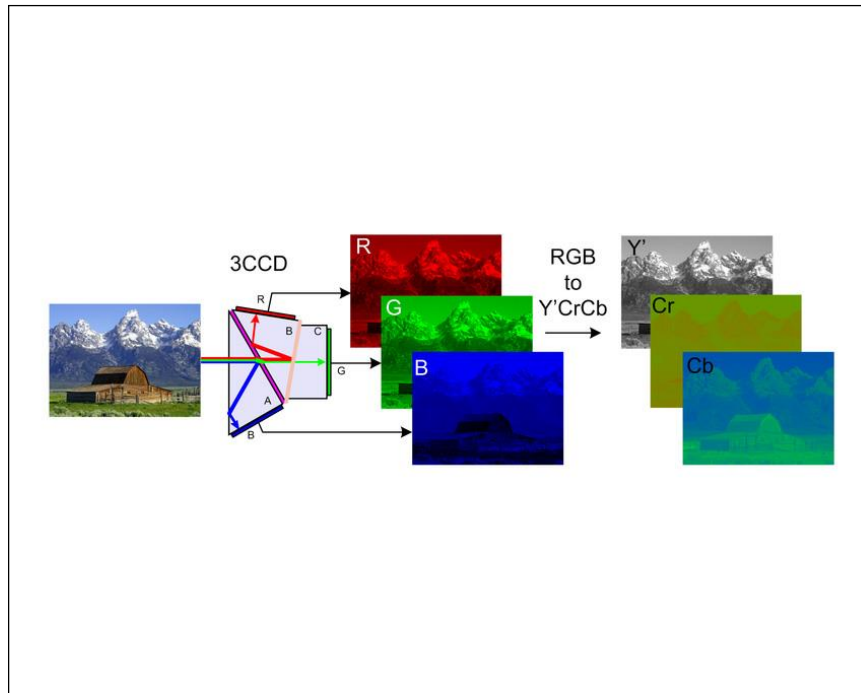
Η απόχρωση (Hue), η καθαρότητα (Saturation) και η φωτεινότητα (Intensity), τα τρία χαρακτηριστικά του χρώματος, αποτελούν το μοντέλο HSI, στο οποίο δεν χρειάζεται την αναλογία RGB, αλλά το χρώμα καθορίζεται από τη μεταβολή του.

[10]



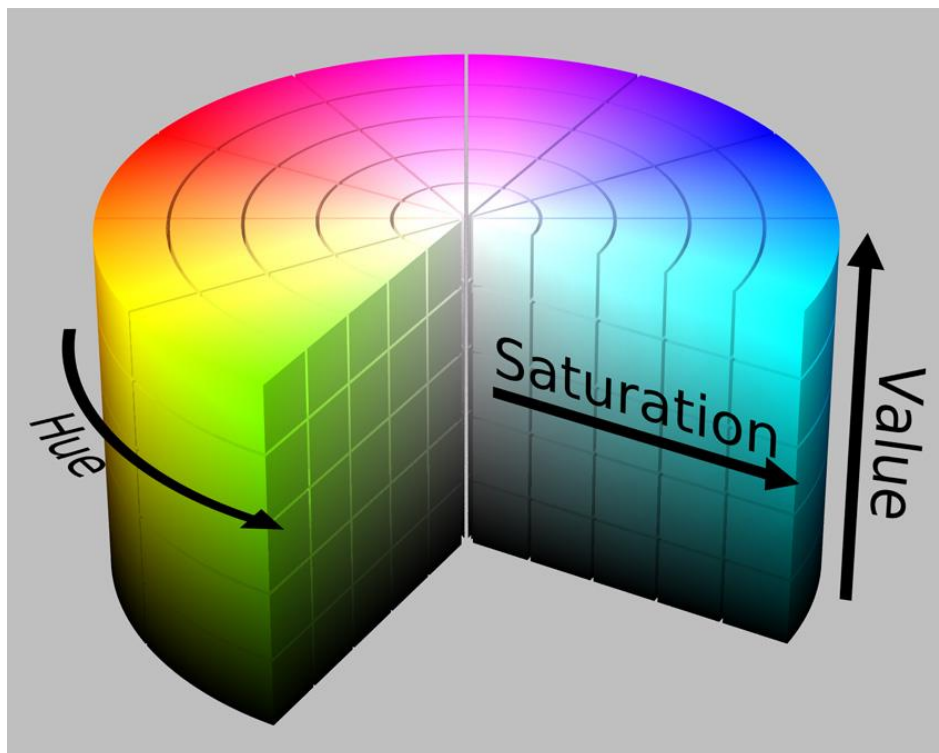
Εικόνα 2.5: Μοντέλο HSI

Στο σύστημα YCbCr παρατηρούμε το διαχωρισμό της φωτεινότητας από την πληροφορία χρώματος (μπλε - κόκκινο). Η φωτεινότητα (Y), η απόχρωση και η χρωματική καθαρότητα (I, Q), είναι τα βασικά χαρακτηριστικά του μοντέλου YIQ, το οποίο αποτελεί επανακωδικοποίηση του RGB και χρησιμοποιείται στην έγχρωμη τηλεοπτική μετάδοση. [2]



Εικόνα 2.6: Μετατροπή RGB σε YCrCb

Η ανάλυση των χρωμάτων στις εξής τρεις παραμέτρους, χροιά (Hue), κορεσμός (Saturation) και αξία (Value), αποτελούν το σύστημα HSV.[9]



Εικόνα 2.7: Μοντέλο HSV

Παρατηρείται ότι η μετατροπή από ένα χρωματικό μοντέλο σε άλλο είναι πολλές φορές αναγκαία, καθώς οι πιθανοί τρόποι επεξεργασίας μιας εικόνας ποικίλλουν, ανάλογα με τις ανάγκες μας.

2.1.3 Τύποι Εικόνων [11]

Κατά την αναζήτηση και την λήψη ηλεκτρονικών αρχείων, παρατηρούμε αναμφίβολα πολλούς διαφορετικούς τύπους μορφών, τους οποίους μπορούμε να προσδιορίσουμε εξετάζοντας την επέκταση του αρχείου, όπου εκφράζεται ως μια τελεία (.) που ακολουθείται από δύο έως τέσσερις χαρακτήρες (.xxx). Ο λόγος που χρειάζεται να γνωρίζουμε τον τύπο ενός αρχείου είναι για να αναγνωρίσουμε κατά πόσο είναι εφικτό να αναπαραχθεί με το υπάρχον λογισμικό του υπολογιστή μας και για τον τρόπο που μπορεί να χρησιμοποιηθεί κατάλληλα, ανάλογα με την ποιότητα της πληροφορίας, η οποία αλλάζει ανάλογα με τη μορφή επέκτασης του αρχείου.

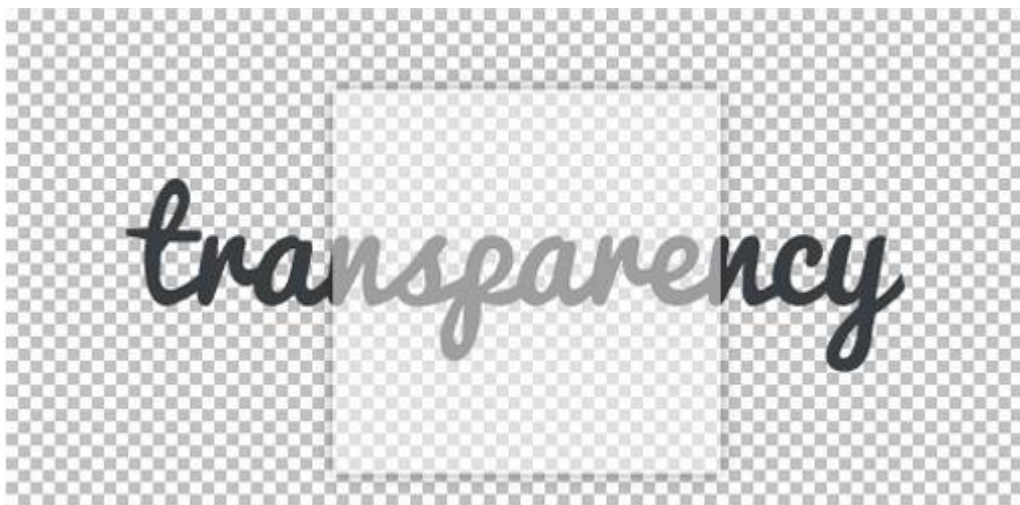


Εικόνα 2.8: Ενδεικτικοί τύποι αρχείων εικόνας

Πολλές ιστοσελίδες καθιστούν τα αρχεία διαθέσιμα για λειτουργικά συστήματα Windows, Linux και Macintosh. Ο φυλλομετρητής (Web browser) έχει τη δυνατότητα να εντοπίσει και να αναπαραγάγει κάποια, αλλά όχι όλα. Είναι χρήσιμο,

λοιπόν να γνωρίζουμε για τις πιο κοινές μορφές αρχείων στο διαδίκτυο και τον τρόπο χρήσης τους.

Οι πιο συνηθισμένες μορφές αρχείων γραφικών στο διαδίκτυο είναι εκείνες με τις επεκτάσεις .jpg .png και .gif. Το .jpg είναι συντόμευση για το JPEG (Joint Photographic Experts Group), το οποίο είναι ένα δημοφιλές πρότυπο συμπίεσης για εικόνες. Ένα από τα δύο format που χρησιμοποιείται στο internet γιατί επιτυγχάνει πολύ μεγάλο βαθμό συμπίεσης. Ο βαθμός συμπίεσης μπορεί να επιλεγεί από το χρήστη σε ένα μεγάλο εύρος. Βάθος χρώματος 24 bit ή 32 bit για RGB ή CMYK μοντέλο αντίστοιχα, και 8 bit ασπρόμαυρο μοντέλο. Η επέκταση .Gif σημαίνει Graphics Interchange Format, ένα πρότυπο που αναπτύχθηκε από την CompuServe στα τέλη του 1980. Αποτελεί το δεύτερο format που χρησιμοποιείται στο internet. Κατάλληλο για γραφικά, όχι για φωτογραφίες. Ειδικά όταν δεν περιέχει πολλά χρώματα παράγει εξαιρετικά μικρού μεγέθους αρχεία. Βάθος χρώματος ως 1-8 bit έγχρωμα ή ασπρόμαυρα RGB. Το .png προέρχεται από τα αρχικά των λέξεων Portable Network Graphics και είναι ένας τύπος εικόνας που υποστηρίζει τη συμπίεση των δεδομένων (για να έχει μικρό μέγεθος σε kb) με ελάχιστη απώλεια ποιότητας. Περιλαμβάνουν παλέτα χρωμάτων 24-bit που σημαίνει ότι οι εικόνες θα έχουν τεράστια γκάμα χρωμάτων με αποτέλεσμα να μην έχουμε μεγάλη αλλοίωση. Αντίστοιχα υπάρχει και παλέτα χρωμάτων 8-bit για εικόνες σε μόνο 256 χρώματα σε αυτόν τον τύπο εικόνας. Επίσης, υποστηρίζει πολύ καλά τη λειτουργία της διαφάνειας (transparency). Το transparency είναι μια ιδιότητα που κάνει το γραφικό να φαίνεται χωρίς βάθος (background).

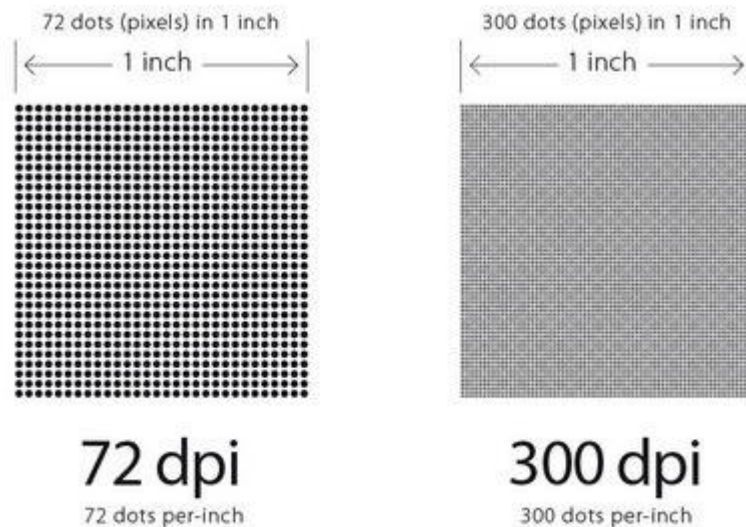


Εικόνα 2.9: Παράδειγμα έλλειψης χρωματικού υπόβαθρου

Όλες αυτές οι μορφές γραφικών είναι ανεξάρτητες από την πλατφόρμα (λειτουργικό) του ηλεκτρονικού υπολογιστή (PC), που σημαίνει ότι μπορούν να αναπαραχθούν εφόσον υπάρχει εγκατεστημένο ένα πρόγραμμα προβολής για αυτούς τους τύπους των αρχείων.

2.1.4 Ανάλυση Εικόνας

Η ανάλυση μίας ψηφιακής εικόνας μπορεί να εκφραστεί με τον λόγο των εικονοστοιχείων ανά ίντσα και συμβολίζεται ως ppi (pixel per inch). Στην περίπτωση των εκτυπωτών συναντάμε τα σημεία ανά ίντσα (dots per inch (dpi)), μία διαφορετική έννοια, καθώς η μονάδα dpi εκφράζει το μέτρο της ευκρίνειας που μπορεί να επιτύχει ο εκτυπωτής. Όσο περισσότερα εικονοστοιχεία χρησιμοποιηθούν για την απεικόνιση ενός αντικειμένου τόσο το οπτικό αποτέλεσμα θα πλησιάζει το πραγματικό.[2]



Εικόνα 2.10: Απεικόνιση σημείων ανά ίντσα

Η έκφραση της ανάλυσης σε εικονοστοιχεία γίνεται είτε ως ενιαίος αριθμός (π.χ 3 megapixels), είτε ως ζεύγος αριθμών (π.χ $1920 \times 1080 = 2.076.600 = 2,1$ megapixels).

Ο καθορισμός του μεγέθους ενός εικονοστοιχείου οθόνης καθορίζεται από την ανάλυση της οθόνης και το μέτρο της διαγωνίου της. Ο λόγος διάστασης ορίζει κατά πόσο το εικονοστοιχείο είναι τετράγωνο ή ορθογώνιο και η διαφορά αυτού

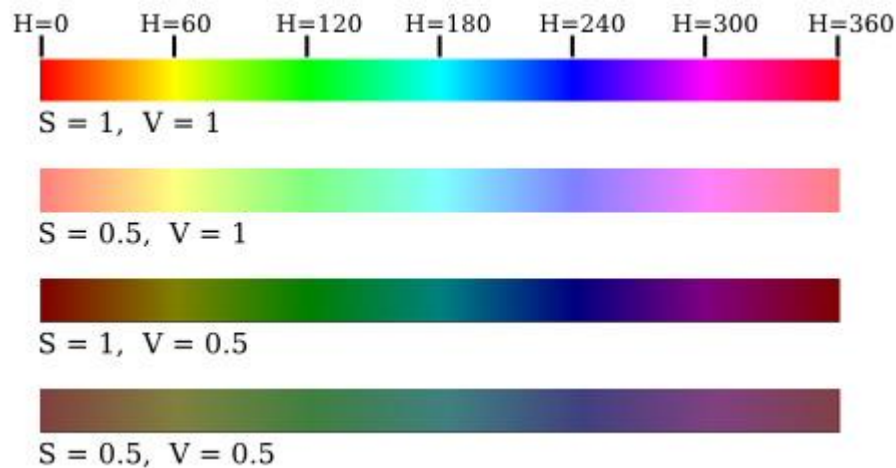
αποτελεί τη βασική αιτία της εμφάνισης παραμορφωμένων εικόνων στις οθόνες με σταθερή ανάλυση.

Σε μια μονοχρωματική εικόνα η μέγιστη τιμή φωτεινότητας ενός εικονοστοιχείου αντιπροσωπεύει το λευκό και η ελάχιστη το μαύρο. Δηλαδή σε μια εικόνα οκτώ δυαδικών ψηφίων (bits), η μέγιστη τιμή που μπορεί να αναπαρασταθεί από τα εικονοστοιχεία είναι $2^8=256$, έτσι το 255 είναι η τιμή που χρησιμοποιείται για το λευκό και η τιμή 0 για το μαύρο. [1]



Εικόνα 2.11: Αναπαράσταση εικονοστοιχείων μονοχρωματικής εικόνας

Στις έγχρωμες εικόνες τα εικονοστοιχεία περιγράφονται από την απόχρωση, τον κορεσμό και τη φωτεινότητα τους (HSV) και αντιπροσωπεύονται από τις τιμές του μοντέλου RGB.[1]



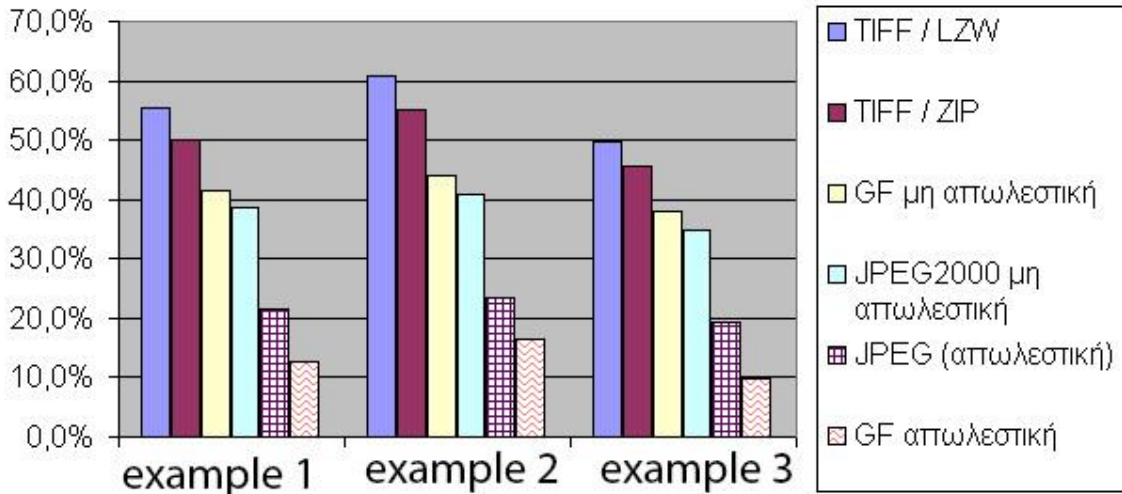
Εικόνα 2.12: Αναπαράσταση εικονοστοιχείων έγχρωμης εικόνας

Ο λόγος φωτεινότητας μεταξύ λευκού και μαύρου που προβάλεται σε μια οθόνη, αποτελεί τον λόγο αντίθεσης. Η αντίθεση αν και θα μπορούσε να θεωρηθεί άπειρη, στην πράξη είναι επιρρεπής στην φωτεινότητα του λευκού και την αχνότητα του μαύρου.

2.1.5 Απωλεστική και Μη Απωλεστική Συμπίεση [3]

Η μετατροπή ενός ψηφιακού αρχείου σε αρχείο με μικρότερο αριθμό τμημάτων (bit) ονομάζεται συμπίεση δεδομένων (data compression). Οι μέθοδοι συμπίεσης

είναι αρκετοί και χωρίζονται σε δύο κατηγορίες: τις μη απωλεστικές και τις απωλεστικές.



Εικόνα 2.13: Αναπαράσταση ενδεικτικών παραδειγμάτων συμπίεσης

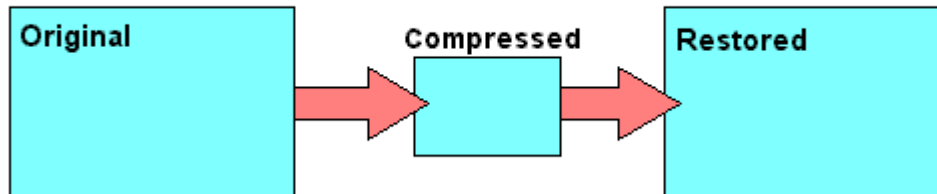
Στο περιεχόμενο πολλών αρχείων παρατηρείται επαναπάληψη μεγάλων τμημάτων ή δεδομένα που η απώλειά τους δεν δημιουργεί σοβαρή μεταβολή του περιεχόμενου.

Για να διατηρήσουμε την ακεραιότητα των δεδομένων, χρησιμοποιούμε την μη απωλεστική συμπίεση (lossless compression), με την οποία τα αρχικά δεδομένα και τα δεδομένα μετά τη συμπίεση και την αποσυμπίεση παραμένουν τα ίδια, και αυτό γιατί ο αλγόριθμος συμπίεσης και ο αλγόριθμος αποσυμπίεσης είναι ακριβώς αντίστροφοι. Κατά τη διαδικασία συμπίεσης δε χάνεται μέρος των δεδομένων. Τα τμήματα του περιεχομένου κωδικοποιούνται κατά τη συμπίεση και αποκωδικοποιούνται κατά την αποσυμπίεση. Η μέθοδος αυτή ενδείκνυται όταν δεν πρέπει να χαθεί ούτε ένα μπιτ δεδομένων όπως στην περίπτωση ενός αρχείου κειμένου ή ενός προγράμματος.

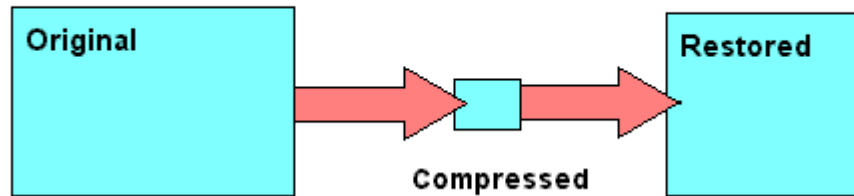
Στην περίπτωση όμως που πρόκειται για αρχεία εικόνων, βίντεο ή ήχου, η απώλεια δεδομένων είναι αποδεκτή, λόγω του ότι δεν μπορούμε να αντιληφθούμε μικρές αλλοιώσεις στο περιεχόμενό τους. Κατάλληλες για αυτές τις περιπτώσεις είναι οι απωλεστικές μέθοδοι συμπίεσης (lossy data compression), και έχουν βασικά τους πλεονεκτήματα είναι η ταχύτητα, ο χώρος και το κόστος, όταν πρόκειται για μεγάλο πλήθος αρχείων.

Αν και για την πλειοψηφία των περιπτώσεων, η συνήθης αποθήκευση των φωτογραφιών σε format JPEG (στη ρύθμιση μέγιστης ποιότητας) επαρκεί για τις ανάγκες μας, υπάρχουν κάποιες φορές που πρέπει να αποθηκεύσουμε κάποιες εικόνες χωρίς την παραμικρή απώλεια πληροφορίας.

LOSSLESS



LOSSY



Εικόνα 2.14: Αναπαράσταση απωλεστικής και μη συμπίεσης

Κατά τη λήψη φωτογραφιών από μηχανές δεν εφαρμόζεται απωλεστική συμπίεση και αποθηκεύονται στη μέγιστη ποιότητα για να υπάρχει η δυνατότητα περαιτέρω επεξεργασίας. Η μη χρήση μεθόδων συμπίεσης, είναι ασύμφορη από πλευράς κόστους, για αυτό το λόγο εξετάζουμε μορφές (format) αποθήκευσης μη απολεστικών αλγόριθμων συμπίεσης. Συχνότερη χρήση γίνεται στους εξής αλγόριθμους:

- RLE (Run Length Encoding): Χρησιμοποιείται από το Adobe Photoshop και από αρκετά διαδεδομένα format αρχείων των Windows.
- LZW (Lempel-Zif-Welch): Μέθοδος που χρησιμοποιείται στα format TIFF, PDF, GIF. Είναι πιο αποτελεσματική για τη συμπίεση εικόνων με μεγάλες περιοχές του ίδιου χρώματος.
- ZIP: Μέθοδος που χρησιμοποιείται στα format PDF and TIFF. Είναι πιο αποτελεσματική για τη συμπίεση εικόνων με μεγάλες περιοχές του ίδιου χρώματος.

Για μια τελική αποτίμηση των μεθόδων που εξετάστηκαν, είναι φανερό πως οι μέθοδοι συμπίεσης που έχουν βασιστεί σε νεότερες τεχνολογίες δίνουν καλύτερα αποτελέσματα. Για μη απωλεστική συμπίεση, όταν το σημαντικότερο κριτήριο

είναι η εξοικονόμηση χώρου, η προτιμότερη λύση, με μικρή διαφορά, είναι η συμπίεση JPEG2000. Εάν κρίνεται βαρύνουσας σημασίας η εξαιρετική ποιότητα που δίνει η συμπίεση με fractals κατά τη μεγέθυνση και τη σμίκρυνση εικόνας, ίσως πρέπει να προτιμηθεί. Θα πρέπει όμως να συνυπολογιστεί το γεγονός πως είναι λύσεις με επιπρόσθετο κόστος, μεγαλύτερες απαιτήσεις σε υπολογιστική ισχύ και εξαρτώμενες από συγκεκριμένο λογισμικό.

2.2 Οπτική Αναγνώριση Χαρακτήρων

Η ηλεκτρονική μετατροπή σαρωμένων εικόνων χειρόγραφου ή τυποποιημένου κειμένου σε κωδικοποιημένη μορφή χαρακτήρων κειμένου ηλεκτρονικού υπολογιστή, ονομάζεται Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition - OCR) και αποτελεί τον πλέον ευρέως χρησιμοποιούμενο τρόπο εξαγωγής δεδομένων από οποιοδήποτε εγγράφο και πεδίο έρευνας για τους τομείς της αναγνώρισης προτύπων, της τεχνητής νοημοσύνης και της κατανόησης των εικόνων από τον ηλεκτρονικό υπολογιστή.



Εικόνα 2.15: Οπτική αναγνώριση χαρακτήρων

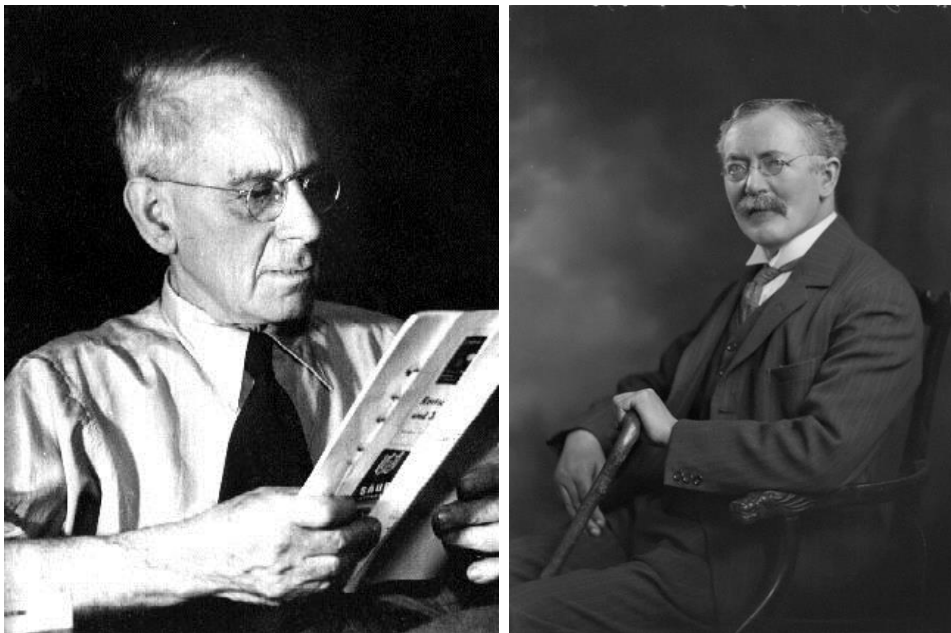
Η μέθοδος ψηφιοποίησης (digitizing) έντυπων κειμένων καθιστά δυνατή την ηλεκτρονική προσπέλαση, την αποθήκευση σε μικρότερο χώρο, την άμεση πρόσβαση από τον παγκόσμιο ιστό και τη χρήση τους σε περαιτέρω διαδικασίες όπως είναι η αυτόματη μετάφραση (machine translation), η εξόρυξη δεδομένων (data mining) και η μετατροπή κειμένου σε ομιλία από τον ηλεκτρονικό υπολογιστή (text-to speech). [8]

2.2.1 Ιστορική Αναδρομή

Οι προγενέστερες υλοποιήσεις εφαρμογών, έκαναν αναζήτηση μεμονομένων χαρακτήρων μέσα στα κείμενα, με βάση απομνημονευμένες φωτογραφίες των γραμμάτων. Αργότερα, αναπτύχθηκαν συστήματα, τα οποία στη προσπάθειά τους να κάνουν σωστή ανάγνωση, εξέταζαν τις λέξεις ως σύνολα, λαμβάνοντας υπ' όψιν τους και το νόημα της λέξης στόχου. Πλέον η εξέλιξη αυτών των εφαρμογών μας επιτρέπει να αναπαράγουμε σε κάποιο βαθμό και τη διαμόρφωση της πρωτότυπης σελίδας.

Η επέκταση της τηλεγραφίας και η προσπάθεια ανάπτυξης συσκευών ανάγνωσης για τυφλούς, οδήγησε στην ενασχόληση με την οπτική αναγνώριση χαρακτήρων και την ανάπτυξη της με την πάροδο των χρόνων.

Το 1914 αναπτύχθηκε από τον Emanuel Goldberg μια μηχανή, η οποία αναγνώριζε χαρακτήρες και τους μετέτρεπε σε κώδικα μορς, ενώ ταυτόχρονα ο Edmund Fournier d' Albe ανέπτυξε έναν φορητό σαρωτή εικόνας ο οποίος κατά τη διάρκεια κίνησης κατά μήκος μιας γραμμής κειμένου, παρήγαγε ηχητικούς τόνους, ανάλογους του χαρακτήρα που συναντούσε. [5][8]



Εικόνα 2.16: Αριστερά ο Emanuel Goldberg και δεξιά ο Edmund Fournier d' Albe

Ο Goldberg αργότερα, πρότεινε μια μέθοδο φωτογράφισης δεδομένων τα οποία μετά, με τη χρήση φωτοκύτταρων, αντιστοιχίζονταν σε κάποια πρότυπα αναγνώρισης. Το 1929 ο Gustav Tauschek πέτυχε μια πατέντα οπτικής

αναγνώρισης χαρακτήρων στη Γερμανία και ο Handel επίσης, εξασφάλισε μια πατέντα ερασιτεχνίας που αφορούσε μια τεχνική αντιστοίχισης προτύπων το 1933 (U.S. Patent 1,915,993). Το 1935 και ο Tauschek εξασφάλισε τελικά αμερικάνικη πατέντα για τη μέθοδό του (U.S. Patent 2,026,329). [5][8]



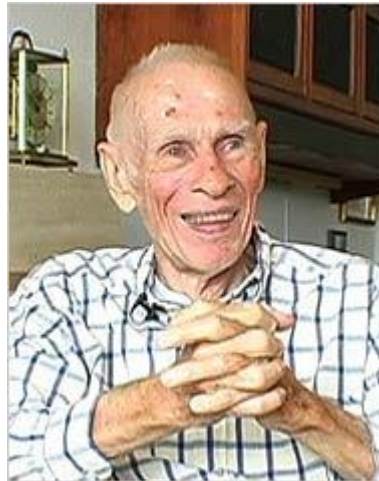
Εικόνα 2.17: Gustav Tauschek

Το 1949, ερευνητές της Ραδιοφωνικής Εταιρείας Αμερικής (Radio Corporation of America - RCA) ασχολήθηκαν με μια πρώιμη μορφή ενός συστήματος οπτικής αναγνώρισης χαρακτήρων με τη βοήθεια ηλεκτρονικού υπολογιστή για λογαριασμό της Γενικής Γραμματείας Απόστρατων των Ηνωμένων Πολιτειών Αμερικής, το οποίο όμως δε λειτουργούσε ως μετατροπέας του κειμένου, σε κείμενο υπολογιστή, αλλά ως εκφωνητής των γραμμάτων. Ήταν δηλαδή μία πρώιμη εφαρμογή text-to-speech, αλλά αποδείχθηκε πολύ ακριβή για την εποχή της ώστε να καθιερωθεί.[5][8]



Εικόνα 2.18: Ραδιοφωνική Εταιρεία Αμερικής

Το 1950 ο David H. Shepard, ένας εκ των κορυφαίων κρυπταναλυτών της Υπηρεσίας Ασφαλείας του Αμερικανικού Στρατού, ασχολήθηκε με το πρόβλημα της μετατροπής εκτυπωμένων μηνυμάτων σε γλώσσα μηχανής για την επεξεργασία των δεδομένων τους από ηλεκτρονικό υπολογιστή και κατασκεύασε μία μηχανή που την ονόμασε «Gismo». Έλαβε πατέντα για την κατασκευή του αυτή το 1953 (U.S. Patent 2,663,758). Ο «Gismo» μπορούσε να διαβάσει 23 από τα 26 γράμματα του λατινικού αλφαβήτου, να καταλάβει κώδικα μορς και μουσικά σύμβολα και να εκφωνήσει και να αντιγράψει εκτυπωμένες σελίδες. Ο Shepard προχώρησε ιδρύοντας την εταιρία IMR (Intelligent Machines Research Corporation) η οποία σε λίγο καιρό ανέπτυξε τα πρώτα συστήματα οπτικής αναγνώρισης χαρακτήρων για εμπορική χρήση.[5][8]



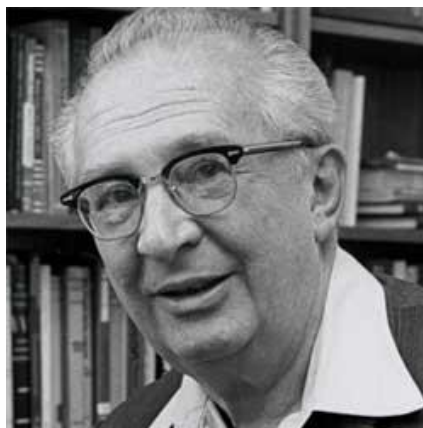
Εικόνα 2.19: David H. Shepard

Το 1955, το πρώτο εμπορικό σύστημα οπτικής αναγνώρισης χαρακτήρων εγκαταστάθηκε στα γραφεία του περιοδικού Reader's Digest και χρησιμοποιήθηκε για να εισάγει αναφορές πωλήσεων σε έναν ηλεκτρονικό υπολογιστή. Μετέτρεπε τις δακτυλογραφημένες αναφορές σε διάτρητες κάρτες τις οποίες αργότερα το προσωπικό του τμήματος συνδρομών έβαζε σ' έναν υπολογιστή που εκτελούσε την επεξεργασία των δεδομένων που αφορούσαν τις 1520 εκατομμύρια αποστολές εντύπων που εκτελούσαν κάθε χρόνο. Το δεύτερο σύστημα πουλήθηκε στην Standard Oil και χρησιμοποιήθηκε για την ανάγνωση αποδείξεων συναλλαγών πιστωτικών καρτών. Άλλα συστήματα που δημιουργήθηκαν και πουλήθηκαν από την IMR περιελάμβαναν έναν αναγνώστη τιμολογίων για την τηλεφωνική εταιρία Bell του Οχίο κι έναν σαρωτή σελίδων για την αμερικάνικη πολεμική αεροπορία ο οποίος χρησιμοποιήθηκε για την τηλεγράφιση

δακτυλογραφημένων μηνυμάτων. Τα επόμενα χρόνια ο Shepard πούλησε της πατέντες του σε διάφορες εταιρίες, μεταξύ των οποίων και στην IBM. [8]

Γύρω στο 1965, το Reader's Digest και η RCA συνεργάστηκαν για την κατασκευή ενός αναγνώστη εγγράφων που είχε ως στόχο την ψηφιοποίηση των σειριακών αριθμών κουπονιών του περιοδικού που επιστράφηκαν από διαφημίσεις. Τα γράμματα που χρησιμοποιούνταν στα έγγραφα αυτά είχαν εκτυπωθεί από έναν εκτυπωτή τυμπάνου (drum printer) της ίδιας εταιρίας με μια γραμματοσειρά ειδικά σχεδιασμένη για τους σκοπούς της οπτικής αναγνώρισης (OCR-A font). Μάλιστα ο αναγνώστης αυτός ήταν απ' ευθείας συνδεδεμένος με έναν υπολογιστή RCA 301, έναν από τους πρώτους ηλεκτρονικούς υπολογιστές τεχνολογίας ημιαγωγών (transistor technology). Ένας άλλος ειδικός αναγνώστης που επίσης αναπτύχθηκε σε συνεργασία, τοποθετήθηκε στα γραφεία της αεροπορικής εταιρίας TWA και χρησιμοποιήθηκε για την επεξεργασία του αποθέματος των καρτών επιβίβασης. Ο αναγνώστης αυτός διάβαζε έγγραφα με το ρυθμό των 1500 σελίδων το λεπτό απορρίπτοντας τις σελίδες που δεν μπορούσε να διαβάσει πλήρως. Το προϊόν αυτό μπήκε τελικά στην κανονική γραμμή παραγωγής της RCA όπου και προωθήθηκε εμπορικά τα επόμενα χρόνια για διάφορες εργασίες αντίστοιχου τύπου. [5]

Το αμερικάνικο ταχυδρομείο χρησιμοποιούσε μηχανές οπτικής αναγνώρισης χαρακτήρων ήδη από το 1965 βασιζόμενο σε τεχνολογία του μεγάλου επιστήμονα κι εφευρέτη Jacob Rabinow.



Εικόνα 2.20: Jacob Rabinow

Η πρώτη χρήση τεχνολογίας οπτικής αναγνώρισης χαρακτήρων στην Ευρώπη έγινε από το βρετανικό ταχυδρομείο. Το 1965 ξεκίνησε το σχεδιασμό ενός ολόκληρου τραπεζικού συστήματος, το «National Giro» το οποίο θα

χρησιμοποιούσε στον πυρήνα των συναλλαγών του τεχνολογία οπτικής αναγνώρισης χαρακτήρων, μια ιδέα ρηξικέλευθη η οποία άλλαξε από τότε τη μορφή των τραπεζικών συναλλαγών στη Βρετανία.[5][8]

Το καναδικό ταχυδρομείο χρησιμοποιούσε τεχνολογία αυτόματης αναγνώρισης χαρακτήρων από το 1971. Το σύστημα διάβαζε το όνομα και τη διεύθυνση του παραλήπτη και εκτύπωνε πάνω στο φάκελο έναν γραμμικό κωδικό (bar code) με οδηγίες δρομολόγησης ανάλογα με τον ταχυδρομικό κώδικα του προορισμού. Για να μην συγχέονται οι γραμμικοί κωδικοί με σύμβολα που μπορεί να σχεδιάσει ο αποστολέας οπουδήποτε πάνω στο φάκελο η εκτύπωση αυτή γινόταν με ειδικό πορτοκαλί μελάνι το οποίο είχε πολύ υψηλά ανακλαστικά χαρακτηριστικά κάτω από υπεριώδεις φως. Από αυτό το στάδιο και πέρα ήταν εύκολο τα γράμματα να ταξινομηθούν και να δρομολογηθούν μηχανικά από κάποιο σύστημα ελέγχου γραμμικών κωδικών.[5][8]

2.2.2 Πλεονεκτήματα OCR

Το εύρος των πλεονεκτημάτων που προσφέρει η οπτική αναγνώριση χαρακτήρων είναι μεγάλο. Ελαχιστοποιούνται τα έξοδα αρχειοθέτησης και αποθήκευσης πολυετών εγγράφων. Η μετατροπή ολόκληρων αρχείων εγγράφων σε ηλεκτρονικής μορφής έγγραφα, είναι αρκετά πρακτική και οικονομικά βιώσιμη.

Η χειροκίνητη εισαγωγή εγγράφων σε κάποιο ηλεκτρονικό σύστημα είναι άκρως χρονοβόρα και κατά συνέπεια αποτελεί ενέργεια υψηλού κόστους. Όπως επίσης και η ανθρώπινη αναζήτηση σε ένα μεγάλο φυσικό αρχείο κοστίζει πολύ περισσότερο σε κάποια επιχείρηση, απ'όσο κοστολογείται η εύρεση μέσω κάποιου ηλεκτρονικού συστήματος.



Εικόνα 2.21: Εισαγωγή εγγράφων σε ηλεκτρονικό σύστημα

Αξιοσημείωτη εξικονόμηση χρόνου προσφέρει επίσης και στην εισαγωγή δεδομένων σε ηλεκτρονικές φόρμες, καθώς ο μέσος όρος συμπλήρωσης φόρμας, παίρνοντας τα στοιχεία από κάποιο έντυπο έγγραφο, υπολογίζεται στα δώδεκα λεπτά. Τα έγγραφα ηλεκτρονικής μορφής είναι ευκόλως προσπελάσιμα, και ταυτοχρόνως ανακτίσιμα από πολλούς εργαζόμενους, εξαλείφοντας την ανάγκη για φωτοτυπίες και κατ' επέκταση η ελαχιστοποίηση των εκτυπώσεων, έχει περιβαλλοντικό αντίκτυπο αφού μειώνει την κατανάλωση χαρτιού και μελάνης.

Σημαντικό πλεονέκτημα αποτελεί και η αύξηση της παραγωγικότητας, καθώς η ηλεκτρονική τροποποίηση ή η αντιγραφή τμημάτων ενός εγγράφου καθίστανται εφικτές χάρη στην τεχνολογία οπτικής αναγνώρισης χαρακτήρων. Βασική είναι ακόμα και η ανάκτηση χώρου, καθώς η αρχειοθέτηση γίνεται σε ηλεκτρονικά μέσα αποθήκευσης δεδομένων, χωρίς επιπλέον έξοδα, αποφεύγοντας κατά συνέπεια τυχόν αλλοιώσεις ή απώλειες εξαιτίας της εύκολης τήρησης αρχείων ανάκτησης (back up).



Εικόνα 2.22: Από το φυσικό αρχείο στο ηλεκτρονικό

Μετά την εκχώρηση των εγγράφων σε κάποιο σύστημα, είναι δυνατό να τα ασφαλίσουμε με κωδικούς ή ακόμα και επιλεκτική πρόσβαση σε αυτά, μειώνοντας τον κίνδυνο διέρευσης απόρρητων πληροφοριών.

Θα ήταν παράλειψη να μην αναφέρουμε την ασφάλεια ανάκτησης που μας παρέχει η χρήση ηλεκτρονικών εγγράφων, αφού έχει σημειωθεί ότι περίπου το 3.5% έντυπου αρχείου σε οργανισμούς έχει απολεσθεί.



Εικόνα 2.23: Δημιουργία αντιγράφων ασφαλείας

Οι διαδικασίες που καθίστανται δυνατές, ή που βελτιώνονται, από την ύπαρξη της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων, αναμφισβήτητα αυξάνουν την παραγωγικότητα και την αποδοτικότητα της ανθρώπινης εργασίας, ανεξαρτήτως του πεδίου δραστηριοποίησης της εταιρίας ή του οργανισμού όπου εφαρμόζονται.

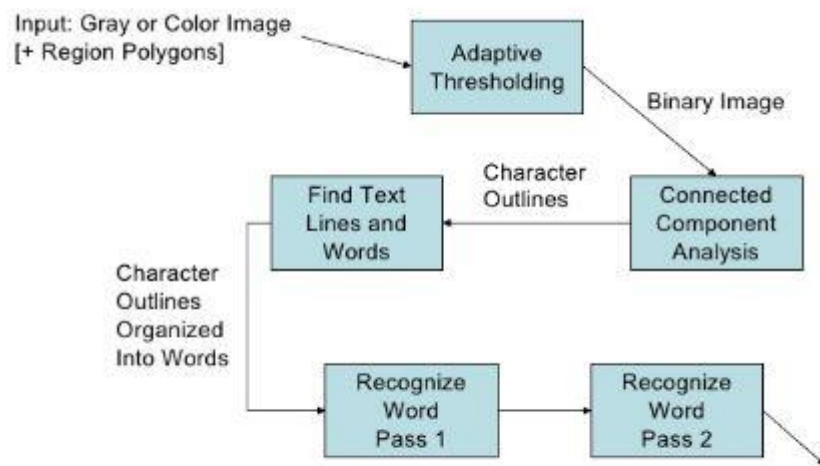
2.2.3 Βιβλιοθήκη Tesseract

Η μηχανή Tesseract αναπτύχθηκε αρχικά ως εμπορικό πρόγραμμα από την Hewlett Packard στο Bristol της Αγγλίας και στο Greeley του Κολοράντο μεταξύ 1985 ως το 1994 σε γλώσσα προγραμματισμού C. Στη συνέχεια έγιναν κάποιες αλλαγές ώστε να είναι συμβατή με λειτουργικό σύστημα Windows και ο κώδικάς της να μπορεί να μεταγλωττιστεί με compiler της C++, ενώ οι προσθήκες στον κώδικα γίνονταν πλέον σε γλώσσα C++. [8]

Η εξέλιξη σταμάτησε στα μέσα της δεκαετίας του 1990, και το 2005 δόθηκε ως πρόγραμμα ελεύθερου κώδικα από την Hewlett Packard και το πανεπιστήμιο της Νεβάδας (UNLV). Από το 2006 η εξέλιξή του χρηματοδοτείται από την Google. Σήμερα διατίθεται ως ελεύθερο λογισμικό με άδεια Apache 2.0. Είναι σχεδιασμένο

να τρέχει από γραμμή εντολών, ενώ έχουν αναπτυχθεί από τρίτους προγράμματα για αλληλεπίδραση με το χρήστη (GUI). [5]

Η μηχανή tesseract-OCR θεωρείται μία από τις καλύτερες ελεύθερες (Open Source) στην οπτική αναγνώριση χαρακτήρων, ενώ μπορεί να αναγνωρίσει κείμενο σε περισσότερες από 35 γλώσσες. Ένας τρόπος για να το επιτύχει αυτό είναι το ενσωματωμένο λεξιλόγιο, που της επιτρέπει να αναγνωρίζει σωστά λέξεις, ακόμα και αν κάποιος από τους χαρακτήρες θα αναγνωριζόταν λανθασμένα ως μεμονωμένος. [8]



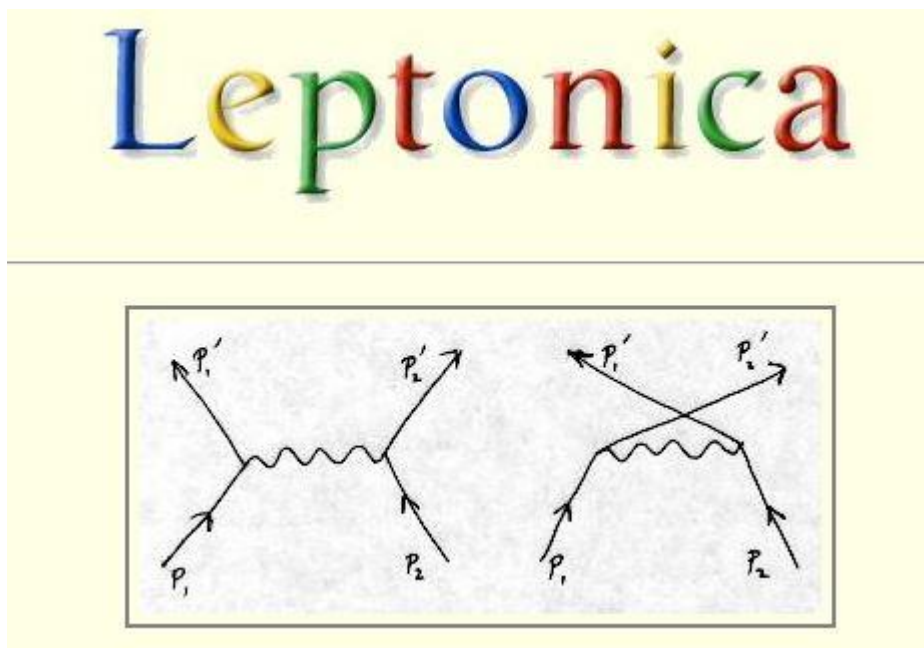
Εικόνα 2.24: Αρχιτεκτονική Tesseract

Θα ήταν σωστότερο λοιπόν να χαρακτηριστεί ως μηχανή αναγνώρισης κειμένου, αφού ο βασικός στόχος με τον οποίο δημιουργήθηκε ήταν η ψηφιοποίηση τυπωμένου και σκαναρισμένου κειμένου. Η μηχανή tesseract δίνει στο χρήστη τη δυνατότητα να προσαρμόσει κάποιες παραμέτρους, ώστε να βελτιώσει τα αποτελέσματα αναγνώρισης. Κατά τη διαδικασία αρχικοποίησης της εφαρμογής, γίνεται φόρτωση κάποιας γλώσσας και του αντίστοιχου λεξικού, δηλαδή των λέξεων που περιέχει. Ακόμη, μπορεί να προσδιοριστεί αν στην προς ανάγνωση εικόνα υπάρχει μια παράγραφος κειμένου, μια γραμμή κειμένου, μια λέξη ή ένας μεμονωμένος χαρακτήρας. [5]

Επίσης, δίνεται η δυνατότητα να προσδιοριστεί ένα υποσύνολο των χαρακτήρων, οι οποίοι θα είναι οι μόνοι που θα γίνονται δεκτοί ως αποτέλεσμα αναγνώρισης, η ονομαζόμενη «λευκή λίστα» (whitelist). Επιπλέον, μπορεί να γίνει

διαχωρισμός, ώστε στην αρχή να αναγνωρίζει μόνο γράμματα και μετά μόνο αριθμούς, κάτι που επίσης μπορεί να βελτιώσει την αναγνώριση. [5]

Η Tesseract ήταν στις 3 πρώτες μηχανές OCR όσον αφορά την ακρίβεια των χαρακτήρων κειμένων το 1995. Είναι διαθέσιμη για Linux, Windows και Mac OS X, ωστόσο, λόγω των περιορισμένων πόρων μόνο τα Windows και το Ubuntu είναι αυστηρά ελεγμένα από τους προγραμματιστές. Η Tesseract μέχρι και την έκδοση 2 μπορούσε να δεχτεί μόνο TIFF εικόνες. Από την έκδοση 3 ξεκίνησε να υποστηρίζει στην έξοδο μορφοποίηση κειμένου, και OCR πληροφορίες θέσης και μορφολογικής ανάλυσης της σελίδας. Η δυνατότητα υποστήριξης νέων μορφών εικόνας προστέθηκε χρησιμοποιώντας τη βιβλιοθήκη Leptonica. [8]



Εικόνα 2.25: Leptonica

Επίσης η Tesseract μπορεί να ανιχνεύσει αν το κείμενο είναι σταθερού πλάτους (monospaced). Οι αρχικές εκδόσεις του Tesseract μπορούσαν να αναγνωρίσουν μόνο αγγλική γλώσσα. Ξεκινώντας με την έκδοση 2 ήταν σε θέση να επεξεργάζεται Αγγλικά, Γαλλικά, Ιταλικά, Γερμανικά, Ισπανικά, Πορτογαλικά Βραζιλίας και Ολλανδικά. Ξεκινώντας με την έκδοση 3, αναγνωρίζει Αγγλικά, Αραβικά, Βουλγαρικά, Κατανικά, Τσέχικα, Κινέζικα (απλοποιημένα και παραδοσιακά), Δανέζικα (πρότυπο και Fraktur script), Γερμανικά, Ελληνικά, Φινλανδικά, Γαλλικά, Εβραϊκά, Κροατικά, Ουγγρικά, Ινδονησιακά, Ιταλικά, ιαπωνικά, Κορεατικά, της Λετονίας, της Λιθουανίας, ολλανδικά, νορβηγικά, πολωνικά, πορτογαλικά, ρουμανικά, ρωσικά, σλοβακικά (πρότυπο και Fraktur

script), της Σλοβενίας, της Ισπανίας, της Σερβίας, Σουηδικά, Ταγκαλόγκ, Ταϊλανδέζικα, Τουρκικά, Ουκρανικά και Βιετναμέζικα. Η Tesseract μπορεί να εκπαιδευθεί για να αναγνωρίζει και άλλες γλώσσες. [6]

Αν η Tesseract χρησιμοποιείται για την επεξεργασία από δεξιά προς τα αριστερά, όπως τα αραβικά ή τα εβραϊκά τα αποτελέσματα ταξινομούνται σαν να είναι από αριστερά προς τα δεξιά σε κείμενο. Η Tesseract είναι κατάλληλη για χρήση στο παρασκήνιο (backend), αλλά μπορεί να χρησιμοποιηθεί και για περισσότερο πολύπλοκα καθήκοντα OCR συμπεριλαμβανομένης ανάλυσης της διάταξης με τη χρήση ενός περιβάλλοντος διεπαφής (frontend). [6][5]

Οι εφαρμογές της είναι αμέτρητες, ενώ συνεχώς εμφανίζονται νέες. Πολλές από αυτές επικεντρώνονται στην ανάγνωση κειμένων από σαρωμένες εικόνες από έντυπα, ενώ πιο σύγχρονες περιλαμβάνουν την εύρεση του κειμένου σε σύνθετες εικόνες, ή την ανάγνωση χειρόγραφου κειμένου.



Εικόνα 2.26: Μηχανή οπτικής αναγνώρισης αρχείων

Χαρακτηριστικό των λύσεων είναι η πολυπλοκότητά τους, καθώς θα πρέπει να αντιμετωπίσουν, πέρα από τις διαφορετικές συνθήκες φωτισμού και αντίθεσης της

εικόνας, προβλήματα θορύβου, περιστραμμένου κειμένου, διαφορετικών γραμματοσειρών, ξένων συμβόλων κ.ά. [5]

2.3 Τεχνολογία αυτόματης προσαρμογής

Η χρήση κινητών συσκευών για серφάρισμα στο διαδίκτυο, αυξάνεται με αλματώδεις ρυθμούς, αλλά, δυστυχώς, ένα μεγάλο μέρος του ιστού δεν έχει βελτιστοποιηθεί για αυτές τις κινητές συσκευές.

Οι κινητές συσκευές συχνά έχουν κάποιο περιορισμό στο μέγεθος της οθόνης και απαιτούν μια διαφορετική προσέγγιση για το πώς το περιεχόμενο θα εμφανίζεται στην οθόνη. Υπάρχει μια πληθώρα από διαφορετικά μεγέθη οθόνων σε τηλέφωνα, ταμπλέτες (tablets), επιτραπέζιους υπολογιστές, κονσόλες παιχνιδιών, τηλεοράσεις. Τα μεγέθη οθόνης θα αλλάζουν πάντα, έτσι είναι σημαντικό η ιστοσελίδα μας (site) να μπορεί να προσαρμοστεί σε οποιοδήποτε ανάλυση οθόνης, σήμερα και στο μέλλον. [7]

Η τεχνολογία της προσαρμόσιμης κατασκευής διαδικτυακών εφαρμογών (responsive web design), ορίστηκε από τον Ethan Marcotte και ανταποκρίνεται πλήρως στις ανάγκες των χρηστών και τις ιδιαιτερότητες των συσκευών τους. Για παράδειγμα σε μια οθόνη επιτραπέζιου υπολογιστή, ο χρήστης θα δει το περιεχόμενο σε 4 στήλες, από ένα tablet σε δύο και από ένα κινητό σε μία στήλη προβολής.[4]



Εικόνα 2.27: Προβολή ιστοσελίδας με χρήση της τεχνολογίας του Responsive

ΚΕΦΑΛΑΙΟ 3

ΠΕΙΡΑΜΑΤΙΚΗ ΥΛΟΠΟΙΗΣΗ

3.1 Περιγραφή Εφαρμογής

Στη παρούσα Πτυχιακή εργασία, υλοποιήθηκε διαδικτυακή εφαρμογή οπτικής αναγνώρισης χαρακτήρων, όπου επιλέχθηκε η χρήση της μηχανής tesseract-OCR, η οποία είναι ελεύθερα διαθέσιμη, και θεωρείται από τις καλύτερες ελεύθερες μηχανές αναγνώρισης χαρακτήρων και συμβατή για ενσωμάτωση σε πρόγραμμα γλώσσας php. Χρησιμοποιήθηκαν επίσης και οι γλώσσες προγραμματισμού html5, css3, php και javascript.



Εικόνα 3.1: Γλώσσες προγραμματισμού HTML5, CSS3, JS, PHP

Η εφαρμογή συμβαδίζει με τις απαιτήσεις της σημερινής εποχής και βασίζεται στην γλώσσα προγραμματισμού php και Javascript. Το μόνο που είναι απαραίτητο για να μετατρέψει κάποιος ένα έντυπο έγγραφο σε επεξεργάσιμο κείμενο είναι ένας εγκατεστημένος φυλλομετρητής (Mozilla Firefox, Google Chrome, Opera, Safari, Internet Explorer κλπ).

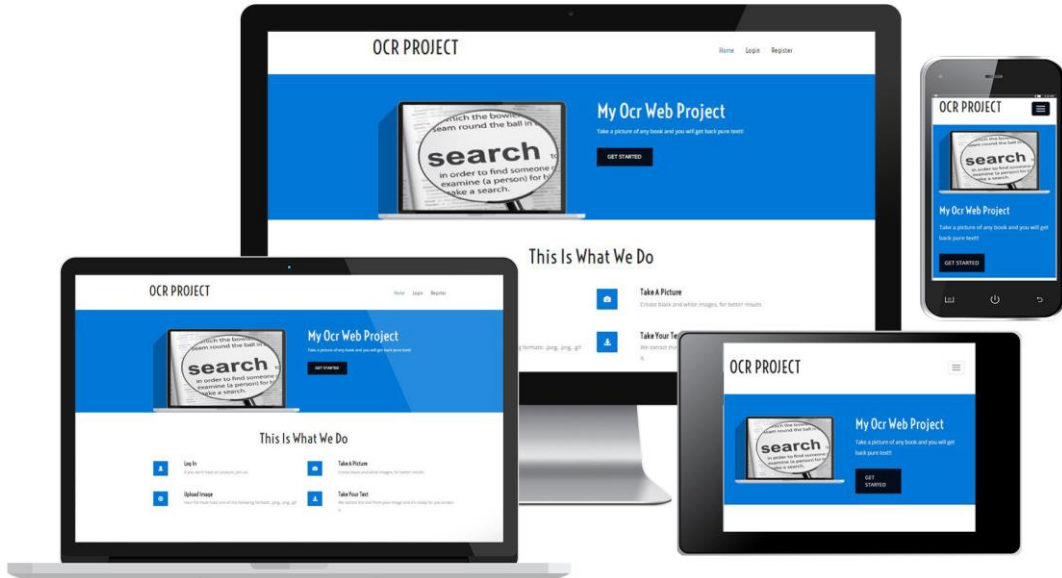


Εικόνα 3.2: Οι γνωστότεροι φυλλομετρητές

Σκοπός ανάπτυξης, είναι η ακόμα μεγαλύτερη ευκολία και αμεσότητα ενός προγράμματος στην οπτική αναγνώριση κειμένου. Οι γλώσσες Javascript και PHP όπως γνωρίζουμε έχουν απίστευτη συμβατότητα με όλες τις σύγχρονες εκδόσεις των web browsers και αυτό τις κάνει απαραίτητες στη δημιουργία νέων εφαρμογών οι οποίες θα μπορούν εύκολα να χρησιμοποιηθούν αλλά και να μεταδοθούν μέσω του internet σε μεγάλο αριθμό χρηστών, ακόμα και να αποκτήσουν εμπορική αξία ανάλογα πάντα με την επιτυχία τους.

Μεγαλύτερος στόχος σε αυτή την εφαρμογή είναι η ευκολία χρήσης και διάδοσής της με βάση τα σημερινά δεδομένα στο χώρο του διαδικτύου. Τέλος ως σκοπός μπορεί να προστεθεί και η κατανόηση των παραμέτρων που πρέπει να ληφθούν υπόψη και των προβλημάτων που πρέπει να ξεπεραστούν ώστε να φτάσουμε στο επιθυμητό αποτέλεσμα.

Όλη η εφαρμογή έχει βασιστεί στην τεχνολογία του responsive, ούτως ώστε ο χρήστης να μπορεί να χρησιμοποιήσει εύκολα την εφαρμογή μας από οποιαδήποτε συσκευή.



Εικόνα 3.3: Προβολή της εφαρμογής σε διάφορες αναλύσεις

Αρχικά, πριν ξεκινήσουμε την υλοποίηση του frontend τμήματος, είναι απαραίτητο να εγκαταστήσουμε τις απαραίτητες βιβλιοθήκες στον διακομιστή (server) που θα φιλοξενήσει την εφαρμογή μας, ώστε να είναι εφικτή η υλοποίησή της. Παρακάτω θα δούμε βήμα βήμα τον τρόπο με τον οποίο εγκαθιστούμε την βιβλιοθήκη tesseract - ocr και οτιδήποτε απαιτούμενο σε Linux Centos server.

- Αναβάθμιση του διακομιστή και εγκατάσταση απαιτούμενου πακέτου:

```
sudo yum -y update  
sudo yum -y install libtiff libtiff-devel libjpeg-devel libpng-devel gcc gcc-c++  
libtool
```
- Εγκατάσταση βιβλιοθήκης Leptonica:

```
cd /tmp/  
sudo wget http://www.leptonica.org/source/leptonlib-1.67.tar.gz  
sudo gunzip leptonlib-1.67.tar.gz  
sudo tar -xvf leptonlib-1.67.tar  
cd leptonlib-1.67/  
sudo ./configure  
sudo make  
sudo make install
```

- Εγκατάσταση πακέτου tesseract και διαμόρφωσή του για να λειτουργήσει:

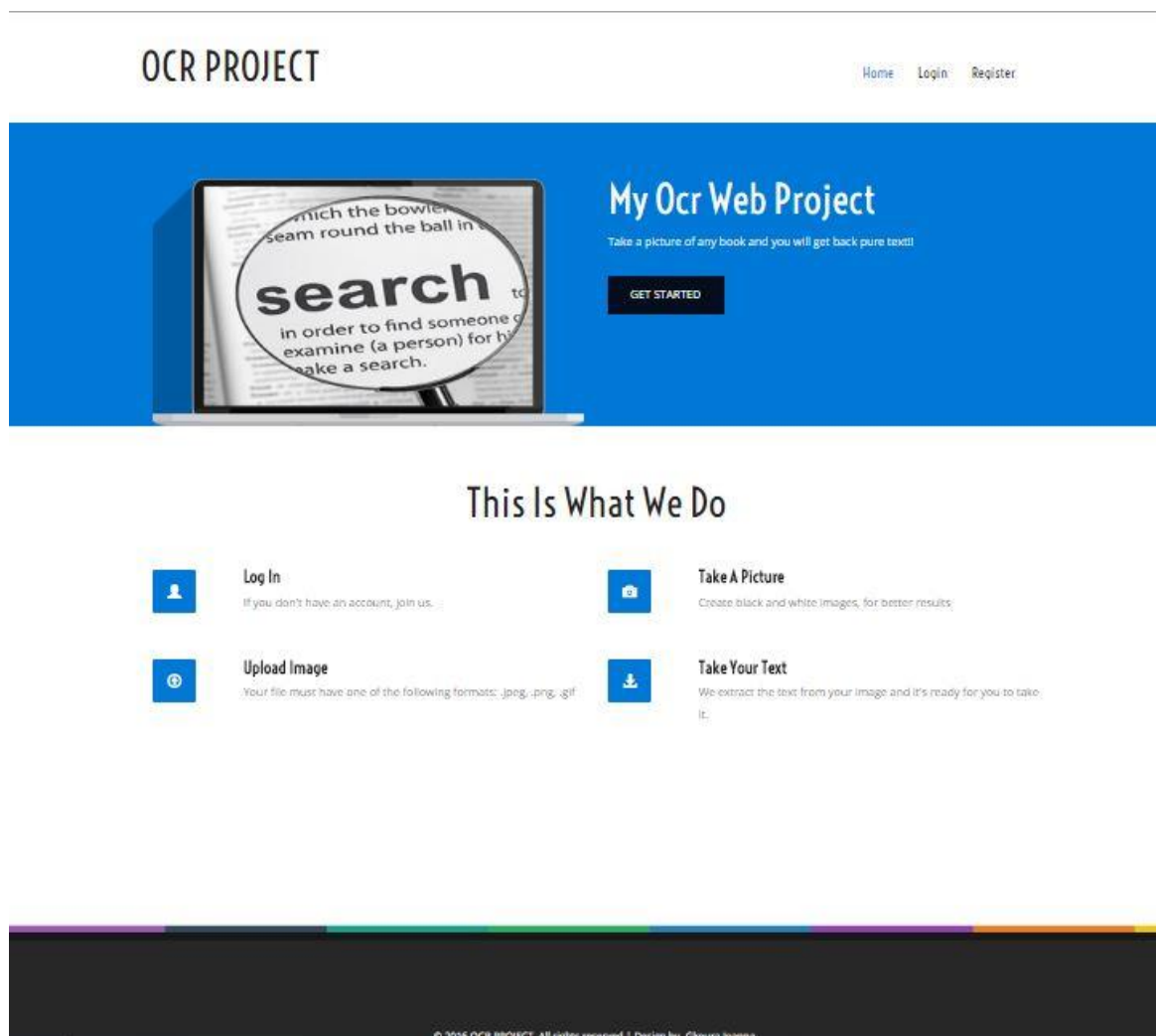
```
cd /tmp/  
sudo wget http://tesseract-ocr.googlecode.com/files/tesseract-3.00.tar.gz  
sudo gunzip tesseract-3.00.tar.gz  
sudo tar -xvf tesseract-3.00.tar  
cd tesseract-3.00/  
sudo ./runautoconf  
sudo ./configure  
sudo make  
sudo make install
```
- Εγκατάσταση εκπαιδευμένων δεδομένων γλώσσας:

```
cd /usr/local/share/tessdata  
sudo wget https://github.com/tesseract-ocr/tessdata/blob/master/grc.traineddata  
sudo wget https://github.com/tesseract-ocr/tessdata/blob/master/eng.traineddata
```
- Εγκατάσταση απλού διακομιστή δικτύου για να μπορέσει να χρησιμοποιηθεί η υπηρεσία από αρχείο PHP:

```
sudo yum -y install httpd php  
sudo chkconfig httpd on  
sudo service httpd start  
cd /var/www/html/  
nano /var/www/html/ocrproject.php
```

Αφού ολοκληρώσουμε την παραπάνω διαδικασία, δημιουργούμε με κατάλληλο κώδικα HTML5 και CSS3 το βασικό μέρος της ιστοσελίδας μας, όπου και θα φιλοξενηθεί η εφαρμογή οπτικής αναγνώρισης χαρακτήρων.

Το site μας αποτελείται από 19 σελίδες, από αυτές ο επισκέπτης έχει άμεση πρόσβαση στις 3, την αρχική σελίδα (Home), τη σελίδα σύνδεσης χρήστη (Login) και τη σελίδα εγγραφής χρήστη (Register).



Εικόνα 3.4: Αρχική σελίδα εφαρμογής

Η σελίδα στην οποία υπάρχει η εφαρμογή OCR δεν είναι ανοιχτή για τους μη εγγεγραμμένους χρήστες, οπότε αν ο επισκέπτης πατήσει στο κουμπί της αρχικής σελίδας που τον παροτρύνει να δοκιμάσει την εφαρμογή και δεν είναι συνδεδεμένος θα τον ανακατευθύνει στη σελίδα σύνδεσης.

OCR PROJECT

Home Login Register

Join Us
Create an account in order to use our services.
[Create An Account](#)

If You Have An Account, Please Log In.

Login

UserName*:
ioanna

Password*:

[SUBMIT](#)

[Forgot Password?](#)

Εικόνα 3.5: Σελίδα σύνδεσης χρήστη

Αν δεν έχει στοιχεία σύνδεσης μπορεί μέσα από τη login page να μεταφερθεί στη σελίδα εγγραφής.

OCR PROJECT

Home Login Register

Register

Your Full Name*:

Email Address*:

UserName*:
ioanna

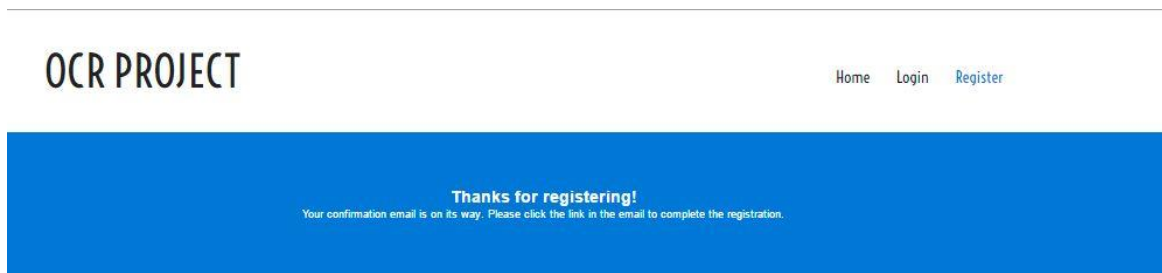
Password*:

Show Generate

[SUBMIT](#)

Εικόνα 3.6: Σελίδα εγγραφής χρήστη

Αφού συμπληρώσει τα απαραίτητα στοιχεία και επικυρώσει τη φόρμα, εμφανίζεται μήνυμα, όπου και τον ενημερώνει πως έχει ολοκληρώσει με επιτυχία την όλη διαδικασία και μπορεί να μεταβεί στο ηλεκτρονικό του ταχυδρομείο και να ακολουθήσει τον επισυναπτόμενο σύνδεσμο ενεργοποίησης της εγγραφής.

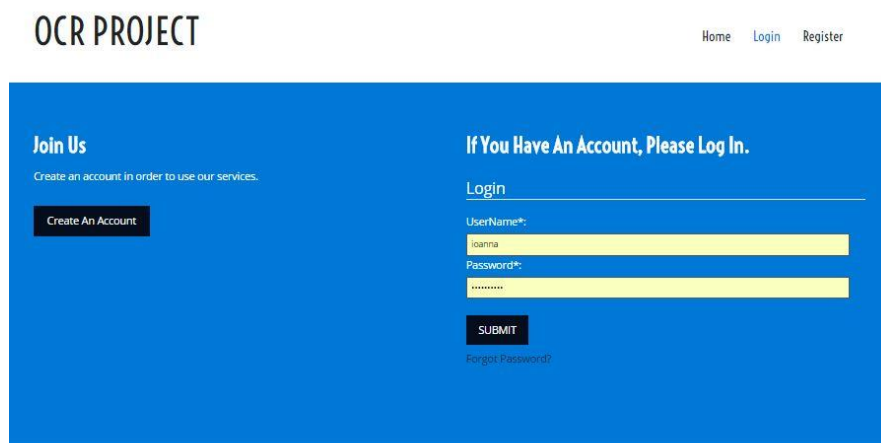


Εικόνα 3.7: Μήνυμα επιβεβαίωσης της διαδικασίας



Εικόνα 3.8: Μήνυμα ηλεκτρονικού ταχυδρομείου με τον σύνδεσμο επιβεβαίωσης

Ο σύνδεσμος τον μεταφέρει αυτόματα στην σελίδα σύνδεσης χρήστη. Όταν βάλει το όνομα χρήστη (Username) και τον κωδικό του (Password), του επιτρέπεται η πρόσβαση στην εφαρμογή.



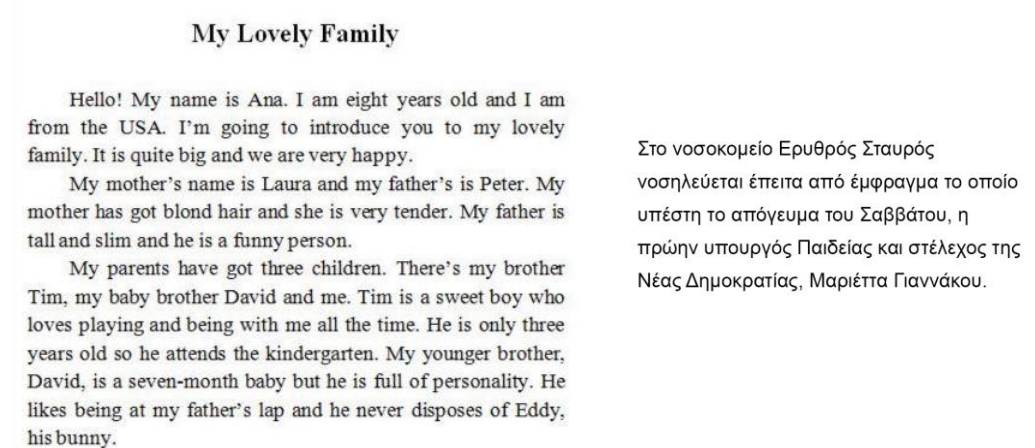
Εικόνα 3.9: Σελίδα σύνδεσης χρήστη

Αν έχει ξεχάσει τον κωδικό του, του δίνεται η δυνατότητα ανάκτησης νέου, εφόσον θυμάται τη διεύθυνση ηλεκτρονικού ταχυδρομείου την οποία είχε δηλώσει κατά τη διάρκεια της εγγραφής. Αν πάλι πατήσει το κουμπί της αποσύνδεσης, μεταφέρεται αυτόματα στη σελίδα σύνδεσης, γίνεται επανεκκίνηση της συνεδρίας, οπότε οποιαδήποτε προσπάθεια επιστροφής στην σελίδα της εφαρμογής θα είναι ανεπιτυχής. Εφόσον ο χρήστης μεταβεί στη σελίδα οπτικής αναγνώρισης χαρακτήρων, του δίνεται η δυνατότητα χρήσης της εφαρμογής.



Εικόνα 3.10: Σελίδα εφαρμογής OCR

Για να λάβει ο χρήστης το καλύτερο δυνατό αποτέλεσμα, θα πρέπει οι φωτογραφίες να είναι gray scale και η γραμματοσειρά των απεικονιζόμενων χαρακτήρων να ξεπερνά σε μέγεθος τις έντεκα μονάδες (11pt.).



Εικόνα 3.11: Αριστερά ενδεικτικό κείμενο αγγλικού κειμένου και δεξιά ελληνικού

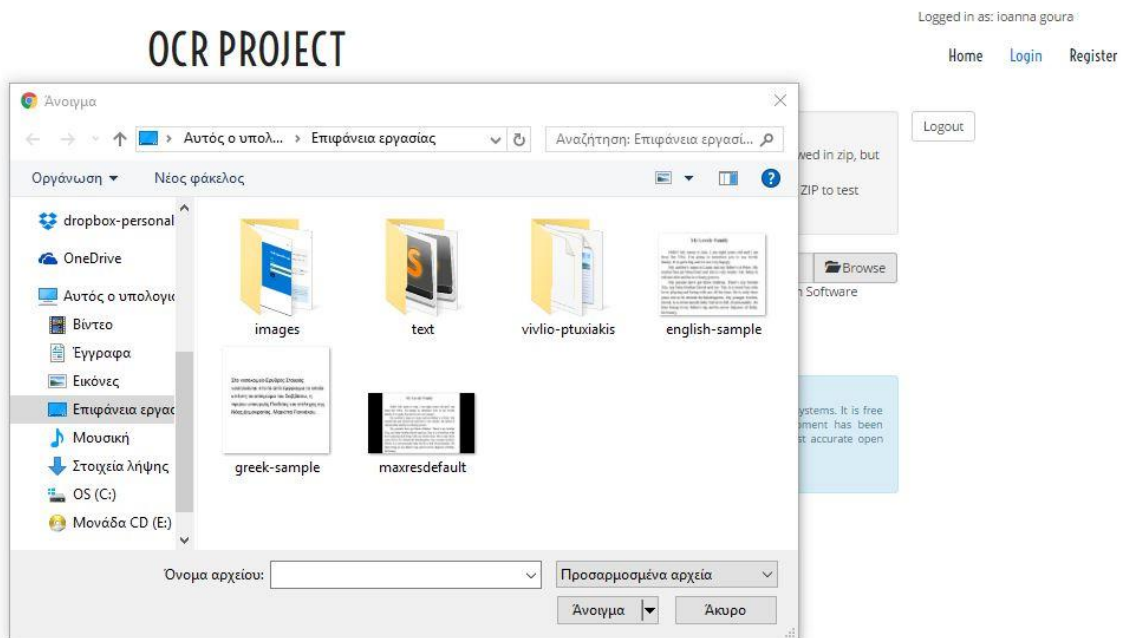
Ανάπτυξη διαδικτυακής εφαρμογής οπτικής αναγνώρισης χαρακτήρων κειμένου

Ο χρήστης θα πρέπει να επιλέξει γλώσσα μεταξύ ελληνικών και αγγλικών, σύμφωνα με τους χαρακτήρες που εμπεριέχονται στην εικόνα που θέλει να χρησιμοποιήσει.



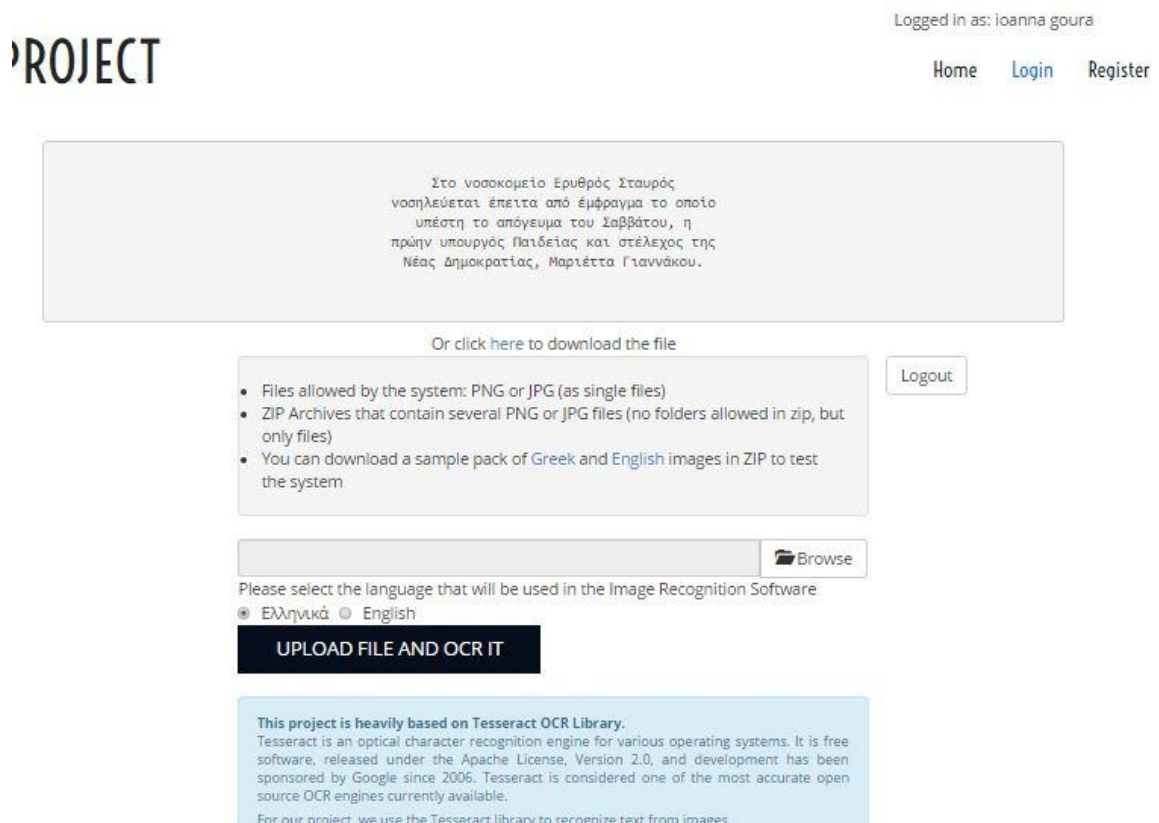
Εικόνα 3.12: Περιοχή επιλογής γλώσσας

Αφού ορίσει γλώσσα, επιλέγει μέσω της διαδικτυακής μεταφόρτωσης αρχείων τη φωτογραφία προς επεξεργασία.



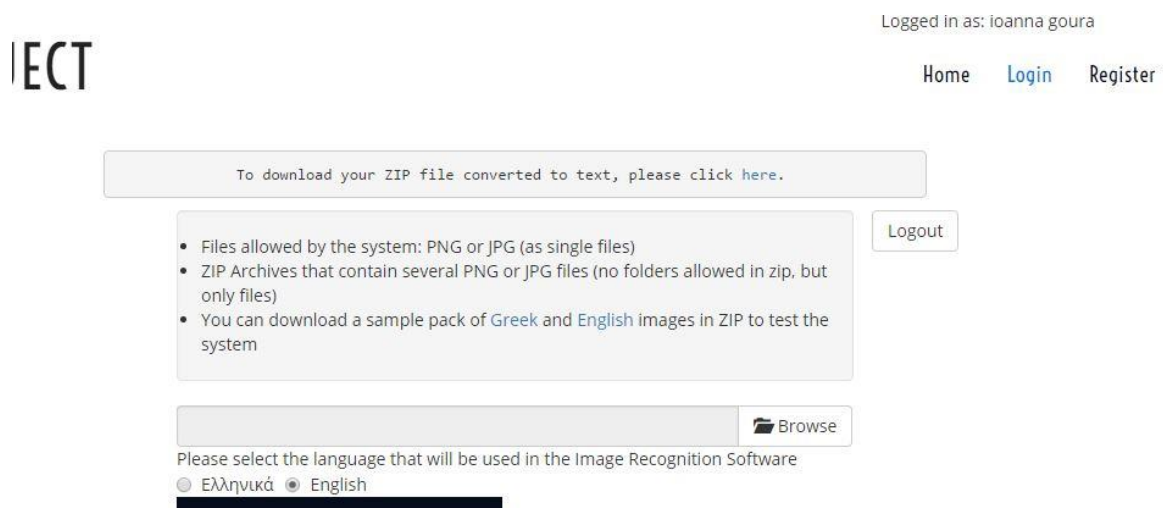
Εικόνα 3.13: Προεπισκόπηση περιβάλλοντος επιλογής αρχείου

Το αρχείο που θα επιλέξει για μεταφόρτωση επιτρέπεται να είναι .jpeg ή .png ως μεμονομένα αρχεία ή πλήθος συμπιεσμένων εικόνων .jpeg και .png σε .zip αρχείο, χωρίς όμως να εμπεριέχει υποφακέλους, παρά μόνο αρχεία εικόνας. Σε περίπτωση μεμονομένου αρχείου, μετά την μετατροπή της φωτογραφίας εμφανίζεται στο χρήστη μια προεπισκόπηση του αποτελέσματος, την οποία και μπορεί να κατεβάσει σε αρχείο κειμένου (.txt).



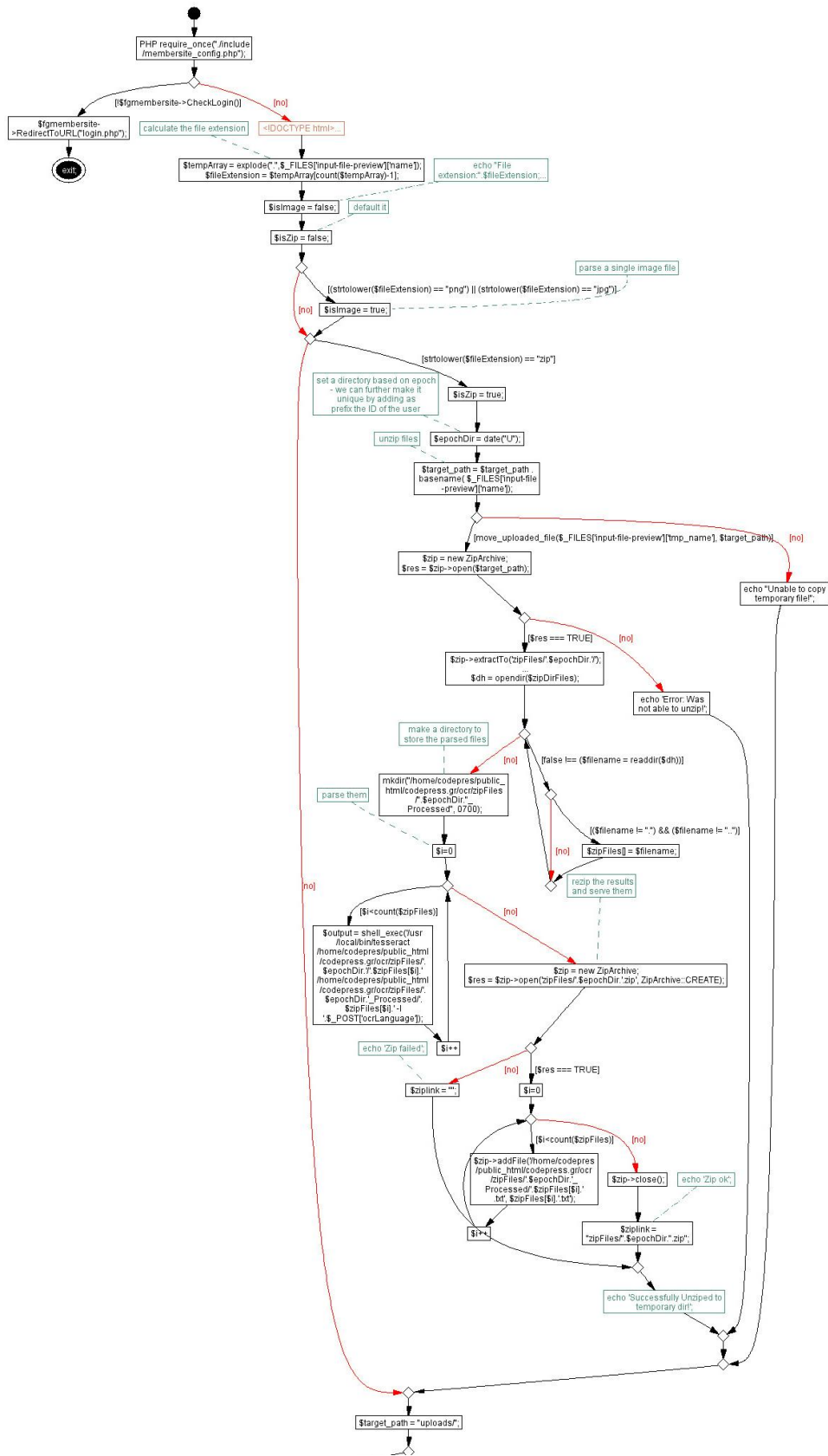
Εικόνα 3.14: Αποτέλεσμα OCR σε μορφή προεπισκόπησης και επιλογή αποθήκευσης του .txt αρχείου

Αν πρόκειται για μετατροπή πλήθους εικόνων από κάποιο .zip αρχείο, μετά την ολοκλήρωση της διαδικασίας ο χρήστης έχει την επιλογή να κατεβάσει εκ νέου ένα συμπιεσμένο αρχείο που θα περιέχει τα αντίστοιχα αρχεία κειμένου.

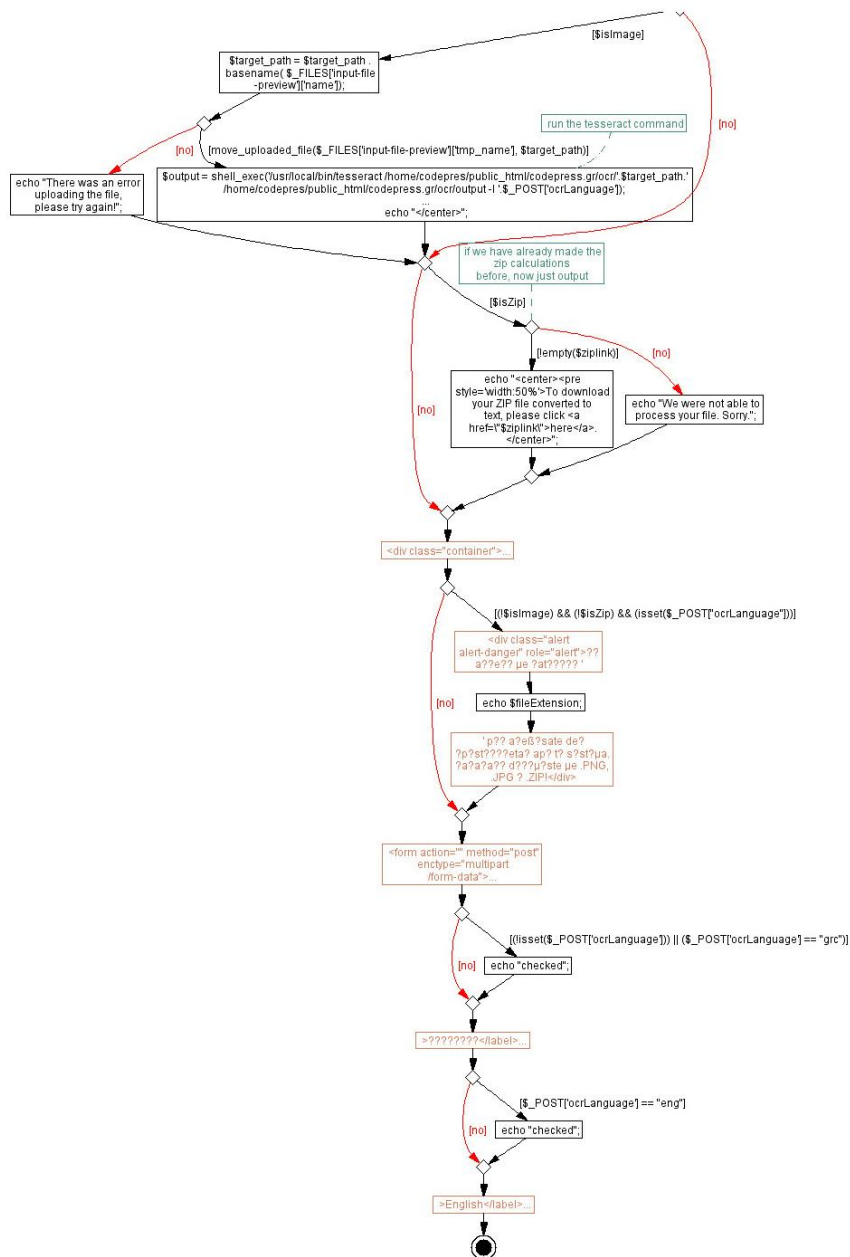


Εικόνα 3.15: Αποτέλεσμα OCR σε .zip αρχείο και επιλογή αποθήκευσης του .

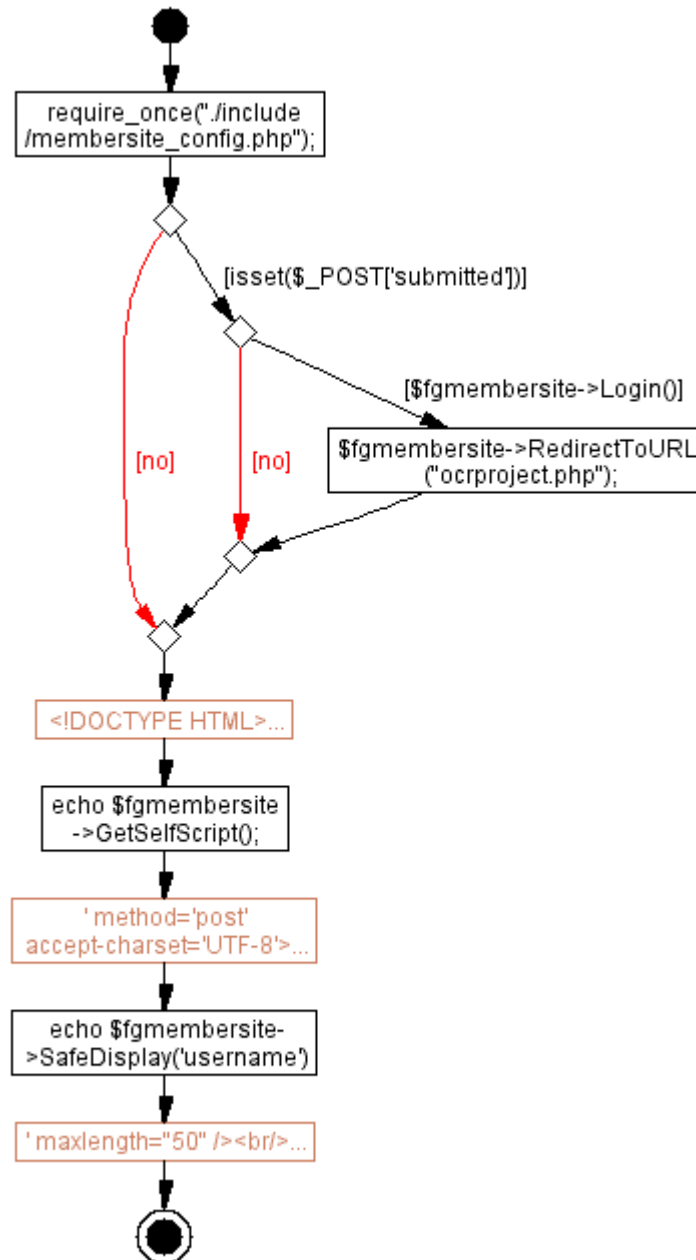
3.2 Διαγράμματα Ροής



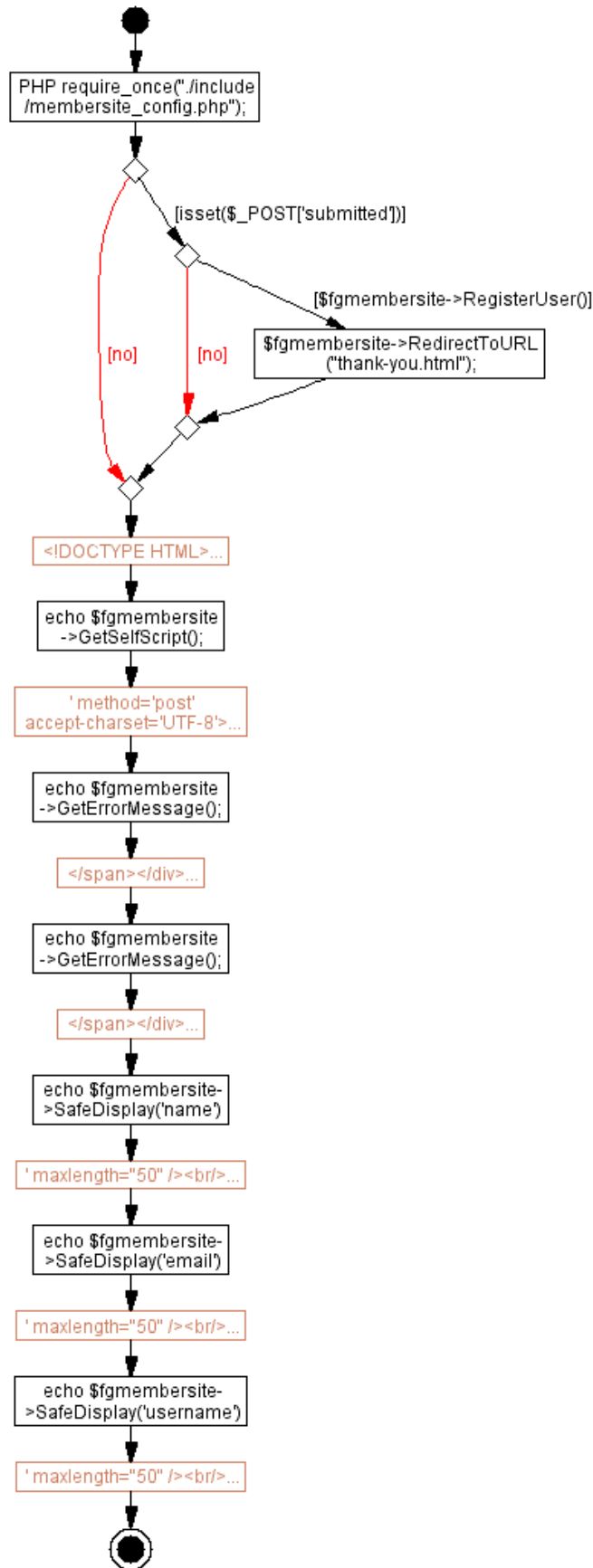
Εικόνα 3.16.1: Διάγραμμα ροής του κώδικα οcr μέρος 1



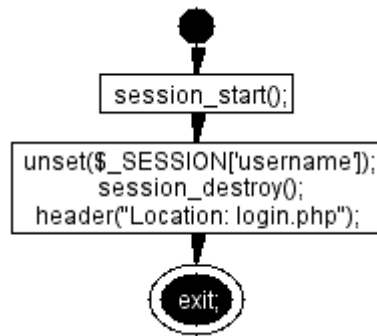
Εικόνα 3.16.2: Διάγραμμα ροής του κώδικα ocr μέρος 2



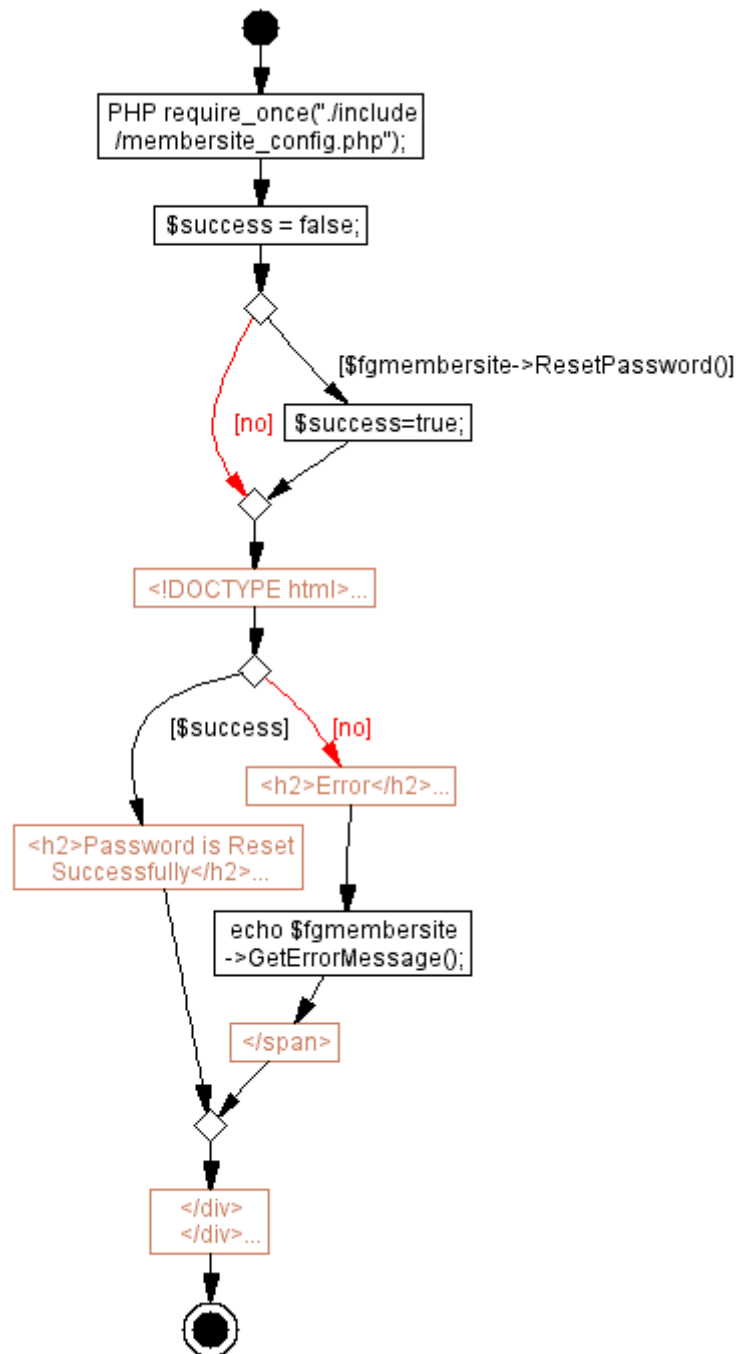
Εικόνα 3.17: Διάγραμμα ροής του κώδικα login



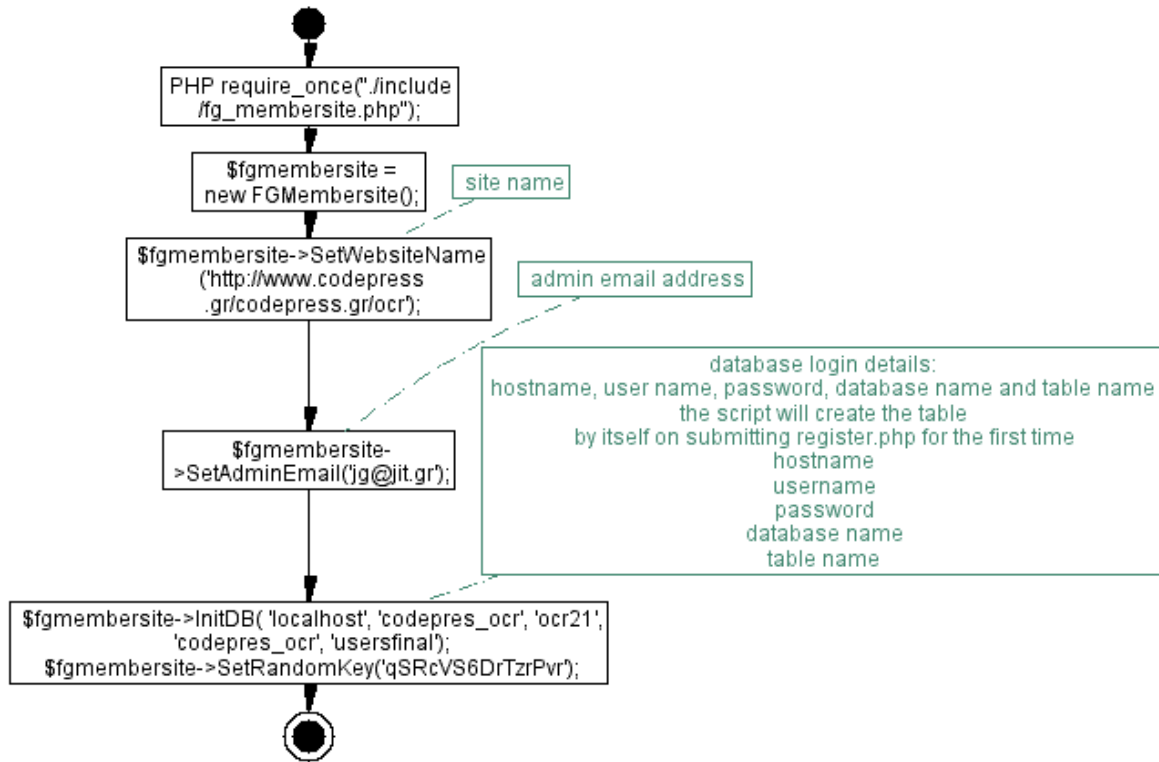
Εικόνα 3.18: Διάγραμμα ροής του κώδικα register



Εικόνα 3.19: Διάγραμμα ροής του κώδικα logout



Εικόνα 3.20: Διάγραμμα ροής του κώδικα password-reset



Εικόνα 3.21: Διάγραμμα ροής του κώδικα δημιουργίας πίνακα χρηστών

3.3 Πειραματική Προσέγγιση

Παρακάτω θα αναλύσουμε τα αποτελέσματα που πήραμε από την εφαρμογή από διαφορετικής μορφής αρχεία εικόνων, είτε στην ελληνική είτε στην αγγλική γλώσσα. Θα παρατηρήσουμε τυχόν απώλειες και σφάλματα, καθώς ανάλογα με το μέγεθος των χαρακτήρων, τη γραμματοσειρά, την μορφοποίηση του κειμένου, τα χρώματα και την ανάλυση της εικόνας, μεταβάλλεται και το ποσοστό επιτυχίας της εφαρμογής.

- Αποτέλεσμα από το πρώτο παράδειγμα με εικόνα που περιλαμβάνει ευδιάκριτους χαρακτήρες στα αγγλικά. Παρατηρούμε 100% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας του μεγέθους των χαρακτήρων, της γραμματοσειράς (Arial), την ανάλυση και την αντίθεση της εικόνας.

Lorem ipsum dolor sit amet, nullam feugiat reformidans eam te. Modo detraxit recusabo ei his, eu vix essent accusam.

Aeque nominati pri eu. Qui ne diam dissentiet, sit at meis atqui oporteat.

Dissentias intellegebat necessitatibus ne sea, nec dissentiet eloquentiam ne.

vis no, cu nec inani putent, cum an mazim iriure vidisse.

Modo detraxit recusabo ei his, eu vix essent accusam

Εικόνα 3.22: Παράδειγμα 1

“Lorem ipsum dolor sit amet, nullam feugiat reformidans earn te. Modo detraxit recusabo ei his, eu vix essent accusam. Aeque nominati pri eu. Qui ne diam dissentiet, sit at meis atqui oporteat. Dissentias intellegebat necessitatibus ne sea, nec dissentiet eloquentiam ne. vis no, cu nec inani putent, cum an mazim iriure vidisse. Modo detraxit recusabo ei his, eu vix essent Accusam”

- Αποτέλεσμα από το δεύτερο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες στα αγγλικά. Παρατηρούμε 80% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας του μεγέθους των χαρακτήρων και την ανάλυση της εικόνας.

Lorem ipsum dolor sit amet, nullam feugiat reformidans eam te. Modo detraxit recusabo ei his, eu vix essent accusam.

Aeque nominati pri eu. Qui ne diam dissentiet, sit at meis atqui oporteat.

Dissentias intellegebat necessitatibus ne sea, nec dissentiet eloquentiam ne.

vis no, cu nec inani putent, cum an mazim iriure vidisse.

Modo detraxit recusabo ei his, eu vix essent accusam

Εικόνα 3.23: Παράδειγμα 2

“Lorem Ipsum dolor sn amel, nuuam teuglat revorrmaans eam 1e Modo aetraxm recusabo ex ms, eu vlx essenl accusam Aeque nommau pn eu Qui ne mam dlssenuet, at at mews alquw oportleal Dwssenuas mlenegebal necessmaunus ne sea, nee dlssenuel emquenham ne ms nu, cu nec mam pmem, cum an mazxm mure vumsse Modo detraxlt recuszbo el ms, eu vlx assent accusam”

- Αποτέλεσμα από το τρίτο παράδειγμα με εικόνα που περιλαμβάνει μικρού μεγέθους χαρακτήρες στα αγγλικά. Παρατηρούμε 65% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας του μεγέθους των χαρακτήρων και την ανάλυση της εικόνας.

Lorem ipsum dolor sit amet, nullam feugiat reformidans eam te. Modo detraxit recusabo ei his,
eu vix essent accusam. Aequae nominati pri eu. Qui ne diam dissentiet, sit at meis atqui
oporteat. Dissentias intellegebat necessitatibus ne sea, nec dissentiet eloquentiam ne.
Ex tota causae laboramus eos, ad vel populo numquam definitionem, per bonorum imperdiet
cu. Pro nulla epicuri in, quo eu consul propriae. Albucius mentitum tincidunt cu duo, et mucius
admodum nec. Veritus maiorum et vix.
Officiis perfecto argumentum has an, nec paulo clita soleat id. Eum minim sonet voluptatibus
ex, cu tibi que indoctum vis.
?!@#\$%^&*()-+<>/";

Εικόνα 3.24: Παράδειγμα 3

"Lorem wpsum d'olor sxt amet. nuHam feugxat reformxdans eam te. Modo detraxxt recusabo ei his, eu vxx essentaccusam. Aequae nommatx pn eu. Qm ne mam dwssentiet' sit at meis atqui oporteat. Dwssermas m(eHegebat necessxtatx bus ne sea. nec dissentiet eloquentiam ne Ex lota causae Vaboramus eos. ad vel popmo numquam definitionem' per bonorum imperdiet cu. Pro mma epwcun m. quo eu consm propnae. Nbucms mentitum tincidunt cu duo' et mucius admodum nec. Ventus maworum et wx. Officus perfects argumentum has an, net: pamo chla sweat wd. Eum rmmm sonel vomplanbus ex, cu nbwque mdoctum ws.
'>!@#s%"&'()—+<>/'H "

- Αποτέλεσμα από το τέταρτο παράδειγμα με εικόνα που περιλαμβάνει ευδιάκριτου μεγέθους χαρακτήρες στα αγγλικά και μικρότερη διαφορά στην αντίθεση του χρώματος των γραμμάτων και του υπόβαθρου, σε σχέση με τα προηγούμενα παραδείγματα. Παρατηρούμε 100% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας του μεγέθους των χαρακτήρων παρόλο που τα χρώματα των χαρακτήρων πλησιάζουν το background της εικόνας.

Lorem ipsum dolor sit amet, nullam feugiat reformidans eam te. Modo detraxit recusabo ei his, eu vix essent accusam. Aequae nominati pri eu. Qui ne diam dissentiet, sit at meis atqui oporteat. Dissentias intellegebat necessitatibus ne sea, nec dissentiet eloquentiam ne. Ex tota causae laboramus eos, ad vel populo numquam definitionem, per bonorum imperdiet

Εικόνα 3.25: Παράδειγμα 4

“Lorem ipsum dolor sit amet, nullam feugiat reformidans eam te. Modo detraxit recusabo ei his, eu vix essent accusam. Aequae nominati pri eu. Qui ne diam dissentiet, sit at meis atqui oporteat. Dissentias intellegebat necessitatibus ne sea, nec dissentiet eloquentiam ne. Ex tota causae laboramus eos, ad vel populo numquam definitionem, per bonorum imperdiet”

- Αποτέλεσμα από το πέμπτο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες στα αγγλικά με μια περιπλοκότητα στη γραμματοσειρά. Παρατηρούμε 65% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της γραμματοσειράς της εικόνας.

*Lorem ipsum dolor sit amet, nullam feugiat
reformidans eam te. Modo detraxit recusabo
ei his, eu vix essent accusam. Aequae
nominati pri eu. Qui ne diam dissentiet, sit
at meis atqui oporteat. Dissentias
intellegebat necessitatibus ne sea, nec
dissentiet eloquentiam ne.*

Εικόνα 3.26: Παράδειγμα 5

“Lorem ipsum cfofor sit amet, nuffam feugiat rqformicfans eam te. 9!/locfo
cflatraxjt recusatio ei fiig eu 1/zbgessent accusam. /leque nominati pri eu. Qyi ne
diam cfissentiet, sit at meis atqui oporteat. (Dzlssentias z'nte[[ege5at
necessitatiiiius ne sea, nec cfissentiet efoquentiam ne. “

- Αποτέλεσμα από το έκτο παράδειγμα με εικόνα που περιλαμβάνει ευδιάκριτου μεγέθους χαρακτήρες στα ελληνικά. Παρατηρούμε 100% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της γραμματοσειράς, του μεγέθους των χαρακτήρων και της ανάλυσης της εικόνας.

Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.

Εικόνα 3.27: Παράδειγμα 6

“Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.”

- Αποτέλεσμα από το έβδομο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες στα ελληνικά. Παρατηρούμε 75% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της γραμματοσειράς, του μεγέθους των χαρακτήρων και της ανάλυσης της εικόνας.

Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.

Εικόνα 3.28: Παράδειγμα 7

“Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.”

- Αποτέλεσμα από το όγδοο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες και μια σειρά από σύμβολα στα ελληνικά. Παρατηρούμε 60% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της απώλειας που έχουμε στα σύμβολα γραμματοσειράς, του μεγέθους των χαρακτήρων και της ανάλυσης της εικόνας.

Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.

? ; : < > ! @ \$ % ^ & * () " . , { } [] \ |

Εικόνα 3.29: Παράδειγμα 8

“Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.

◆ ; ; < > ; ρ i 5 ; ' i i i ' 8 ' () " - ' () [] Η “

- Αποτέλεσμα από το ένατο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες και μικρή αντίθεση στα χρώματα μεταξύ γραμμάτων και υπόβαθρου εικόνας. Παρατηρούμε 80% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της αντίθεσης και της ανάλυσης της εικόνας.

Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.

Εικόνα 3.30: Παράδειγμα 9

“Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.”

- Αποτέλεσμα από το δέκατο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες και χρήση καλλιγραφικής γραμματοσειράς στα ελληνικά. Παρατηρούμε 55% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της ιδιαιτερότητας της γραμματοσειράς και της ανάλυσης της εικόνας.

Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά. Οι πρώτες εκδόσεις ήταν προγραμματισμένες με εικόνες για κάθε χαρακτήρα και δούλευαν μια γραμματοσειρά την φορά. Τα ευφυή συστήματα με υψηλό δείκτη αναγνώρισης είναι πλέον κοινά. Μερικά συστήματα είναι ικανά να αναπαράγουν ακόμη και τις πληροφορίες που δεν είναι κείμενο σε ένα έγγραφο, όπως εικόνες, στήλες, γραμμές, γωνίες κτλ.

Εικόνα 3.31: Παράδειγμα 10

“Το(οπλοτήματο(Οπτικής Αναγνώρισης Χαρικκτήρων ατιοατούν βαθμονόμηση γιο(νο(διαβάσουν μω(ουγκεκριμένη γραμμιχτοσειρά. οι τιρώτες εκδόσεις ήτιχν προγραμματισμένες με εικόνες γιο(κάθε χαρακτήρο(κοα δούλευαν μιο(γραμμιχτοσειρό(την φορό(Το(ευφνή ουστήματο(με νψηλο' δείκτη ανιχγνώρισης είνοα πλέον κοινό(Μερικό(οπλοτήματο(είνιχι ικικνό(νο(ικνιχτιαράγουν ακόμη κοα τις πληροφορίες τιοι δεν είνιχι κείμενο σε ένο(έγγραφο, ότιως εικόνες, στήλες, γραμμές, γωνίες κτλ.”

- Αποτέλεσμα από το ενδέκατο παράδειγμα με εικόνα που περιλαμβάνει μεσαίου μεγέθους χαρακτήρες και στα αγγλικά και στα ελληνικά. Παρατηρούμε 50% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της επιλογής αναγνώρισης γλώσσας σε αγγλικά, ενώ οι χαρακτήρες της εικόνας περιλαμβάνουν και ελληνικούς χαρακτήρες.

Τα συστήματα Οπτικής Αναγνώρισης Χαρακτήρων απαιτούν βαθμονόμηση για να διαβάσουν μια συγκεκριμένη γραμματοσειρά.

Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs

Εικόνα 3.32: Παράδειγμα 11

“Tor ouotnpdrd O1'rTiKr'1g AVdVVUup|0'i']§ Xdpo<KTr1pwn drrourou Bdeponopnon via vd 5io<Bo'rooUv pid oUvKeKripévn vpdppdtoosipd. Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs”

- Αποτέλεσμα από το δωδέκατο παράδειγμα με εικόνα που περιλαμβάνει διάφορα μεγέθη χαρακτήρων στα αγγλικά. Παρατηρούμε 75% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της αύξησης του μεγέθους των χαρακτήρων και της ανάλυσης της εικόνας.

A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789

Εικόνα 3.33: Παράδειγμα 12

“. um um... rm mu 0.. u. my -my unxasazn
A Quid: Brmm my umus Dverme my my nmass
A am Brvwn my lumns OverTn: iazy nag 0123456739
A Quick Bhwtn Fox Jumps OverTh2 lazy Dog 0123455739
A Quick Brown Fnx Jumps Over The Lazy Dog 0123455799
A Quilt Bmwtn Fox Jumps overme Lazy Dog 0123456789
A Quick Brawn Fox Jumps Over The Lazy Dag 0123456789
A Quick Brown Fox Jumps over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789
A Quick Brown Fox Jumps Over The Lazy Dog 0123456789”

- Αποτέλεσμα από το δεκατο τρίτο παράδειγμα με εικόνα που περιλαμβάνει μορφοποίηση στο κείμενο στα αγγλικά. Παρατηρούμε 70% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας της μορφοποίησης του κειμένου της εικόνας.

- This subtitle is first.
- This is the *second* subtitle shown.
- Red with single underline: render
- Black with double underline: copy from current event
- Blue with dotted underline: copy from previous event

Εικόνα 3.34: Παράδειγμα 13

“- This subtitle is first.

th second bti_tl_g.shown._

- Red with single underline: render

- Black with double underline: (:02; from

current event

- Blue with dotted__u_n_derline' _g__(_)__py__f1j9_ni_”

- Αποτέλεσμα από το δεκατοτέταρτο παράδειγμα με εικόνα που περιλαμβάνει μορφοποίηση στο κείμενο στα αγγλικά. Παρατηρούμε 99% επιτυχία στο αποτέλεσμα και αυτό εξαιτίας του εικονιδίου στη θέση χαρακτήρα που υπάρχει στη λέξη «own» της εικόνας.

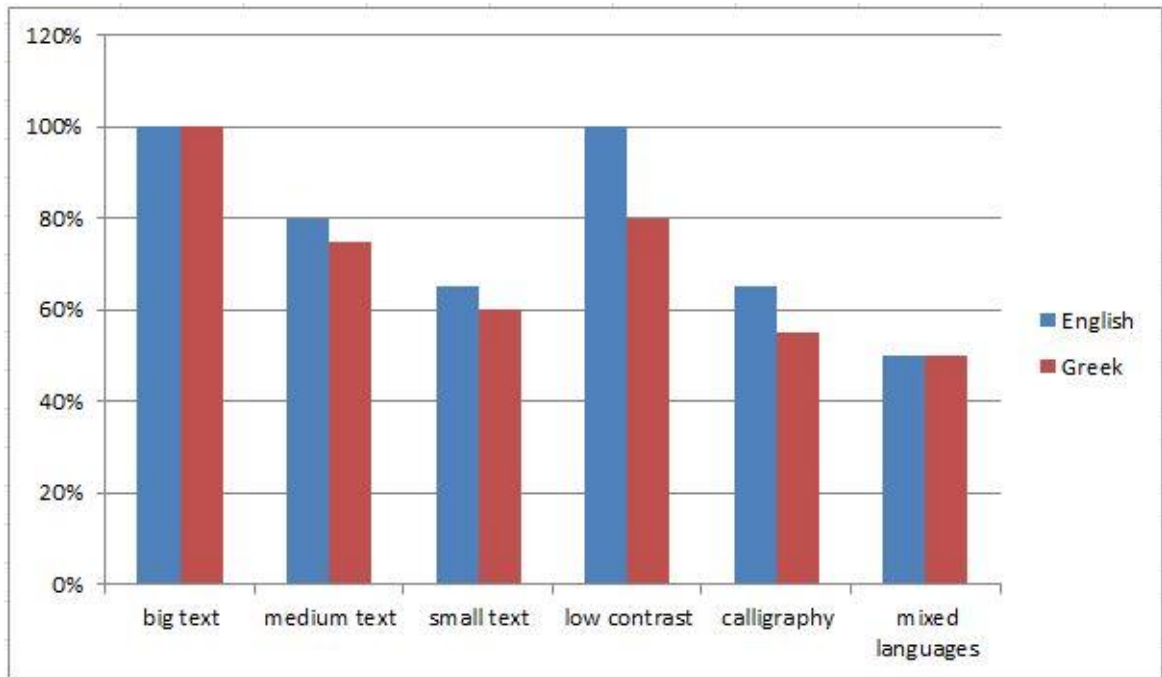
my-wn-voice

Voice replacement solution
available in 10 languages - Apply now

Εικόνα 3.35: Παράδειγμα 14

“my—Qwn—voice

Voice replacement solution
available in 10 languages — Apply now”

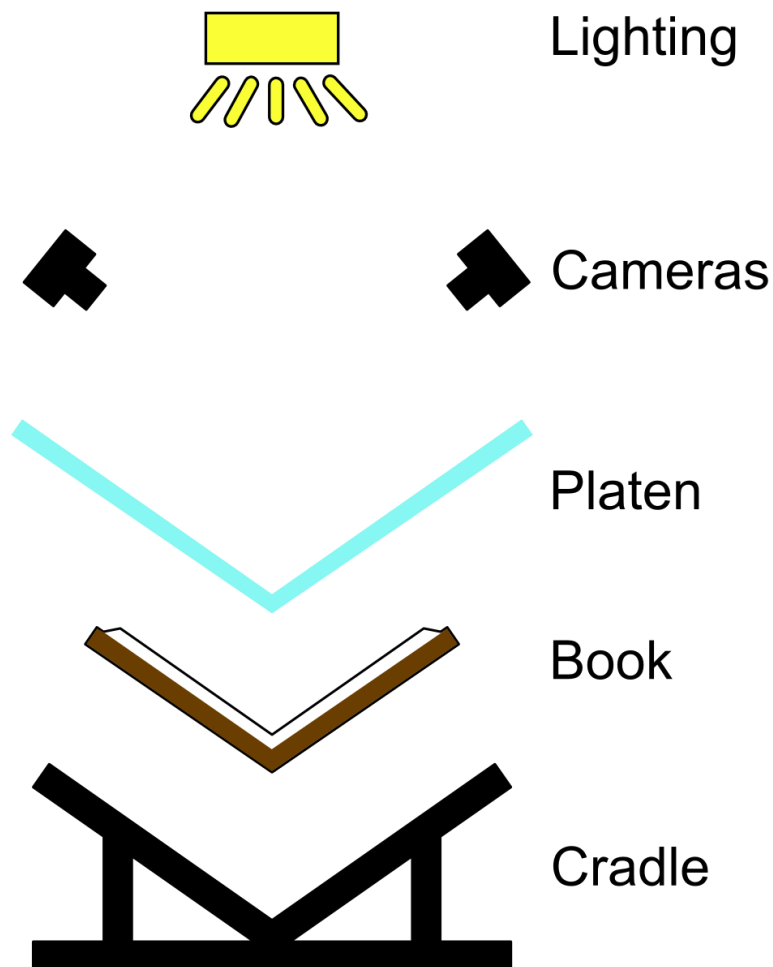


Εικόνα 3.36: Γράφημα Αποτελεσμάτων

3.4 Μελέτη Συστήματος Ψηφιοποίησης

Κατά τη διάρκεια της υλοποίησης της πτυχιακής εργασίας, δημιουργήθηκε η ιδέα ανάπτυξης ενός αυτοματοποιημένου συστήματος ψηφιοποίησης εγγράφων. Κατά καιρούς έχουν υλοποιηθεί αρκετά τέτοια συστήματα, εκ των οποίων άλλα λειτουργούν χειροκίνητα και άλλα αυτόματα, με τα τελευταία να ξεπερνάνε τα 1700\$. Μετά την ολοκλήρωση κάποιων ερευνών σχετικά με τους τρόπους λειτουργικής υλοποίησης ενός αυτόματου σαρωτή, έγινε καταγραφή των απαραίτητων εξαρτημάτων και σχεδίαση του μοντέλου. Παρακάτω βλέπουμε τα απαραίτητα στοιχεία:

- ✓ Δύο ξύλα σε σχήμα V που κρατάει το βιβλίο ανοιχτό (για την προστασία των πιο ευαίσθητων βιβλίων).
- ✓ Μία πλάκα γυαλί ή ακρυλικό σε σχήμα V, η οποία ισοπεδώνει τις σελίδες
- ✓ Δύο κάμερες συμβατές με πλακέτα Arduino οι οποίες λαμβάνουν εικόνες από τις σελίδες. Μία αριστερά και μία δεξιά. (ArduCAM shield to MEGA 2560)
- ✓ Μία λάμπα.
- ✓ Μία πλακέτα Arduino Mega 2560
- ✓ Δύο βηματικούς κινητήρες για την αλλαγή σελίδων
- ✓ Ένα σιδηρόδρομο με αλυσίδα και γρανάζι



Scanner-Design (image) 2016, Diybookscanner [www.diybookscanner.org]

Εικόνα 3.37: Αναπαράσταση Συστήματος Ψηφιοποίησης

Το κόστος κατασκευής με τα παραπάνω υλικά, υπολογίζεται στα 375 ευρώ. Σε ένα τέτοιο σύστημα ψηφιοποίησης, εφόσον ολοκληρωθεί η σάρωση κάποιου βιβλίου, τοποθετούμε τη μνήμη από την μνήμη από την κάμερα και συνδέοντάς την σε κάποιο ηλεκτρονικό υπολογιστή, μπορούμε να ανεβάσουμε το .zip αρχείο με τις φωτογραφίες στην εφαρμογή που έχουμε δημιουργήσει και μετά την διαδικασία OCR να αποθηκεύσουμε τα .txt αρχεία, έχοντας το βιβλίο πλέον σε ηλεκτρονική επεξεργάσιμη μορφή κειμένου.

ΚΕΦΑΛΑΙΟ 4

ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

4.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η αξιοποίηση των εφαρμογών μπορεί να είναι και πέρα από το πεδίο της πληροφορικής. Η οπτική αναγνώριση χαρακτήρων μπορεί να χρησιμοποιηθεί για διάφορους σκοπούς, όπως η ασφάλεια, η επιβεβαίωση στοιχείων και η μετατροπή όγκου κειμένου σε ψηφιακή μορφή.

Η εφαρμογή που υλοποιήθηκε και αναλύσαμε στο προηγούμενο κεφάλαιο, μπορεί να αξιοποιηθεί για την περαιτέρω κατανόηση της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων και γενικά της έννοιας επεξεργασία εικόνας. Χρησιμοποιήσαμε τις δυνατότητες που μας δίνουν οι σύγχρονες γλώσσες web προγραμματισμού, προσφέροντας στο χρήστη ένα λειτουργικό και εύχρηστο περιβάλλον.

Βλέποντας τα αποτελέσματα της πειραματικής προσέγγισης και παρατηρώντας τις απώλειες που είχαμε κατά τη μετατροπή των έντυπων χαρακτήρων σε ψηφιακό κείμενο, συμπεραίνουμε ότι στην παρούσα εφαρμογή οι παράγοντες που επηρεάζουν άμεσα το ποσοστό επιτυχίας του OCR, είναι το μέγεθος των χαρακτήρων, η ανάλυση και η αντίθεση της εικόνας, όπως επίσης και η γραμματοσειρά του κειμένου.

4.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Βασική προϋπόθεση για την υλοποίηση εφαρμογών και τη μακροχρόνια χρήση τους, αποτελεί η αναβάθμιση (update). Θα πρέπει τακτικά να γίνεται έλεγχος από τον προγραμματιστή, αν κάποια από τις γλώσσες, τις οποίες έχει χρησιμοποιήσει στην εφαρμογή, έχει βγάλει κάποια καινούργια έκδοση. Θα πρέπει ο κώδικας του να ενημερώνεται συνεχώς και να ακολουθεί την εξέλιξη.

Πέραν όμως αυτού, υπάρχουν και πολλοί τρόποι να επεκτείνουμε την παρούσα όμως εφαρμογή. Ο πιο εύκολος και παράλληλα συντομότερος τρόπος επέκτασης, είναι η προσθήκη περισσότερων γλωσσών.

Ένας άλλος τρόπος να εξελίξουμε την εφαρμογή, είναι εκπαιδύοντας τη βιβλιοθήκη tesseract-ocr και κάνοντας τις απαραίτητες ρυθμίσεις, να μπορέσουμε

να υλοποιήσουμε σε ένα μικρό έστω ποσοστό, οπτική αναγνώριση χαρακτήρων σε χειρόγραφο έντυπο.

Επίσης, κρατώντας τη λογική και μετατρέποντας τον κώδικα PHP και js, σε Java και XML, μπορούμε να τη μετατρέψουμε σε Web Application για συσκευές Android. Κάτι το οποίο ανταποκρίνεται στις απαιτήσεις της εποχής και μπορεί να δώσει και επιπλέον δυνατότητες στο χρήστη, ενεργοποιώντας παραδείγματος χάριν την κάμερα του κινητού του για άμεση λήψη φωτογραφίας και επεξεργασίας της στην εφαρμογή.

ΠΑΡΑΡΤΗΜΑ Α'

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης του βασικού αρχείου της παρούσας εφαρμογής.

```
<?PHP
require_once("../include/membersite_config.php");

if(!$fgmembersite->CheckLogin())
{
    $fgmembersite->RedirectToURL("login.php");
    exit;
}
?>
<script>
$(document).on('click', '#close-preview', function(){
    $('.image-preview').popover('hide');
    // Hover befor close the preview
    $('.image-preview').hover(
        function () {
            $('.image-preview').popover('show');
        },
        function () {
            $('.image-preview').popover('hide');
        }
    );
});
$(function() {
    // Create the close button
    var closebtn = $('<button/>', {
        type:"button",
        text: 'x',
```

```
        id: 'close-preview',
        style: 'font-size: initial;',
    });
    closebtn.attr("class","close pull-right");
    // Set the popover default content
    $('.image-preview').popover({
        trigger:'manual',
        html:true,
        title:
"<strong>Preview</strong>" + $(closebtn)[0].outerHTML,
        content: "There's no image",
        placement:'bottom'
    });
    // Clear event
    $('.image-preview-clear').click(function(){
        $('.image-preview').attr("data-
content","").popover('hide');
        $('.image-preview-filename').val("");
        $('.image-preview-clear').hide();
        $('.image-preview-input input:file').val("");
        $(".image-preview-input-title").text("Browse");
    });
    // Create the preview image
    $(".image-preview-input input:file").change(function (){
        var img = $('<img/>', {
            id: 'dynamic',
            width:250,
            height:200
        });
        var file = this.files[0];
        var reader = new FileReader();
        // Set preview image into the popover data-content
        reader.onload = function (e) {
```

```
        $(".image-preview-input-title").text("Change");
        $(".image-preview-clear").show();
        $(".image-preview-filename").val(file.name);
        img.attr('src', e.target.result);
        //          $(".image-preview").attr("data-
content",$(img)[0].outerHTML).popover("show");
    }
    reader.readAsDataURL(file);
});
});
</script>
<?
// calculate the file extension
$tempArray = explode(".",$_FILES['input-file-preview']['name']);
$fileExtension = $tempArray[count($tempArray)-1];
//echo "File extension: ".$fileExtension;
$isImage = false; // default it
$isZip = false; // default it
if((strtolower($fileExtension) == "png") ||
(strtolower($fileExtension) == "jpg")) {
    // parse a single image file
    $isImage = true;
}
if(strtolower($fileExtension) == "zip") {
    $isZip = true;
    $epochDir = date("U"); // set a directory based on epoch -
we can further make it unique by adding as prefix the ID of the
user
    // unzip files
    $target_path = $target_path . basename( $_FILES['input-file-
preview']['name']);
```

```
if(move_uploaded_file($_FILES['input-file-
preview']['tmp_name'], $target_path)) {
    $zip = new ZipArchive;
    $res = $zip->open($target_path);
    if ($res === TRUE) {
        $zip->extractTo('zipFiles/' . $epochDir . '/');
        $zip->close();
        $zipDirFiles = 'zipFiles/' . $epochDir . '/';
        $dh = opendir($zipDirFiles);
        while (false !== ($filename = readdir($dh))) {
            if(($filename != ".") && ($filename != "..")) {
                $zipFiles[] = $filename;
            }
        }

        // make a directory to store the parsed files

mkdir("/home/codepres/public_html/codepress.gr/ocr/zipFiles/" . $epochDir . "_Processed", 0700);
        // parse them
        for($i=0;$i<count($zipFiles);$i++) {
            $output = shell_exec('/usr/local/bin/tesseract
/home/codepres/public_html/codepress.gr/ocr/zipFiles/' . $epochDir .
 '/' . $zipFiles[$i] .
/home/codepres/public_html/codepress.gr/ocr/zipFiles/' . $epochDir .
 _Processed/' . $zipFiles[$i] . ' -l ' . $_POST['ocrLanguage']);
        }
        // rezip the results and serve them
        $zip = new ZipArchive;
        $res = $zip->open('zipFiles/' . $epochDir . '.zip',
ZipArchive::CREATE);
        if ($res === TRUE) {
            for($i=0;$i<count($zipFiles);$i++) {
```



```
        $zip-
>addFile('/home/codepres/public_html/codepress.gr/ocr/zipFiles/' . $
epochDir . '_Processed/' . $zipFiles[$i] . '.txt',
$zipFiles[$i] . '.txt');
    }
    $zip->close();
    //echo 'Zip ok';
    $ziplink = "zipFiles/" . $epochDir . ".zip";
} else {
    //echo 'Zip failed';
    $ziplink = "";
}
//echo 'Successfully Unzipped to temporary dir!';
} else {
echo 'Error: Was not able to unzip!';
}

} else { echo "Unable to copy temporary file!"; }
}
$target_path = "uploads/";

if($isImage) {
    $target_path = $target_path . basename( $_FILES['input-
file-preview']['name']);
    if(move_uploaded_file($_FILES['input-file-
preview']['tmp_name'], $target_path)) {
        // run the tesseract command
        $output = shell_exec('/usr/local/bin/tesseract
/home/codepres/public_html/codepress.gr/ocr/' . $target_path .
/home/codepres/public_html/codepress.gr/ocr/output -l
' . $_POST['ocrLanguage']);
        echo "<center><pre style='width:50%'>$output <br>";
        $output = shell_exec('cat output.txt ');
    }
}
```

```
        echo "$output </pre>";
        echo          "Or          click          <a
href='download.php?file=output.txt' target=\"_blank\">here</a> to
download the file";
        echo "</center>";
    } else{
        echo "There was an error uploading the file, please
try again!";
    }
}
if($isZip) { // if we have already made the zip calculations
before, now just output
    if(!empty($ziplink)) { echo          "<center><pre
style='width:50%'>To download your ZIP file converted to text,
please click <a href=\"\$ziplink\">here</a>. </center>"; }
    else { echo "We were not able to process your file.
Sorry."; }
}
?>
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Digital image processing, G Kang - Quest, vol. 1, Autumn 1977, p. 2-20., 1977.
- [2] Digital image processing algorithms and applications, I Pitas -2000-
books.google.com.
- [3] Method and apparatus for real-time lossless compression and decompression of image data, SM Callahan - US Patent 5,408,542, 1995 - Google Patents.
- [4] Responsive Web Design Version of 06 Dec 2011 Eva Harb, Paul Kapellari, Steven Luong, Norbert Spot
- [5] Tesseract OCR Engine: What it is, where it came from, where it is going., Ray Smith, Google Inc, OSCON 2007
- [6] International Journal of Computer Applications (0975 – 8887) Volume 55– No.10, October 2012
- [7] Responsive Web Design with HTML5 and CSS3, by Ben Frain
- [8] A Study on Tesseract Open Source Optical Character Recognition Engine, Banerjee, Saikat
- [9] Content Based Image Retrieval: Survey and Comparison between RGB and HSV model, Simardeep Kaur 1 and Dr. Vijay Kumar Banga
- [10] Hue-Saturation-Density (HSD) Model for Stain Recognition in Digital Images From Transmitted Light Microscopy Jeroen A.W.M. van der Laak,* Martin M.M. Pahlplatz, Antonius G.J.M. Hanselaar, and Peter C.M. de Wilde, 1999
- [11] Image File Formats: Past, Present, and Future¹ Richard H. Wiggins III, MD H.Christian Davidson, MD.H. RicHarnsberger, MD Jason R. Lauman, BS Patricia A. Goede, BS