



**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Διπλωματική Εργασία

**Καταγραφή real-time ιατρικών μετρήσεως από
μη – διασυνδεδεμένες ιατρικές συσκευές μέσω
εφαρμογής android**

Καγιαλίκος Βασίλειος
ΑΜ: 841

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΡ. ΑΓΓΕΛΙΔΗΣ ΠΑΝΤΕΛΗΣ

Οκτώβριος 2020, Κοζάνη



**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Διπλωματική Εργασία

**Καταγραφή real-time ιατρικών μετρήσεως από
μη – διασυνδεδεμένες ιατρικές συσκευές μέσω
εφαρμογής android**

Καγιαλίκος Βασίλειος
ΑΜ: 841

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΡ. ΑΓΓΕΛΙΔΗΣ ΠΑΝΤΕΛΗΣ

Οκτώβριος 2020, Κοζάνη

Copyright © 2020 – Καγιαλίκος Βασίλειος

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Περίληψη

Καθώς η τεχνολογία γίνεται ολοένα και περισσότερο μέρος της καθημερινότητας τόσο των νεότερων όσο και των ηλικιωμένων ανθρώπων, ο καλύτερος τρόπος για να την αξιοποιήσουμε είναι βελτιώνοντας τον τομέα υγείας. Αυτό θα το πετύχουμε αυτοματοποιώντας τις διαδικασίες καταγραφής και αποθήκευσης των μετρήσεων από ιατρικές συσκευές. Έτσι απλοποιώντας την διαδικασία μειώνεται ο κίνδυνος λάθους και καταστεί την εφαρμογή διαθέσιμη για όλους, ακόμα και για τον πιο αρχάριο αφού χρειάζεται μόνο τρία κλικ για να “διαβάσει” και να καταχωρίσει μια νέα μέτρηση. Η εφαρμογή μπορεί να φαίνεται ιδιαίτερα απλή για τον τελικό χρήστη, δεν ισχύει το όμως ίδιο και για της διαδικασίες που εκτελούνται στο παρασκήνιο. Οι διαδικασίες αυτές περιλαμβάνουν επεξεργασία και ενίσχυση της εικόνας σε αρκετά στάδια, εντοπισμό και απομόνωση περιοχής ενδιαφέροντος, αυτόματη αναγνώριση κειμένου και πολλά άλλα. Τέλος για πετύχουμε όσο δυνατόν καλύτερα ποσοστά χρειάστηκε να εκπαιδεύσουμε την δικιά μας γλώσσα έτσι ώστε οι πιθανότητες να χρειαστεί ο χρήστης να κάνει διορθώσεις να μειώνεται σημαντικά.

Λέξεις κλειδιά: Android, OpenCV, Tesseract, OCR, οπτική αναγνώριση, βάση δεδομένων, ηλεκτρονική υγεία, ιατρικές συσκευές

Abstract

As technology becomes more and more part of the daily lives of both young and old people, the best way to take advantage of it is by improving the health sector. We can achieve this by automating the processes of recording and storing measurements by medical devices. By simplifying the process reduces the risk of error and makes the application easy to use by everyone, even the beginner since it only need three clicks to "read" and register a new measurement. The application may seem very simple to the end user, but the processes needed to make it work is not as simple. These processes include editing and enhancing the image in several stages, locating and isolating an area of interest, automatic text recognition and much more. Finally, in order to achieve the best possible recognition percentage, we had to train our own language so that the chances of the user having to make corrections are significantly reduced.

Keywords: Android, OpenCV, Tesseract, OCR, visual recognition, database, electronic health record, medical devices

Ευχαριστίες

Αρχικά θα ήθελα να απευθύνω θερμές ευχαριστίες στον επιβλέποντα καθηγητή της παρούσας διπλωματικής εργασίας, κύριο Αγγελίδη Παντελή που με καθοδήγησε σε όλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας με χρήσιμες συμβουλές καθώς και με τη βιβλιογραφία που έπρεπε να επικεντρώσω για τη συγγραφή της εργασίας.

Ιδιαίτερα θερμές ευχαριστίες θέλω να δώσω στην οικογένεια μου για την συνεχή συμπαράσταση του, για τις πολύτιμες συμβουλές τους και για όλα όσα μου έχουν προσφέρει όλα αυτά τα χρόνια της ζωής μου αλλά και των σπουδών μου.

ΕΙΣΑΓΩΓΗ	10
Κεφάλαιο 1	11
1.1 Δομή Εργασίας	11
1.2 Γιατί tesseract	12
1.3 Γιατί android	16
1.4 Γιατι Opencv	17
1.5 Μηχανική μάθηση	19
Κεφάλαιο 2	20
2.1 PHP	20
2.2 Visual Studio Code	21
2.3 Android Studio.	22
2.4 MySQL	23
2.5 REST	24
2.6 Json	25
Κεφάλαιο 3	26
3.1 Βασικές λειτουργίες Opencv	26
3.1.1 Εισαγωγή Opencv στο Android Studio	34
3.2 Βασικές λειτουργίες tesseract	35
3.2.1 Εύρεση των βασικών γραμμών (Finding the baselines)	35
3.2.3 Προσαρμογή βασής (Baseline fitting)	35
3.2.4 Τμηματοποίηση χαρακτήρων (Character segmentation)	36
3.2.5 Συσχετισμός σπασμένων χαρακτήρων	36
3.2.6 Ταξινόμηση χαρακτήρων (Character Classification)	37
3.2.7 Δεδομένα Εκπαίδευσης (Training Data)	38
Κεφάλαιο 4	42
4.1 Εντοπισμός οθόνης	42
4.2 Αλγόριθμος απομόνωσης οθόνης	42
4.3 Επεξεργασία αριθμών οθόνης	46

4.3.1	Αλγόριθμος επεξεργασία αριθμών οθόνης	47
4.3.2	Αναγνώριση κείμενου	51
4.3.3	Αλγόριθμος αυτόματης αναγνώρισης	51
Κεφάλαιο 6		52
6.1	Σχεδιασμός βάσης δεδομένων	52
6.2	Λειτουργίες της εφαρμογής	54
6.3	Σύνδεση εφαρμογής με την βάση δεδομένων	62
Κεφάλαιο 7		66
7.1	Συμπεράσματα	66
7.2	Μελλοντικές επεκτάσεις	66
Βιβλιογραφία		67

ΕΙΣΑΓΩΓΗ

Σκοπός της εργασίας είναι η δημιουργία εφαρμογής η οποία θα καθιστά εύκολη την καταγραφή σε πραγματικό χρόνο των ιατρικών μετρήσεων, από μη διασυνδεδεμένες ιατρικές συσκευές με μοναδική προϋπόθεση ο χρήστης να έχει android smartphone, όποτε και μηδενικό επιπλέον κόστος.

Καθώς υπάρχει η δυνατότητα αποθήκευσης των μετρήσεων δημιουργείτε αυτόματα ένας ηλεκτρονικό ιατρικό φάκελος ο οποίος επιτρέπει την πληρέστερη αντιμετώπιση του ασθενούς από τον θεράποντα ιατρό του, αφού ο τελευταίος έχει στην διάθεση του συνεχώς όλες τις πληροφορίες που σχετίζονται με την μέχρι εκείνη στιγμή πορεία της υγείας του πρώτου. Ο ασθενής θα ήταν ιδιαίτερα χρονοβόρο να κρατάει σε καθημερινή βάση το ιστορικό των μετρήσεων του, ενώ ακόμα και αν το έκανε τα δεδομένα αυτά δεν θα ήταν ασφαλές καθώς δεν θα υπήρχαν σε ηλεκτρονική μορφή άλλα ούτε και εύκολα προσβάσιμα από οποιοδήποτε μέρος.

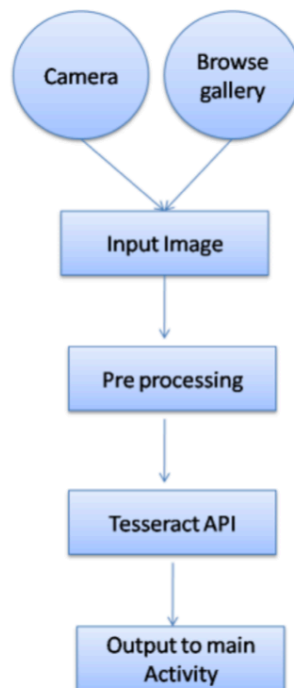
Η εφαρμογή επίσης θα πρέπει να είναι όσο το δυνατόν πιο εύχρηστη καθώς απευθύνεται σε άτομα μεγαλύτερης ηλικίας που δεν έχουν εξοικείωση με τα smartphones, κάτι που έχει επιτευχθεί καθώς αφού ο ασθενής βγάλει φωτογραφία με το κινητό του, ο αλγόριθμος διαβάζει της μετρήσεις αυτόματα χωρίς να χρειάζεται να τις πληκτρολογήσει ξεχωριστά, απλοποιώντας έτσι την διαδικασία.

Ένα σημαντικό ακόμα πλεονέκτημα είναι ότι δεν περιορίζεται σε μια μόνο ιατρική συσκευή, αλλά υπάρχει υποστήριξη για τις δημοφιλέστερες, οι οποίες είναι πιεσόμετρο, μετρητές γλυκόζης και οξύμετρο.

Κεφάλαιο 1

1.1 Δομή Εργασίας

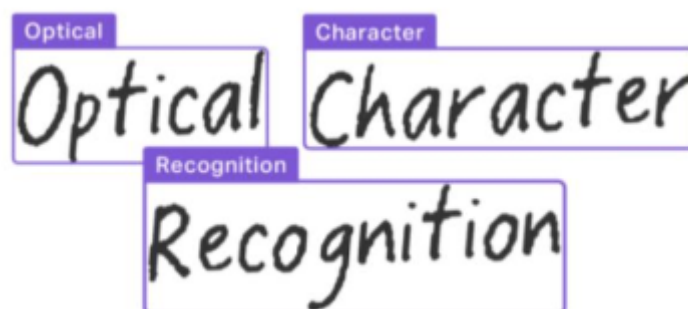
Το Κεφάλαιο αυτό, έχει ως στόχο να κάνει μία συνοπτική αναφορά, στα επιμέρους στοιχεία-αντικείμενα, τα οποία αποτελούν βάση για την παρούσα Διπλωματική Εργασία, καθώς και μια σύντομη περιγραφή της λειτουργίας της εφαρμογής. Αρχικά χρησιμοποιώντας τις βιβλιοθήκες του android ο χρήστης επιλέγει μια φωτογραφία από το κινητό ή από την κάμερα προς επεξεργασία. Στην συνέχεια με χρήση της βιβλιοθήκης OpenCV επεξεργάζεται την εικόνα αυτή κάνοντας τους χαρακτήρες πιο ευανάγνωστους και τέλος με χρήση της μηχανής αναγνώρισης Tesseract, αναγνωρίζει το κείμενο και το επιστρέφει στον χρήστη για να το αποθηκεύσει .



Διάγραμμα ροής αναγνώρισης εικόνας

Στην συνέχεια, αναφέρονται ορισμένα στοιχεία για το λειτουργικό σύστημα android για το οποίο θα δημιουργηθεί η εφαρμογή, τα επιστημονικά πεδία που αφορούν την όραση σε συστήματα τεχνολογίας (υπολογιστική όραση), και την μηχανική μαθηση και οι βιβλιοθήκες που χρησιμοποιήθηκαν πέραν των βασικών βιβλιοθηκών που έχει το android οι οποίες κατέστησαν δυνατή την επεξεργασία εικόνων και την αναγνώριση κείμενου από τις εικόνες αυτές !

1.2 Γιατί tesseract



Η Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition) ή αλλιώς Αυτόματη Αναγνώριση Χαρακτήρων Κειμένου ονομάζεται η διαδικασία μετατροπής σαρωμένων εικόνων χειρογράφων ή έντυπων κειμένων σε κείμενο αναγνώσιμο από ηλεκτρονικό υπολογιστή. Η Οπτική Αναγνώριση Χαρακτήρων καθιστά εφικτή την εκ νέου επεξεργασία του κειμένου, αποφεύγοντας την δακτυλογράφηση του από την αρχή.

Χρησιμοποιείται για την εξαγωγή δεδομένων από κάποιες μορφές έντυπου, όπως είναι τα έγγραφα, τα βιβλία, τα γράμματα αλληλογραφίας, οι αποδείξεις πωλήσεων κτλ. Δηλαδή είναι μέθοδος με την οποία ψηφιοποιούνται έντυπα κειμένων έτσι ώστε να είναι προσπελάσιμα ηλεκτρονικά, να μπορούν αποθηκεύονται σε μικρότερο χώρο, να είναι προσβάσιμα από τον παγκόσμιο ιστό και να μπορούν να χρησιμοποιηθούν σε διάφορες διαδικασίες όπως η αυτόματη μετάφραση, η εξόρυξη δεδομένων ή η μετατροπή κειμένου σε φωνή από τον ηλεκτρονικό υπολογιστή.

Η αντιστοίχιση με πρότυπα αφορά την αναγνώριση χαρακτήρων από έτοιμα πρότυπα ή περιγράμματα χαρακτήρων. Ο σαρωτής ψηφιοποιεί την εικόνα ενός εγγράφου στον υπολογιστή και το λογισμικό Οπτικής Αναγνώρισης προσπαθεί να ταιριάξει, με ένα βαθμό πιθανότητας, τους χαρακτήρες από το σαρωμένο αρχείο εικόνας με τα πρότυπα που έχει αποθηκευμένα. Αν η εικόνα ενός χαρακτήρα αντιστοιχεί με αναγνωρισμένο χαρακτήρα, τότε αντιστοιχίζεται με χαρακτήρα κειμένου για τον ηλεκτρονικό υπολογιστή.

Tesseract



Λογότυπο της βιβλιοθήκης Tesseract

Μια από τις σημαντικότερες επιλογές που έπρεπε να γίνουν έτσι ώστε να διασφαλίσουμε την σωστή λειτουργία της εφαρμογής ήταν αυτή της τεχνολογίας αναγνώρισης κειμένου. Οι τεχνολογίες αυτές χωρίζονται σε δυο μεγάλες κατηγορίες, η μια από αυτές είναι οι online υπηρεσίες μέσω API όπως το Cloud Vision API της Google, το Amazon Rekognition της Amazon, Microsoft Image Recognition API της Microsoft και άλλα, οι οποίες παράγουν πολύ καλά αποτελέσματα και έχουν δημιουργηθεί και υποστηρίζονται από εταιρίες κολοσσούς.

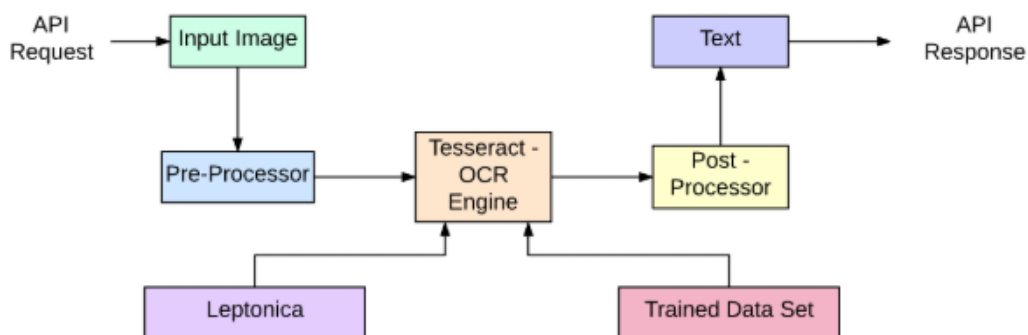
Παρόλα αυτά για την υλοποίηση της εφαρμογής θα χρησιμοποιηθεί μια τεχνολογία από την δεύτερη κατηγορία που είναι οι offline βιβλιοθήκες. Αυτό γίνεται κυρίως γιατί δεν θέλουμε η εφαρμογή να απαιτεί σύνδεση στο διαδίκτυο, αλλά και επειδή οι offline βιβλιοθήκες είναι συνήθως open source άρα δεν υπάρχει κόστος για την χρήση τους και προσφέρουν περισσότερη παραμετροποίηση σε σχέση με τις online όπου είναι μηδαμινή.

Κάποιες από τις δημοφιλέστερες offline βιβλιοθήκες αναγνώρισης κειμένου είναι οι tesseract, simpleCV, Ocropus, SwiftOCR, OpenALPR κ.α. Η βιβλιοθήκη θα πρέπει προφανώς να μπορεί να χρησιμοποιηθεί σε java/android άρα λύσεις όπως η SwiftOCR (που είναι μόνο για iOS) απορρίπτονται, έτσι επιλέγουμε την δημοφιλέστερη από τις παραπάνω την tesseract, για την οποία έχουν δημιουργηθεί τα περισσότερα εργαλεία παραμετροποίησης, έχει τα καλύτερα αποτελέσματα σε ακρίβεια αναγνώρισης άλλα και ταχύτητα και είναι υπό την εποπτεία και την συντήρηση της Google.

Tesseract-OCR είναι μια μηχανή ανοιχτού κώδικα. Είναι γραμμένο σε C / C ++ και αναπτύχθηκε στα εργαστήρια της Hewlett-Packard (HP) μεταξύ 1985 και 1996. Αλλά ποτέ δεν το χρησιμοποιούσαν στα προϊόντα τους. Ήταν μία από τις κορυφαία 3 μηχανές OCR στη δοκιμή ακρίβειας στο UNLV 1995 [6]. Το 2005, κυκλοφόρησε ως μηχανή ελεύθερου λογισμικού. Από το 2007, η Google έχει λάβει την εποπτεία της μηχανής αναγνώρισης χαρακτήρων (tesseract-OCR) για την περαιτέρω ανάπτυξη και τη συντήρηση της. Παίρνει γκρι ή έγχρωμη εικόνα ως είσοδο και δίνει έξοδο σε μορφή κειμένου. Στην αρχή, μόνο .tiff τύπος εικόνων υποστηριζόταν αλλά τώρα υποστηρίζει επίσης άλλους τύπους εικόνων, όπως .png, .jpg, κλπ Μπορεί να διαβάσει τα δεδομένα σε οποιαδήποτε γλώσσα από την εικόνα όπως τα αγγλικά, σουηδικά, δανικά, κλπ και προγραμματιστές μπορούν να “εκπαιδεύσουν” τη δική τους γλώσσα, εάν η υποστήριξη για μια συγκεκριμένη γλώσσα δεν είναι διαθέσιμη. Μπορεί να χρησιμοποιηθεί σε Ubuntu, Windows αλλά και σε άλλες πλατφόρμες Linux και Mac, κλπ. Μπορούμε επίσης να χρησιμοποιήσουμε αυτή τη βιβλιοθήκη σε κινητές πλατφόρμες όπως το Android και iPhone κ.λπ.

Το Tesseract 4.00 περιλαμβάνει ένα νέο υποσύστημα νευρωνικού δικτύου που έχει διαμορφωθεί ως αναγνώριση γραμμής κειμένου. Έχει τις ρίζες του στην εφαρμογή LSTM με βάση το Python του OCRopus, αλλά έχει επανασχεδιαστεί για το Tesseract στο C ++. Το νευρωνικό σύστημα δικτύου στο Tesseract προ-χρονολογεί το TensorFlow αλλά είναι συμβατό με αυτό, καθώς υπάρχει μια γλώσσα περιγραφής δικτύου που ονομάζεται Variable Graph Specification Language (VGSL), η οποία είναι επίσης διαθέσιμη για το TensorFlow.

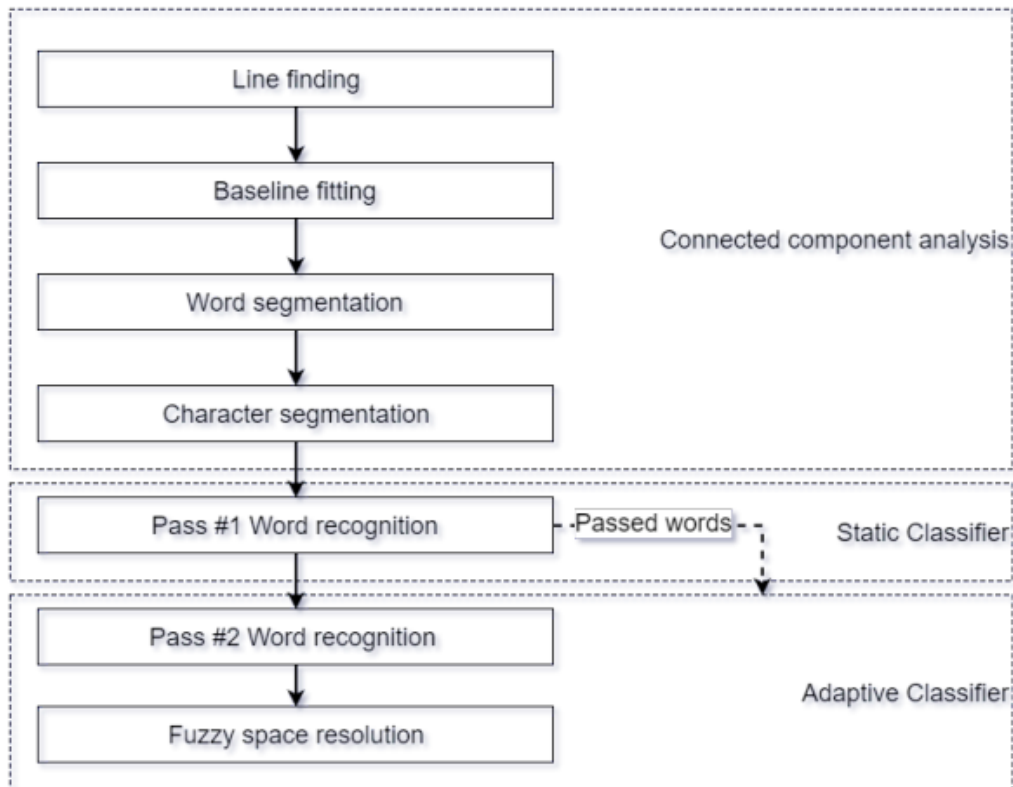
OCR Process Flow



Διαδικασία Ocr

Πως λειτουργεί η Tesseract;

Η αρχιτεκτονική ανάλυσης Tesseract είναι ενσωματωμένη σε μια επαναληπτική διαδικασία αγωγού, αλλά επαναλαμβάνει τα παλιά βήματα. Η αναγνώριση γίνεται δύο φορές. κατά την πρώτη αναγνώριση εκτελείται ένας στατικός ταξινομητής χρησιμοποιείται και για την δεύτερη χρησιμοποιείται ο προσαρμοστικός ταξινομητής. Το Tesseract έχει σχεδιαστεί για να αναγνωρίζει το κείμενο ακόμη και αν έχει ένα μικρή λοξότητα χωρίς να χρειάζεται να την διορθώσει, παρόλο που είναι προτιμότερο να έχει το κείμενο οριζόντιο για καλύτερη αναγνώριση. Το πρώτο μέρος της διαδικασίας αναγνώρισης είναι η ανάλυση συνδεδεμένων συστατικών. Περιλαμβάνει την εύρεση γραμμής, την προσαρμογή βάσης, και η τμηματοποίηση λέξεων και χαρακτήρων. Στη συνέχεια, κάθε λέξη που περνά στον στατικό ταξινομητή στέλνεται στο προσαρμοστικό ταξινομητή για εκπαίδευση. Κατά τη δεύτερη αναγνώριση χρησιμοποιείται ο προσαρμοστικός ταξινομητής και οι λέξεις που δεν είχαν αναγνωριστεί προηγουμένως από τον στατικό ταξινομητή θα μπορούσαν τώρα να αναγνωριστούν.



Διαδικασία αναγνώρισης κειμένου οθόνης

1.3 Γιατί android

android

Λογότυπο του λογισμικού Android

Το smartphone είναι ένα κινητό τηλέφωνο βασισμένο σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερο προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα συμβατικό κινητό τηλέφων. Τα λειτουργικά συστήματα (OS) των κινητών τηλεφώνων που χρησιμοποιούνται από τα σύγχρονα έξυπνα τηλέφωνα περιλαμβάνουν, μεταξύ άλλων, το Android της Google, το iOS της Apple, το Symbian της Nokia, το BlackBerry OS της RIM, το Bada της Samsung, τα Windows Phone της Microsoft.

Το Android τρέχει τον πυρήνα του λειτουργικού Linux. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear).

Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X μαζί, για αυτό και επιλέχτηκε για την δημιουργία της εφαρμογής, με αποτέλεσμα να είναι διαθέσιμο σε όσο το δυνατόν περισσότερους χρήστες. Επίσης κατά την επιλογή πλατφόρμας για την οποία θα δημιουργούνταν η εφαρμογή ήταν απαραίτητο να είναι φορητή συσκευή καθώς ο χρήστης θα έπρεπε να την έχει συνέχεια μαζί σου για να είναι πιο πρακτικό για τον χρήστη να την χρησιμοποιεί συνέχεια για να αποθηκεύει τις ιατρικές του μετρήσεις και να έχει πρόσβαση σε αυτές. Επίσης επιλέγοντας μας δίνεται η δυνατότητα να διαθέσουμε την εφαρμογή τόσο στο Play Store όσο και στην μορφή apk σε διαφορά αλλά sites (github, ApkPure), κάτι που δεν θα ήταν τόσο εύκολο αν επειλεγαμε iOS.

Διάγραμμα μεριδίου αγοράς κινητών



1.4 Γιατι OpenCV



Λογότυπο της βιβλιοθήκης OpenCV

Επεξεργασία εικόνας (Image Processing). Είναι η κάθε είδους επεξεργασία σήματος στην οποία είσοδος είναι μία εικόνα όπως μία φωτογραφία ή ένα καρέ από ένα βίντεο και η έξοδος μπορεί να είναι είτε μία εικόνα είτε ένα σύνολο χαρακτηριστικών και παραμέτρων που σχετίζονται με εικόνα. Οι περισσότερες τεχνικές επεξεργασίας περιλαμβάνουν τον χειρισμό της εικόνας σαν να είναι ένα σήμα δύο διαστάσεων και στη συνέχεια απλά εφαρμόζουν απλές τεχνικές επεξεργασίας σήματος πάνω σε αυτό το ζεύγος σημάτων. Η επεξεργασία εικόνας συνήθως αναφέρεται στην ψηφιακή επεξεργασία εικόνας αλλά και η αναλογική επεξεργασία είναι επίσης εφικτή από αυτή. Παρακάτω γίνεται αναφορά στις γενικές τεχνικές οι οποίες έχουν εφαρμογή και στα δύο είδη (ψηφιακή και αναλογική επεξεργασία) .

Η επεξεργασία εικόνας κρυνεται απαραίτητη για την ακριβή λειτουργία της εφαρμογής, καθώς η εφαρμογή βασίζεται σε φωτογραφίες τις οποίες έχει τραβήξει ο χρήστης, με αποτέλεσμα να υπάρχουν αρκετές μεταβλητές (φωτεινότητα, γωνία λήψης, “θορυβός”) που μπορούν να επηρεάσουν το αποτέλεσμα, για τις οποίες δεν είναι υπεύθυνος ο χρήστης αλλά η εφαρμογή μέσω της επεξεργασίας εικόνας, όπου εξαλφονται οι μεταβλητές αυτές , έτσι ώστε να παραχτεί ένα αριστο αποτέλεσμα.

Για την επεξεργασία εικόνας σε πειβαλλον android υπάρχουν οι διαφορες βιβλιοθυκες οπως OpenCV, OpenIMAJ, JIJIL, JHLABS, LIB-MAGICK, IMAGEJ κ.α. Το OpenCV επιλεχτηκε καθώς παρέχετε σε πολλές γλώσσες (C++, Java, Python, MATLAB) και λειτουργικά συστήματα (Linux, Windows, Android, iOS) έτσι αν χρειαστεί να μεταφερθεί η εφαρμογή σε άλλη πλατφόρμα να είναι ευκολή η μεταβαση. Η βιβλιοθήκη περιέχει πάνω από 2500 αλγορίθμους επεξεργασίας και είναι μακρυν καλύτεροι και αποτελεσματικότεροι από τις υπολοιπες εναλακτικες, οπως διαπιστοθηκε μετα απο δοκιμες.

OpenCV (Open Computer Vision) είναι μία υψηλού επιπέδου,βιβλιοθήκη τεχνητής όρασης, η οποία δημιουργήθηκε απο την Intel το 1999. Μετά από κάποιες beta εκδόσεις που προϋπήρξαν στο παρελθόν ,η πρώτη έκδοση της OpenCV κυκλοφόρησε το 2006 (10). Η βιβλιοθήκη αυτή είναι ανοιχτού κώδικα και είναι ελεύθερη για εκπαιδευτική αλλά και για εμπορική χρήση .Είναι γραμμένη κυρίως στην γλώσσα προγραμματισμού C ενω περιέχει διεπαφές για C++, Python, Java και MATLAB και μπορεί να χρησιμοποιηθεί σε λειτουργικά Linux, Windows, Mac Os κλπ . Η OpenCV έχει ενσωματωμένους αλγορίθμους επεξεργασίας εικόνας και τεχνητής όρασης.Η χρήση της χαρακτηρίζεται από υψηλή ταχύτητα, ευκολία στη χρήση και αποδοτικότητα γεγονός που εξηγεί την ευρεία χρήση της σε ακαδημαϊκό αλλά και εμπορικό επίπεδο.

Η βιβλιοθήκη OpenCV οργανώνεται σε ενότητες. Κάθε συνάρτηση της βιβλιοθήκης ανήκει σε μία συγκεκριμένη ενότητα. Οι ενότητες που χρησιμοποιούνται στην παρούσα διπλωματική εργασία είναι οι εξής.

Core Functionality. Η ενότητα core περιλαμβάνει όλους τους βασικούς τύπους δεδομένων οι οποίοι αποτελούνται από απλά διανύσματα και πίνακες, παράσταση απλών γεωμετρικών δομών όπως σημεία, ορθογώνια κ.λ.π., καθώς και μεγάλες παρατάξεις δεδομένων.

Image Processing. Η ενότητα imgproc περιέχει ένα πλήθος συναρτήσεων και κλάσεων σχετικά με την επεξεργασία εικόνας.Οι συναρτήσεις και οι κλάσεις αυτές αφορούν διαδικασίες φιλτραρίσματος, γεωμετρικούς και ποικίλους μετασχηματισμούς εικόνας, ιστογράμματα, δομική ανάλυση και περιγραφή σχήματος. Οι συναρτήσεις και οι κλάσεις που χρησιμοποιήθηκαν στην παρούσα εργασία αφορούν τις διαδικασίες φιλτραρίσματος, την ανίχνευση χαρακτηριστικών σημείων.

1.5 Μηχανική μάθηση

Μηχανική μάθηση είναι υποπεδίο της επιστήμης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Το 1959, ο Άρθουρ Σάμουελ ορίζει τη μηχανική μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μάθαινουν, χωρίς να έχουν ρητά προγραμματιστεί". Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μάθαινουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα.

Η μηχανική μάθηση είναι στενά συνδεδεμένη και συχνά συγχέεται με υπολογιστική στατιστική, ένας κλάδος, που επίσης επικεντρώνεται στην πρόβλεψη μέσω της χρήσης των υπολογιστών.

Έχει ισχυρούς δεσμούς με την μαθηματική βελτιστοποίηση, η οποία παρέχει μεθόδους, τη θεωρία και τομείς εφαρμογής. Η Μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ανέφικτος. Παραδείγματα εφαρμογών αποτελούν τα φίλτρα spam (spam filtering), η οπτική αναγνώριση χαρακτήρων (OCR), οι μηχανές αναζήτησης και η υπολογιστική όραση. Η Μηχανική μάθηση μερικές φορές συγχέεται με την εξόρυξη δεδομένων, όπου η τελευταία επικεντρώνεται περισσότερο στην εξερευνητική ανάλυση των δεδομένων, γνωστή και ως μη επιτηρούμενη μάθηση.

Κεφάλαιο 2

Τεχνολογίες που χρησιμοποιήθηκαν

2.1 PHP



Λογότυπο της γλώσσας php

Το PHP ή αλλιώς Hypertext Preprocessor είναι μια πολύ διαδεδομένη γλώσσα λογισμικού, που είναι κατάλληλη για ανάπτυξη ιστοσελίδων και μπορεί να ενσωματωθεί σε HTML.

Η PHP χρησιμοποιεί μια μίξη από interpretation (διερμίνευση) και compilation (μεταγλώττιση) και με αυτόν τον τρόπο παρέχει στους προγραμματιστές τον καλύτερο συνδυασμό απόδοσης και ευελιξίας. Δηλαδή η PHP αρχικά μεταγλωττίζει το script σε μια σειρά από εντολές, οι οποίες εκτελούνται μια μία μέχρι να τελειώσει το script. Αυτό συμβαίνει μόνο στην PHP και όχι στις υπόλοιπες παραδοσιακές γλώσσες προγραμματισμού.

Ο κώδικας PHP περικλείεται σε οδηγίες έναρξης και λήξης `<? php και?>` που μας επιτρέπουν να μεταβούμε μέσα και έξω από την λειτουργία του PHP. Αυτό που διακρίνει την PHP από κάτι σαν το JavaScript από την πλευρά του χρήστη, είναι ότι ο κώδικας εκτελείται στον server, και στην συνέχεια αποστέλλεται στον client (“πελάτη”). Ο client, θα λάβει τα αποτελέσματα της εκτέλεσης του κώδικα, αλλά δεν θα ξέρει τι είναι ο υποκείμενος κώδικας. Το πιο αναγνωρισμένο σημείο της PHP, είναι το ότι αποτελεί μια εξαιρετικά απλή γλώσσα για κάποιον νέο, παράλληλα όμως προσφέρει πολλά προηγμένα χαρακτηριστικά για έναν επαγγελματία προγραμματιστή. Παρόλο που η PHP επικεντρώνεται στην δημιουργία server όπως στην περίπτωση μας, μπορούμε να την χρησιμοποιήσουμε και σε πολλές άλλες εφαρμογές .

2.2 Visual Studio Code



Λογότυπο του λογισμικού Visual Studio Code

Για τη συγγραφή κώδικα σε php χρησιμοποιήθηκε το περιβάλλον που παρέχει το visual studio code . Το Visual Studio Code είναι ένας επεξεργαστής πηγαίου κώδικα που αναπτύχθηκε από τη Microsoft για Windows, Linux και macOS.

Περιλαμβάνει υποστήριξη για εντοπισμό σφαλμάτων, ενσωματωμένο έλεγχο Git, επισήμανση σύνταξης, έξυπνη ολοκλήρωση κώδικα, αποσπάσματα και refactoring κώδικα. Είναι επίσης προσαρμόσιμη, ώστε οι χρήστες να μπορούν να αλλάζουν το θέμα του συντάκτη, τις συντομεύσεις του πληκτρολογίου και τις προτιμήσεις. Το Visual StudioCode βασίζεται στο Electron, ένα πλαίσιο που χρησιμοποιείται για την ανάπτυξη εφαρμογών Node.js για την επιφάνεια εργασίας που εκτελείται στη μηχανή εμφάνισης Blink. Παρόλο που χρησιμοποιεί το ηλεκτρονικό πλαίσιο, το λογισμικό δεν χρησιμοποιεί το Atom και χρησιμοποιεί την ίδια συνιστώσα επεξεργαστή (με την κωδική ονομασία "Monaco") που χρησιμοποιείται στα Visual Studio Team Services (πρώην Visual Studio Online). Στην Έρευνα Προγραμματιστών Stack Overflow 2018, ο κώδικας Visual Studio κατατάχθηκε ως το πιο δημοφιλέστερο εργαλείο περιβάλλοντος για προγραμματιστές, με το 34,9% των 75.398 ερωτηθέντων να ισχυρίζονται ότι το χρησιμοποιούν. Ο κώδικας Visual Studio συνδυάζει την απλότητα ενός επεξεργαστή πηγαίου κώδικα με ισχυρά εργαλεία ανάπτυξης, όπως η ολοκλήρωση και η αποσφαλμάτωση κώδικα IntelliSense.

Είναι δωρεάν και ανοικτού κώδικα, αν και η επίσημη λήψη είναι υπό ιδιωτική άδεια. Ένας εύχρηστο editor, με φιλικό περιβάλλον στον οποίο η δημιουργία κώδικα καθώς και η επεξεργασία, και αποδιοργάνωση εντοπισμού σφαλμάτων γίνονται με ευκολία .

2.3 Android Studio.



Λογότυπο του λογισμικού Android studio

Το Android Studio παρέχει τα γρηγορότερο εργαλείο για τη δημιουργία εφαρμογών σε κάθε τύπο συσκευής Android, είναι η επίσημη έκδοση περιβάλλοντος δημιουργίας (IDE) Integrated Development Environment για το Android λογισμικό της Google, βασισμένο στο JetBrains' IntelliJ IDEA software και σχεδιασμένο για να εξυπηρετεί τους Android Developers.

Το Android Studio, λειτουργεί όπως ακριβώς και ο προκάτοχος του (το Eclipse σε συνδυασμό με το ADT) , ωστόσο έχει να προσφέρει πολλές περισσότερες δυνατότητες στον χρήστη οι οποίες καθιστούν την δημιουργία μιας mobile εφαρμογής ευκολότερη και γρηγορότερη. Για το λόγο αυτό η Google εστίασε 100% στην ανάπτυξη και τη βελτίωση του Android Studio μετά από ένα σημείο, προτείνοντας το πλέον ως το καταλληλότερο IDE για ανάπτυξη εφαρμογών σε Android. Κάποιες από αυτές τις δυνατότητες είναι:

α) η πολύ καλή πρόβλεψη του συστήματος για το τι επιθυμεί ο χρήστης να γράψει κατά την πληκτρολόγηση του κώδικα που ειδικά σε ότι αφορά το κομμάτι του Android ξεπερνάει κάθε άλλη προσπάθεια που έχει γίνει στο παρελθόν.

β) η ύπαρξη ενός χειροκίνητου συστήματος με το οποίο ο χρήστης μπορεί να κατασκευάσει εξ ολοκλήρου το γραφικό κομμάτι της εφαρμογής μόνο κάνοντας “drag and drop” τα αντικείμενα που επιθυμεί να εισάγει, χωρίς να χρειαστεί να γράψει ούτε τον ελάχιστο κώδικα.

γ) παραμετροποιήσιμα, έτοιμα αντικείμενα για χρήση από τον developer όπως activities, menus, taskbars, views, layouts, buttons κτλ. και πολλά άλλα

Η γλώσσα προγραμματισμού που χρησιμοποιεί είναι η Java και μπορεί να εγκατασταθεί στα λειτουργικά συστήματα Windows, macOS και Linux. Είναι ένα σχετικά απαιτητικό πρόγραμμα, αλλά με πάρα πολλά εργαλεία τα οποία προσφέρουν ευελιξία και αμέτρητες εφαρμογές.

2.4 MySQL



Λογότυπο του συστήματος Mysql

Το MySQL είναι ένα δωρεάν σύστημα διαχείρισης και σχεδίασης βάσης δεδομένων που χρησιμοποιεί Structure Query Language (SQL). Η SQL είναι η πιο διάσημη γλώσσα για την πρόσβαση, την προσθήκη και την διαχείριση περιεχομένου σε μια βάση δεδομένων. Είναι αρκετά γνώστη για την ταχύτητα που προσφέρει, την ευελιξία της και την αξιοπιστία της. Αποτελεί προϊόν ελεύθερου λογισμικού, μπορεί να χρησιμοποιηθεί ερασιτεχνικά η επαγγελματικά ενώ αντίστοιχα με αυτήν υπάρχουν και άλλα αντίστοιχα συστήματα, στα πιο γνωστά από αυτά εντάσσεται ο SQL server της Microsoft, ο Oracle SQL server και η DB2. Η MySQL αποτελεί σημαντικό κομμάτι σχεδόν κάθε εφαρμογής PHP. Ο κώδικας της έχει βασιστεί στις γλώσσες προγραμματισμού C και C++ ενώ η MySQL είναι συμβατή πλέον με οποιοδήποτε λειτουργικό σύστημα. Ανάλογα με τον τύπο δεδομένων που θέλει κανείς να αποθηκεύσει αλλά και τις απαιτήσεις που έχει, χρησιμοποιεί και το αντίστοιχο διαχειριστικό σύστημα το οποίο μπορεί να παρέχεται είτε δωρεάν είτε επί πληρωμή. Στην ουσία, μια βάση δεδομένων σε MySQL είναι ένας χώρος ο οποίος είναι κατάλληλα διαμορφωμένος ώστε να αποθηκευτούν δεδομένα που χρειάζεται μια εφαρμογή για τη λειτουργία της.

Η MySQL είναι μια κλασική σχεσιακή βάση δεδομένων. Αυτό σημαίνει ότι η χρήση της βασίζεται στη σωστή αναγνώριση οντοτήτων (των αντικειμένων για τα οποία θέλουμε να αποθηκεύσουμε δεδομένα σε πίνακες) και συσχετίσεων (τον τρόπο με τον οποίο οι πίνακες σχετίζονται μεταξύ τους). Όπως είναι προφανές, η MySQL βασίζεται στην γλώσσα βάσεων δεδομένων SQL, αν και δεν υλοποιεί πλήρως το πρότυπο της. Μας παρέχει πάντως τα εργαλεία για να δημιουργήσουμε πίνακες, κλειδιά (primary / foreign keys), ευρετήρια (indexes) και φυσικά να γράψουμε αναζητήσεις.

2.5 REST

{REST:API}

Ένα REST (REpresentational State Transfer) API είναι ένας από τους πιο δημοφιλείς τρόπους για την ανάπτυξη ενός Web API σήμερα, χρησιμοποιεί τους μεθόδους αιτήματος/απόκρισης του διαδικτύου όπως τα GET και POST. Το REST API είναι απλό, ελαφρύ και γρήγορο καθώς οι πόροι κάθε REST API προσδιορίζονται απλά μέσω URL's, έτσι ένας πελάτης (client) το μόνο που χρειάζεται να γνωρίζει για να χρησιμοποιήσει ένα Rest API και να έχει πρόσβαση στους πόρους του είναι τα διαθέσιμα URL's του.

Ένα REST API είναι κώδικας που υπάρχει και εκτελείται σε κάποιο server στο οποίο χρησιμοποιεί HTTP ερωτήματα (GET, POST, PUT, DELETE) για να τροποποιεί τα δεδομένα του.

Το REST χρησιμοποιεί μεθόδους ώστε να κάνει ένα HTTP αίτημα. Οι μέθοδοι αυτοί είναι:

- Η μέθοδος POST χρησιμοποιείται για την δημιουργία ενός πόρου στον server
- Η μέθοδος GET χρησιμοποιείται για την ανάσυρση / ανάκτηση ενός πόρου
- Η μέθοδος PUT χρησιμοποιείται για την αλλαγή της κατάστασης ενός πόρου ή την ενημέρωσή του
- Η μέθοδος DELETE χρησιμοποιείται για την διαγραφή ή απομάκρυνση ενός πόρου

Κάθε μέθοδος προσδιορίζεται από ένα μοναδικό url (Uniform resource identifier), κατά συνέπεια όταν ο εκτελεστής λαμβάνει ένα αίτημα ξέρει αμέσως ποια λειτουργία να εκτελέσει. Ένα url εκφράζει ένα αντικείμενο στο οποίο παρέχει πρόσβαση η υπηρεσία μέσω ενός HTTP αιτήματος. Αναλόγως με τον τύπο του αιτήματος καθορίζετε η ενέργεια που θέλουμε να εφαρμόσουμε στο αντικείμενο αυτό και το περιεχόμενο του αιτήματος περιέχει διάφορες εξειδικεύσεις τις ενέργειας

2.6 Json



Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο να διαβαστεί και να γραφτεί από τον άνθρωπο, και είναι εύκολο για τις μηχανές να το αναλύσουν και να το παράγουν. Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript.

Το JSON όπως και το XML δεν είναι γλώσσες προγραμματισμού αλλά μια σύνταξη για αποθήκευση και ανταλλαγή δεδομένων. Όπως αναφέρθηκε το κύριο πλεονέκτημα του JSON format έναντι του XML είναι ότι ο ογκος που χρειάζεται για να περιγράψει τα δεδομένα είναι σημαντικά μικρότερος.

Το JSON είναι ένα πρότυπο κειμένου το οποίο χρησιμοποιεί πρακτικές οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Η βασική κωδικοποίηση που χρησιμοποιείται είναι τα χαρακτηριστικά και οι τιμές των αντικειμένων να βρίσκονται μέσα σε εισαγωγικά και διαχωρισμένα μεταξύ τους με άνω-κάτω τελεία, ενώ κάθε επόμενο ζεύγος χαρακτηριστικού-τιμής διαδέχεται το προηγούμενο έπειτα από κόμμα. Το JSON αντικείμενο βρίσκεται εξ'ολοκλήρου μέσα σε αγκύλες ({"key1": "value1", "key2": "value2"}).

Κεφάλαιο 3

Σε αυτό το κεφάλαιο θα αναλύσουμε τις εντολές και συναρτήσεις των βασικών βιβλιοθηκών, τις οποίες θα χρησιμοποιήσουμε στην συνέχεια. Θα εξετάσουμε τον τρόπο με τον οποίο δουλεύουν, την χρησιμότητα τους, τον λόγο που τις επιλέξαμε αλλά και τις παραμέτρους που δέχεται η κάθε μια εντολή και την σημασία τους.

3.1 Βασικές λειτουργίες OpenCV

- **Mat:** Η κλάση παράταξης δεδομένων δημιουργεί n-διάστατους πυκνούς πίνακες οι οποίοι μπορούν να χρησιμοποιηθούν για αποθήκευση διανυσμάτων, εικόνων, νέφος σημείων και ιστογραμμάτων. Για τη δημιουργία στιγμιοτύπου της κλάσης παράταξης δεδομένων χρησιμοποιείται η έκφραση:

`Mat destImage = new Mat(rows, columns, type)` όπου: `destImage` το όνομα του αντικειμένου, `rows` και `columns` τα ορίσματα του πλήθους των γραμμών και στηλών και `type` η μορφή που θα έχουν τα δεδομένα.

Ομαλοποίηση εικόνων

- **cvtColor:** Η συνάρτηση μετατρέπει μια εικόνα εισόδου από έναν χρωματικό χώρο σε έναν άλλο. Σε περίπτωση μετασχηματισμού σε χρωματικό χώρο από-RGB, η σειρά των καναλιών πρέπει να προσδιορίζεται ρητά (RGB ή BGR).

`Imgproc.cvtColor(src, dst, code)` όπου:

`src`: Είναι η εικόνα που εισάγεται

`dst`: Η εικόνα που εξάγεται ίδιου μεγέθους και βάθους

`code`: Ο κωδικός μετατροπής χώρου χρώματος



Παράδειγμα χρήσης της εντολής `cvtColor`

•Gaussian Blurring: Η χρήση φίλτρων για εξομάλυνση της εικόνας έτσι ώστε να επιτυγχάνεται ελάττωση θορύβου, να απομακρύνονται οι ακραίες τιμές στο φάσμα των χρωμάτων και να γίνεται η περαιτέρω επεξεργασία ευκολότερη. Το φίλτρο που θα χρησιμοποιηθεί στηρίζεται στη θεωρία του Gauss για τον λευκό θόρυβο. Το Gaussian Filter πρόκειται για ένα χαμηλής διέλευσης φίλτρο (low-pass). Από μαθηματική άποψη το φιλτράρισμα είναι η συνέλιξη της εικόνας με τη δισδιάστατη γκαουσιανή συνάρτηση.

cv2.GaussianBlur(src, dst, (width,height), sd)

src: Η εικόνα προς επεξεργασία

dst: Η εικόνα με το τελικό αποτέλεσμα

width/height: οι διαστάσεις του πίνακα

Sd: η τυπική απόκλιση



Η αρχική και η φιλτραρισμένη με Gaussian Blurring εικόνα

•Image Thresholding: Για να διαχωρίσουμε την εικόνα σε άσπρο και μαύρο θα χρησιμοποιηθεί το Image Thresholding μετατρέπει μια εικόνα στη κλίμακα του γκρι σε μια εικόνα που αποτελείται από μόνο δύο χρώματα που ορίζονται από το χρήστη. Το ποιο από τα δύο χρώματα θα επιλεγεί για το κάθε pixel έχει να κάνει με το όριο (threshold) που επίσης ορίζει ο χρήστης. Αν η ένταση του pixel είναι πάνω από το όριο τότε επιλέγεται το ένα χρώμα ενώ εάν είναι χαμηλότερη το άλλο. Ο διαχωρισμός σε άσπρο και μαύρο γίνεται για να είναι ξεκάθαρα τα τμήματα της εικόνας που μας ενδιαφέρουν.

`cv.threshold(src, dst, thresh, maxval, type)`

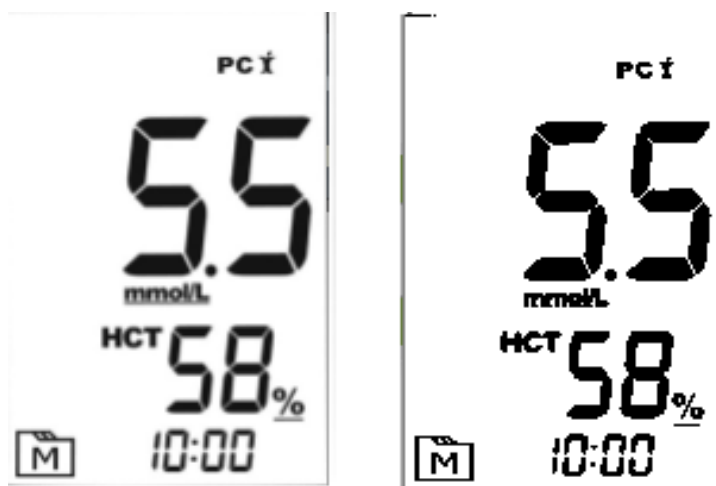
src: Η εικόνα προς επεξεργασία

dst: Η εικόνα με το τελικό αποτέλεσμα

thresh: το κατώφλι/όριο

Maxval: το χρώμα που αντικαθιστά τα pixel που υπερβαίνουν το κατώφλι

Type: το είδος του διαχωρισμού. Επιλέγουμε `cv2.THRESH_BINARY`



Το αποτέλεσμα του Image Thresholding ανάλογα με τα διάφορα Type

•Adaptive Thresholding: Στην περίπτωση που η εικόνα έχει διαφορετικές συνθήκες φωτισμού σε διαφορετικές περιοχές μια ενιαία τιμή thresholding δεν θα ήταν η πιο σωστή προσέγγιση. Για αυτό τον λόγο χρησιμοποιήσαμε το όπου ο αλγόριθμος καθορίζει το κατώτατο όριο για ένα pixel με βάση μια μικρή περιοχή γύρω από αυτό. Έχουμε λοιπόν διαφορετικά όρια για διαφορετικές περιοχές της ίδιας εικόνας που δίνουν καλύτερα αποτελέσματα για εικόνες με διαφορετικό φωτισμό.

cv.adaptiveThreshold(src, dst, maxValue, adaptiveMethod, thresholdType, blockSize)

src: Η εικόνα προς επεξεργασία

dst: Η εικόνα με το τελικό αποτέλεσμα

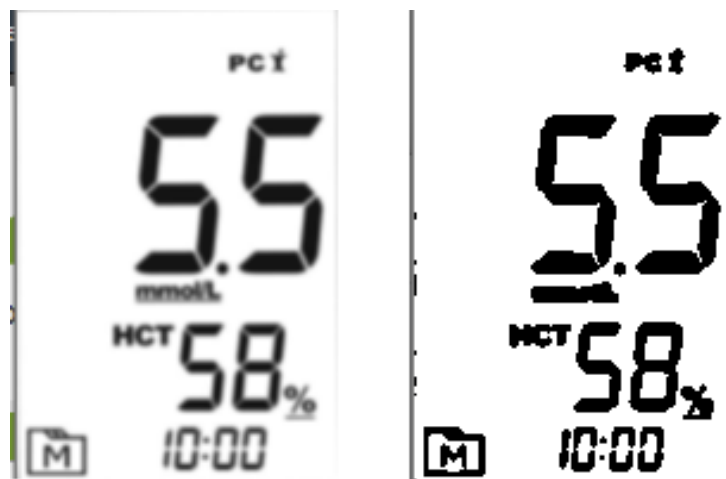
maxValue: Τα εικονοστοιχεία που περνούν το κατώτατο όριο μετατρέπονται σε λευκό

adaptiveMethod: Αλγόριθμος για τον υπολογισμό της τιμής κατωφλίου.

thresholdType: Ο τύπος του threshold που μπορεί να είναι THRESH_BINARY ή

THRESH_BINARY_INV

blockSize: Το μέγεθος του μπλοκ που θα χρησιμοποιηθεί για τον υπολογισμό του πίνακα διασποράς μιας ομάδας pixel



Η διαφορά ανάμεσα Thresholding με το Adaptive Thresholding και το Adaptive Mean Thresholding

Φιλτράρισμα εικόνων

• Dilate και erode : Οι βασικοί μορφολογικοί μετασχηματισμοί ονομάζονται dilation (διαστολή) και erosion (διάβρωση) και χρησιμοποιούνται κυρίως για αφαίρεση θορύβου, απομόνωση μεμονωμένων αντικειμένων και συνένωση ξεχωριστών αντικειμένων αντίστοιχα μέσα σε μία εικόνα

• Dilation: Αυξάνει την περιοχή του αντικειμένου. Χρησιμοποιείται για να τονίσει τα χαρακτηριστικά.

Ένας kernel (ένας περιττός πινάκας (3,5,7) είναι συνυφασμένος με την εικόνα.

Ένα εικονοστοιχείο στην αρχική εικόνα είναι «1» εάν τουλάχιστον ένα εικονοστοιχείο κάτω από τον kernel είναι «1».

Με αποτέλεσμα να αυξάνει τη λευκή περιοχή στην εικόνα ή να αυξάνει το μέγεθος του αντικειμένου.

```
void dilate(src, dst, kernel, Point anchor=Point(-1,-1), int iterations=1)
```

src: Η εικόνα προς επεξεργασία

dst: Η εικόνα με το τελικό αποτέλεσμα

kernel: Ο πίνακας kernel με προεπιλογή 3x3 διαστάσεων

anchor: Η θέση του προς αντικατάσταση pixel στον πίνακα kernel, με προεπιλογή το κέντρο του

iterations: Το πλήθος των επαναλήψεων της λειτουργίας erode/dilate.



Παράδειγμα χρήσης της εντολής Dilation

• Erode: Διαβρώνει τα όρια του αντικειμένου. Χρησιμοποιείται για τη μείωση των χαρακτηριστικών μιας εικόνας.

Ένας kernel (ένας περιττός πίνακας (3,5,7) είναι συνυφασμένος με την εικόνα.

Ένα εικονοστοιχείο στην αρχική εικόνα (είναι 1 ή 0) θα θεωρείται 1 μόνο εάν όλα τα εικονοστοιχεία κάτω από τον kernel είναι 1, αλλιώς διαβρώνεται (γίνεται μηδέν). Ετσι, όλα τα pixels κοντά στο όριο θα απορριφθούν ανάλογα με το μέγεθος του kernel. Με αποτέλεσμα το πάχος ή το μέγεθος του αντικειμένου μειώνεται δηλαδή η λευκή περιοχή στην εικόνα μειώνεται.

```
void erode(src, dst, kernel, Point anchor=Point(-1,-1), int iterations=1)
```

Οι παράμετροι είναι όμοιοι με τη λειτουργία dilate.

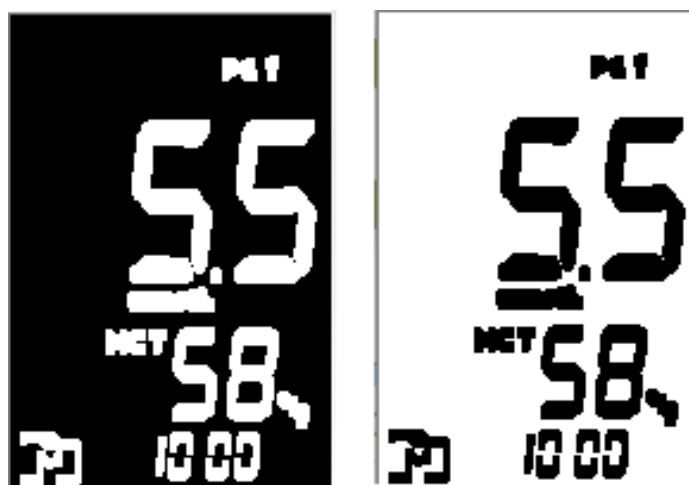


Παράδειγμα χρήσης της εντολής Erode

• bitwise_not: Συνάρτηση όπου αντιστρέφει κάθε bit ενός πίνακα. Θα χρησιμοποιηθεί για να αντιστρέψουμε το άσπρο με το μαύρο μέρος μιας εικόνας .

```
cv2.bitwise_not(src, dst)
```

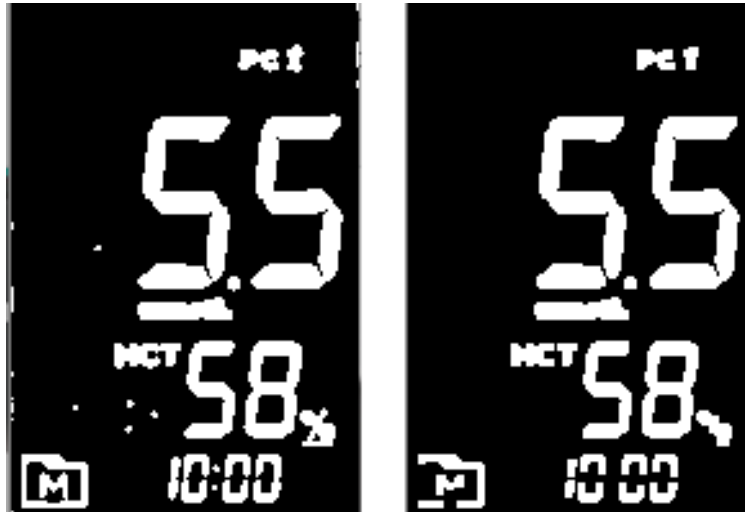
src: Η εικόνα προς επεξεργασία, dst: Η εικόνα με το τελικό αποτέλεσμα



Παράδειγμα χρήσης της εντολής bitwise_not

• Το Opening είναι απλώς ένα άλλο όνομα διάβρωσης(erosion) που ακολουθείται από διαστολή (dilation). Είναι χρήσιμο στην απομάκρυνση του θορύβου.

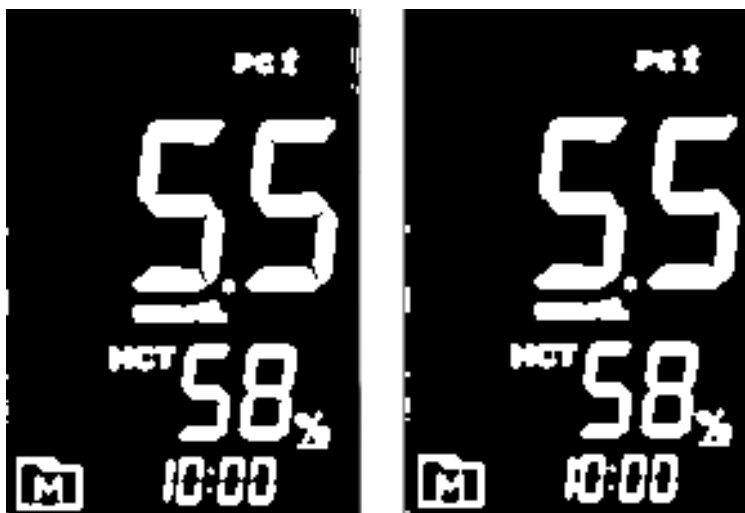
```
opening = cv.morphologyEx(img, cv.MORPH_OPEN, kernel)
```



Παράδειγμα χρήσης της εντολής Opening

• Το Closing είναι αντίστροφο του ανοίγματος, διαστολή ακολουθούμενο από διάβρωση. Είναι χρήσιμο στο κλείσιμο μικρών οπών μέσα στα αντικείμενα του προσκηνίου ή μικρών μαύρων σημείων στο αντικείμενο.

```
closing = cv.morphologyEx(img, cv.MORPH_CLOSE, kernel)
```



Παράδειγμα χρήσης της εντολής Closing

- `findContours` Η συνάρτηση `findContours` υπολογίζει περιγράμματα από δυαδικές εικόνες που δημιουργούνται από συναρτήσεις όπως η μέθοδος εύρεσης άκρων Canny και οι διάφορες μέθοδοι κατωφλίωσης.

`Imgproc.findContours(src, contours, hierarchy, Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE);`

Όπου:

`src`: Η εικόνα που εισάγεται

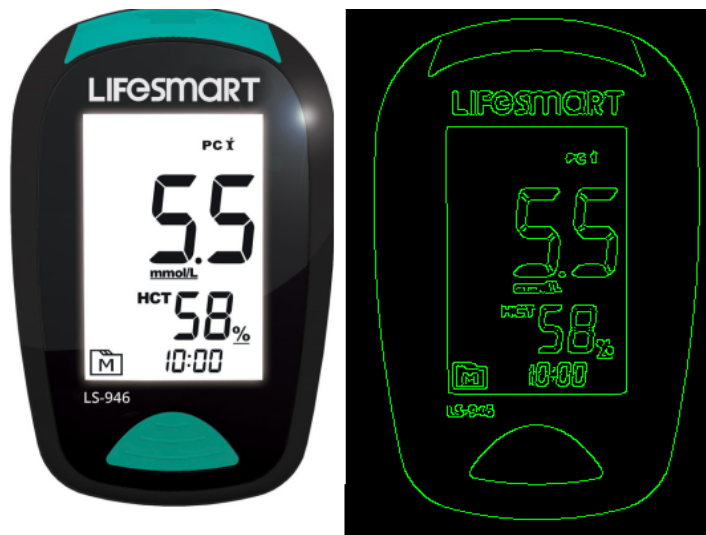
`contours`: Εντοπισμένα περιγράμματα σε μορφή διάνυσμα διανυσμάτων. Κάθε περίγραμμα

αποθηκεύεται ως διάνυσμα σημείων `Hierarchy`: Διάνυσμα που λαμβάνεται ως έξοδος και

περιέχει πληροφορίες για την τοπολογία της εικόνας. Έχει τόσα στοιχεία όσα και το πλήθος των περιγραμμάτων.

`Mode`: Λειτουργίες ανάκτησης περιγραμμάτων `CV_RETR_LIST`: Ανάκτηση όλων των περιγραμμάτων χωρίς ιεραρχική σχέση

`method` :`CHAIN_APPROX_SIMPLE`: Αφαιρεί όλα τα περιττά σημεία και συμπιέζει το περίγραμμα, εξοικονομώντας έτσι μνήμη.

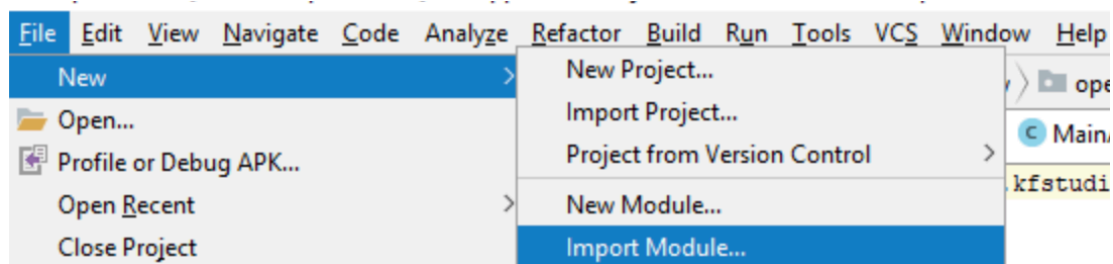


Παράδειγμα χρήσης της εντολής `findContours`

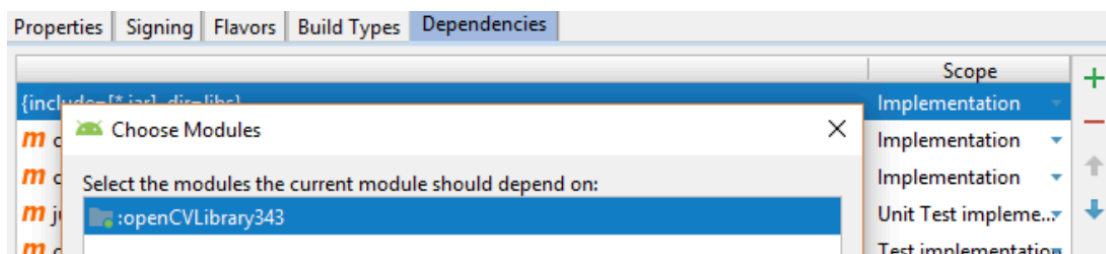
3.1.1 Εισαγωγή OpenCV στο Android Studio

Για να χρησιμοποιήσουμε το openCV στην εφαρμογή μας πρέπει πρώτα να το εισάγουμε στο android Studio.

Αρχικά θα χρησιμοποιήσουμε το Version 3.4.3 που θα κατεβάσουμε από το sourceforge. Στην συνέχεια αφού δημιουργήσουμε το project μας θα επιλέξουμε στο android Studio το import module και μετά το αρχείο που κατεβάσαμε.



Στην συνέχεια θα πατήσουμε δεξή κλικ στο project , και επιλέγουμε Open Module Settings, μετά Dependency και τέλος το '+', οπού εκεί θα επιλέξουμε το OpenCVLibrary module και θα πατήσουμε ok.



Τέλος για να ενεργοποιήσουμε το OpenCV στην εφαρμογή μας αρκεί να γράψουμε την εντολή `OpenCVLoader.initDebug();`

Αφού έχουμε εισάγει τις βιβλιοθήκες:

```
import org.opencv.android.OpenCVLoader;  
import org.opencv.android.Utils;  
import org.opencv.core.Core;
```

3.2 Βασικές λειτουργίες tesseract

3.2.1 Εύρεση των βασικών γραμμών (Finding the baselines)

Το βασικό συστατικό για την εύρεση της γραμμής βάσης είναι το φιλτράρισμα blob (Binary Long Objects). και η κατασκευή γραμμών. Blob είναι μια λέξη, ή ένα σύμβολο ή οποιοδήποτε περιεχόμενο που δεν είναι συνδεδεμένο με την υπόλοιπη εικόνα.

Το μέσο ύψος τους προσεγγίζεται, βοηθώντας την μηχανή να φιλτράρει και να αφαιρέσει τον θόρυβο που ενδεχόμενος υπάρχει.



Παράδειγμα εύρεσης δύο βασικών γραμμών

3.2.3 Προσαρμογή βάσης (Baseline fitting)

Όταν βρεθούν οι γραμμές, το Tesseract τις εξετάζει λίγο πιο προσεκτικά. Οι γραμμές βάσης ταιριάζουν με μεγαλύτερη ακρίβεια με ένα τετράγωνο, δηλαδή τέσσερις παράλληλες γραμμές. Αυτή η λειτουργία είναι πολύ χρήσιμη για να βοηθήσει το Tesseract να χειριστεί καμπυλωμένες λέξεις. π.χ. Από σάρωση βιβλίων όπου οι λέξεις είναι πιο κυρτές στο κέντρο κοντά στις βιβλιοδεσίες.

Τμηματοποίηση λέξεων (Word segmentation)

Οι λέξεις με χαρακτήρες που έχουν το ίδιο πλάτος (σταθερό βήμα) αντιμετωπίζονται ως ειδικές περίπτωση, οι λέξεις τεμαχίζονται εξίσου με βάση το μέγεθος και επιλέγονται για αναγνώριση. Ωστόσο, πιο συχνά οι χαρακτήρες των λέξεων έχουν διαφορετικά μεγέθη και πρέπει να αντιμετωπίζονται ξεχωριστά

Το Tesseract χειρίζεται τα διαφορετικά μεγέθη μετρώντας τα κενά σε ένα περιορισμένο κατακόρυφο εύρος μεταξύ της γραμμής βάσης και της μέσης γραμμής.



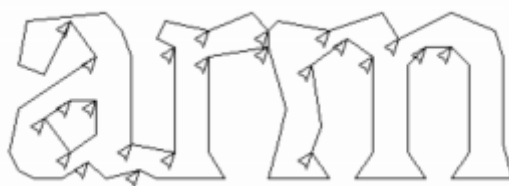
Παράδειγμα λέξης με σχεδόν σταθερό βήμα (n).

3.2.4 Τμηματοποίηση χαρακτήρων (Character segmentation)

Το Tesseract προσπαθεί να επιλύσει την τμηματοποίηση των χαρακτήρων κόβοντας το blob που είχε χαρακτηριστεί μη έμπιστο από τον ταξινομητή χαρακτήρων. Προσπαθεί να βρει σημεία υποψήφια κοπής από κοίλες κορυφές μιας πολυγωνικής προσέγγισης του περιγράμματος. Η Εικόνα δείχνει πώς το Tesseract αναγνωρίζει τον χαρακτήρα όταν ο χαρακτήρας «t» αγγίζει το χαρακτήρα «m».

Οι διαχωρισμοί εκτελούνται με σειρά προτεραιότητας . Οποιαδήποτε κοπή δεν βελτιώνει την εμπιστοσύνη του αποτελέσματος θα αναιρείτε, αλλά δεν θα απορριφθεί πλήρως. Θα αναλυθεί για άλλη μια φορά από τον “συσχετισμός σπασμένων χαρακτήρων”

Παράδειγμα λέξης προς τμηματοποίηση



3.2.5 Συσχετισμός σπασμένων χαρακτήρων

Εάν έχουν δοκιμαστεί όλα τα πιθανά σημεία κοπής και η λέξη εξακολουθεί να μην είναι αρκετά καλή, δίνεται στον associator. Ο associator δοκιμάζει διαφορετικά υποψήφια κομμάτια από μια ουρά προτεραιότητας και τις αξιολογεί με την ταξινόμηση μη ταξινομημένων συνδυασμών θραυσμάτων. Το πλεονέκτημα είναι ότι λέξεις που λείπουν απαραίτητα κομμάτια θα μπορούσαν να αναγνωριστούν όπως στο παράδειγμα του φαίνεται στο σχήμα .



Παράδειγμα λέξης σπασμένων χαρακτήρων

3.2.6 Ταξινόμηση χαρακτήρων (Character Classification)

Η ταξινόμηση χαρακτήρων χωρίζεται σε δύο περάσματα, πέρασμα 1 και πέρασμα 2.

Το πρώτο πέρασμα γίνεται μέσω του στατικού ταξινομητή και το δεύτερο μέσω του προσαρμοστικού ταξινομητή. Το πρώτο βήμα περιλαμβάνει μια διέλευση μέσω του στατικού ταξινομητή, από τα οποία εξάγονται τα περιγράμματα ο χαρακτήρας που πρέπει να αναγνωρισθεί. Αν και το περίγραμμα των χαρακτήρων μερικές φορές δεν έχει συγκεκριμένες λεπτομέρειες για να αναγνωρισθεί απευθείας η μηχανή μπορεί να χρησιμοποιήσει πολυγωνική προσέγγιση για να ταιριάζει τους σχετικούς σπασμένους χαρακτήρες μετατρέποντας τους σε αναγνωρίσιμους.

Κάθε χαρακτήρας που ταιριάζει επιτυχώς με τον στατικό ταξινομητή αποστέλλεται στον προσαρμοστικό ταξινομητή για εκπαίδευση. Αυτή η εκπαίδευση είναι απαραίτητη για το δεύτερο βήμα της διαδικασίας αναγνώρισης όπου θα χρησιμοποιήσει τις γενικές πληροφορίες που συλλέγονται από τον στατικό ταξινομητή για να τους ταιριάζει με προηγούμενους αποτυχημένους χαρακτήρες. Όταν ένας χαρακτήρας περάσει τον προσαρμοστικό ταξινομητή, είναι ασφαλές να θεωρήσουμε ότι έχει ταιριάζει.

Ο στατικός ταξινομητής χαρακτήρων μπορεί να αναγνωρίσει πλήρεις και σπασμένους χαρακτήρες

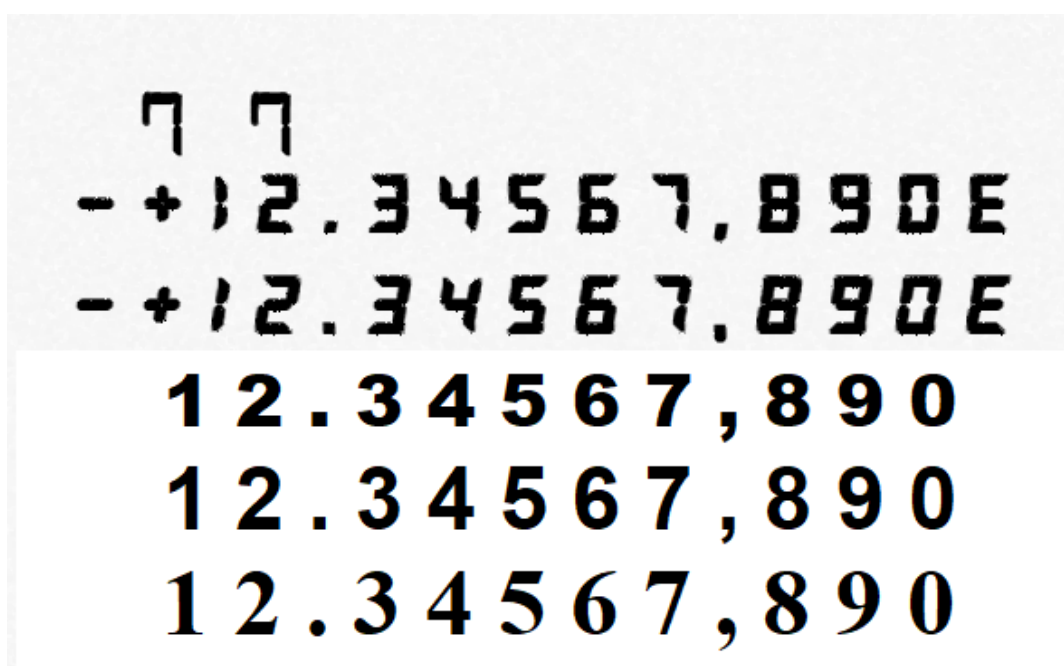
Παράδειγμα λέξης με ταξινόμηση χαρακτήρων



3.2.7 Δεδομένα Εκπαίδευσης (Training Data)

Για να βελτιώσουν την ακρίβεια και την επέκταση των δυνατοτήτων της, η μηχανή Tesseract έχει σχεδιαστεί για να είναι πλήρως εκπαιδύσιμη. Αυτό ακριβώς χρειάστηκε να γίνει στα πλαίσια αυτής της εργασίας, καθώς παρά την επεξεργασία εικόνας, σε αριθμούς της μορφής 7-segment (τμηματικούς) δεν κατέστη δυνατόν από τις υπάρχουσες γλώσσες να αναγνωρίζουν τους χαρακτήρες με συνεπεία και υψηλό ποσοστό επιτυχίας. Έτσι χρειάστηκε να εκπαιδύσουμε την δικιά μας γλωσσά (traineddata)

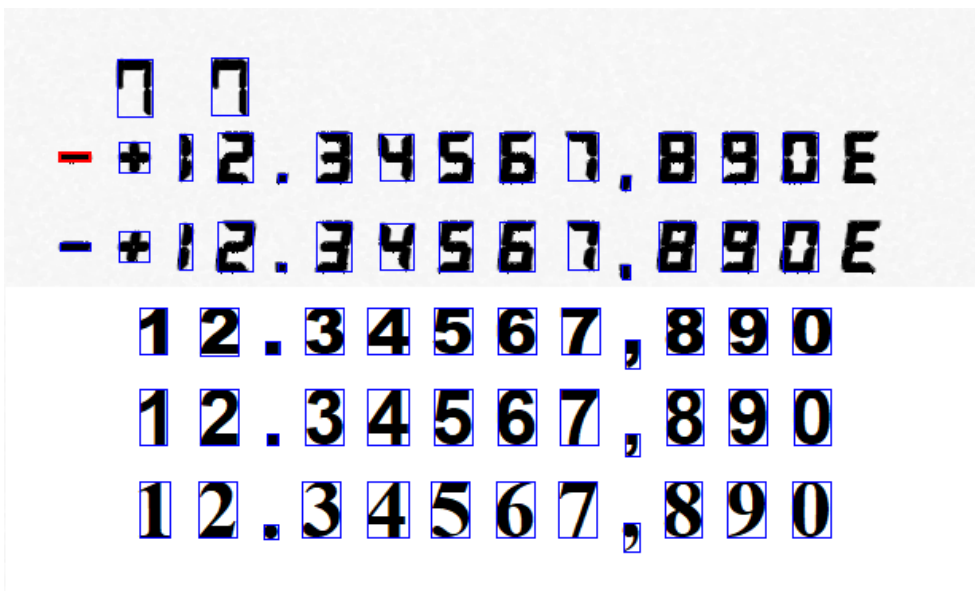
Αρχικά δημιουργήθηκε ένα .tif αρχείο εικόνας με τους διαθέσιμους χαρακτήρες για τους οποίους θέλαμε να δημιουργήσουμε την γλωσσά.



Χαρακτήρες προς εκπαίδευση

Στην συνέχεια χρησιμοποιώντας το πρόγραμμα jTessBoxEditor στο οποίο αντιστοιχούμε κάθε χαρακτήρα της προηγούμενης εικόνας με το αποτέλεσμα που θα παράγει όταν αναγνωρίζεται από τον αλγόριθμο. Έτσι δημιουργείτε ένα .box αρχείο στο οποίο υπάρχουν οι συντεταγμένες των χαρακτήρων πάνω στην εικόνα και ο χαρακτήρας στον οποίο αντιστοιχεί όταν αναγνωρίζεται από τον αλγόριθμο.

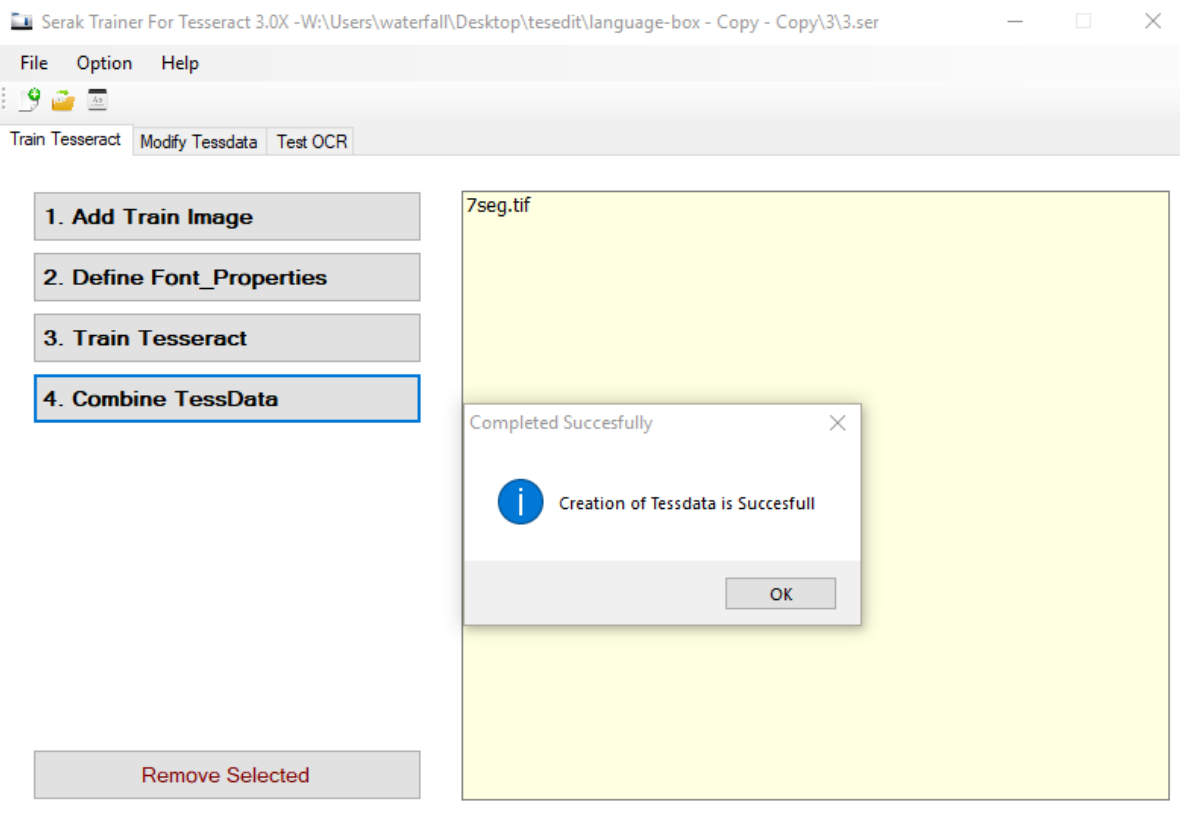
Επιλογή χαρακτήρων



Δεδομένα .box αρχείου

Box Coordinates		Box Data		Box View	
	Char	X	Y	Width	Height
1	-	103	121	19	5
2	+	140	114	19	19
3	7	139	62	22	36
4	7	198	61	23	36
5	1	178	109	8	29
6	2	202	108	22	31
7	.	240	134	5	5
8	3	264	108	22	31
9	4	304	109	21	29
10	5	341	108	21	31
11	6	379	108	22	31
12	7	422	108	19	30
13	,	456	135	5	9
14	8	479	108	23	31
15	9	519	108	22	31
16	0	557	108	22	31
17	-	103	177	19	5
18	+	140	170	19	19
19	1	178	165	8	29
20	2	202	164	22	31
21	.	240	190	5	5
22	3	264	164	22	31
23	4	304	165	21	29
24	5	341	164	21	31
25	6	379	164	22	31
26	7	422	164	19	30
27	,	456	191	5	9
28	8	479	164	23	31
29	9	519	164	22	31
30	0	557	164	22	31
31	1	153	218	17	29
32	2	191	218	24	30
33	.	232	239	9	8
34	3	257	218	24	29
35	4	296	218	26	29

Τέλος για να ενώσουμε την .tif εικόνα και το .box αρχείο και να παράγουμε το .traineddata που είναι η εκπαιδευμένη γλώσσα (λεξικό) που θα εισάγουμε στην μηχανή tesseract για να αναγνωρίζει τους χαρακτηρίστηκες, μπορούμε να το κάνουμε μεσώ γραμμών εντολών από το cmd η να χρησιμοποιήσουμε ένα πρόγραμμα που λέγεται SerakTesseractTrainer στο οποίο επιλέγουμε την εικόνα (το αρχείο .box θα πρέπει να βρίσκεται στον ίδιο φάκελο με ίδιο όνομα πχ. 7Seg.tif και 7seg.box) επιλεγουμε train tesseract για να εκπαίδευση τον αλγόριθμο (δημιουργία tessdata) και αν δεν υπάρχουν errors με το combine tessdata δημιουργείτε το .traineddata αρχείο.



Ολοκλήρωση εκπαίδευσης γλώσσας

Τα αρχεία traineddata πρέπει να αποθηκευτούν στην συσκευή για να διαβαστούν από τον αλγόριθμο, κάτι όπου κάνουμε ήδη πρώτα ελέγχουμε ότι δεν τα έχουμε αποθηκεύσει είδη.

```
if(dir.exists()) {
    String datapath1 = datapath+ "/tessdata/7-num.traineddata";
    String datapath2 = datapath+ "/tessdata/7-only.traineddata";
    File datafile = new File(datapath1);
    File datafile1 = new File(datapath2);

    if (!datafile.exists() || !datafile1.exists() ) {
        copyFiles( name: "7-num");
        copyFiles( name: "7-only");
    }
}

private void copyFiles(String name) {
    try {
        String filepath = datapath + "/tessdata/"+name+".traineddata";
        AssetManager assetManager = getAssets();

        InputStream instream = assetManager.open( fileName: "tessdata/"+name+".traineddata");
        OutputStream outstream = new FileOutputStream(filepath);

        byte[] buffer = new byte[1024];
        int read;
        while ((read = instream.read(buffer)) != -1) {
            outstream.write(buffer, off: 0, read);
        }
        outstream.flush();
        outstream.close();
        instream.close();
    }
}
```

Αποθήκευση traineddata αρχείων στην συσκευή

Εισαγωγή tesseract στο android studio

Η εισαγωγή της βιβλιοθήκης tesseract στο android studio είναι αρκετά εύκολη αφού το μόνο που χρειάζεται είναι να προσθέσουμε την εντολή

implementation 'com.rmtheis:tess-two:9.0.0'

που Tess-two είναι το όνομα της εφαρμογής για το Android στο build.gradle αρχείο του project και κανοντας build την εφαρμογή θα κατεβάσει αυτόματα και θα εγκαταστήσει την βιβλιοθήκη.

Κεφάλαιο 4

4.1 Εντοπισμός οθόνης

Καθώς η ανίχνευση κείμενου σε φωτογραφίες είναι αρκετά δυσκολότερη σε σχέση με τις σκαναρισμένες σελίδες καθώς υπάρχουν αρκετά περισσότερες μεταβλητές όπως η απόσταση από το αντικείμενο, η γωνία λήψης κ.ά. Θα πρέπει να απομονώσουμε την οθόνη της εκάστοτε συσκευής με αποτέλεσμα να μειωθεί σημαντικά ο θόρυβος άλλα και να βεβαιωθούμε ότι θα υπάρχει η ίδια περιοχή ενδιαφέροντος για κάθε συσκευή ανεξάρτητα από την απόσταση στην οποία θα τραβηχτεί η φωτογραφία, κάτι που είναι απαραίτητο για να δουλέψει ο αλγόριθμος μηχανικής εκμάθησης όπως θα αναφερθούμε και στην συνέχεια !

4.2 Αλγόριθμος απομόνωσης οθόνης

Πιο συγκεκριμένα ο αλγόριθμος για την απομόνωση της οθόνης από την εικόνα είναι ο ακόλουθος:

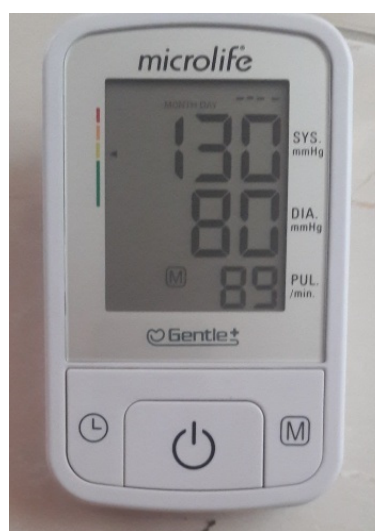
- Αρχικά μετατρέπουμε την εικόνα από bitmap σε mat, δηλαδή πίνακες που περιέχουν τα εικονοστοιχεία της εικόνας, μέσω της συνάρτησης :

```
Utils.bitmapToMat(original, src);
```

όπου original είναι η bitmap εικόνα που εισάγουμε και src ο πίνακας mat που θα επιστραφεί !

- Μετατρέπουμε την εικόνα σε αποχρώσεις του γκρι διαγράφοντας έτσι την περιττή πληροφορία χρησιμοποιώντας την συνάρτηση :

```
Imgproc.cvtColor(src, src, Imgproc.COLOR_RGBA2GRAY);
```



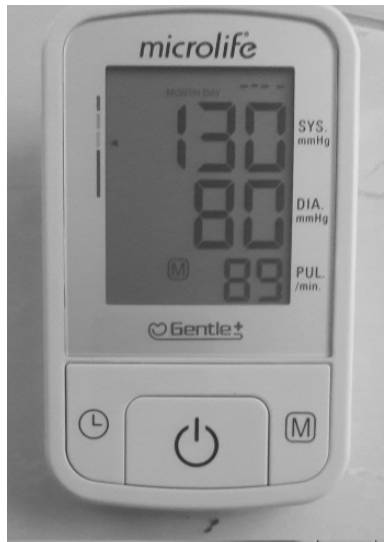
original



Gray

Χρήση εντολής cvtColor

- Μειώνουμε την ευκρίνεια εφαρμόζοντας το φίλτρο Gauss με σκοπό να μειωθεί ο περιττός θόρυβος μέσω της συνάρτησης: `Gauss I`
`imgproc.GaussianBlur(src, src, new Size(5, 5), 0);`



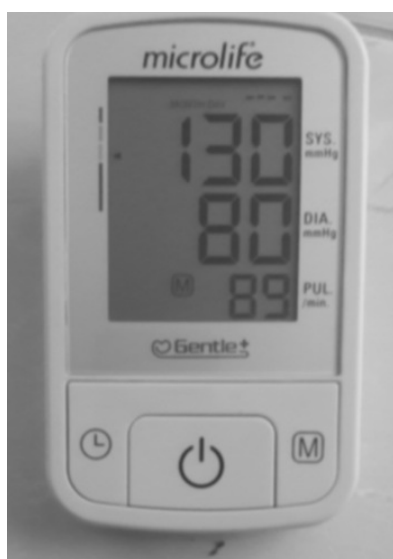
Gray



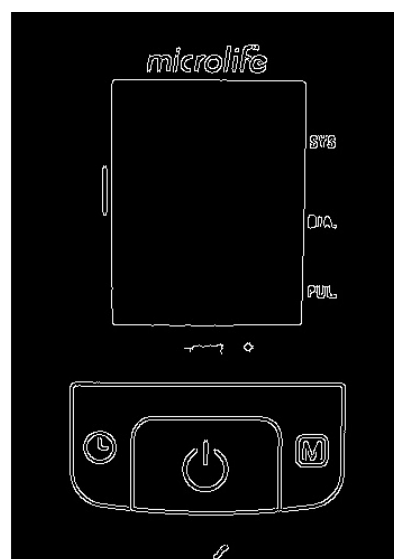
Blur

Χρήση εντολής GaussianBlur

- Για να προσδιορίσουμε το περίγραμμα της οθόνης πρέπει να εντοπίσουμε τα άκρα του τα οποία χαρακτηρίζονται από μια απότομη αλλαγή στην διαβάθμιση της φωτεινότητας. Αυτό επιτυγχάνεται χρησιμοποιώντας την μέθοδο `canny` η οποία αναγνωρίζει τα εικονοστοιχεία ακμών της εικόνας `Imgproc.Canny(src, src, 30, 200);`



Blur



Edged

Χρήση εντολής Canny

• Το επόμενο βήμα είναι η πραγματική ανίχνευση των περιγραμμάτων των αντικειμένων στην εικόνα. Για να γίνει αυτό, θα χρησιμοποιήσουμε τη συνάρτηση `cv2.findContours`: `Imgproc.findContours(src, contours, hierarchy, Imgproc.RETR_LIST, Imgproc.CHAIN_APPROX_SIMPLE)`;



Χρήση εντολής `findContours`

• Η οθόνη είναι ένα ορθογώνιο με τέσσερις κορυφές, έτσι αν ελέγξουμε τα περιγράμματα και βρούμε τέσσερις κορυφές σημαίνει ότι είναι η οθόνη ! Για καθέναν από τους βρόχους, υπολογίζουμε την περίμετρο χρησιμοποιώντας `cv2.arcLength` και, στη συνέχεια, προσεγγίζουμε (ομαλά) τον βρόχο χρησιμοποιώντας `cv2.approxPolyDP`. Όπως φαίνεται και στην εικόνα το ορθογώνιο μπορεί να μην είναι τέλειο για αυτό και το προσεγγίζουμε ομαλά. Ελέγχουμε αν το περίγραμμα έχει τέσσερις κορυφές και αν ναι απομονώνουμε αυτό το μέρος της εικόνας. Τέλος για να μειώσουμε την επιρροή που θα έχει η γωνία λήψης στο τελικό αποτέλεσμα (για παράδειγμα στην εικόνα η πάνω πλευρά της οθόνης είναι μεγαλύτερη από την κάτω ενώ στην πραγματικότητα είναι ίδιες) , χρησιμοποιούμε την συνάρτηση: `mgproc.getPerspectiveTransform(src1, dst)`.

Η οθόνη που παίρνουμε ως αποτέλεσμα είναι αυτή:



Τελική οθόνη μετά την επεξεργασία

Ενώ ο κώδικας είναι αυτός:

```
for (int c = 0; c < contours.size(); c++) {
    MatOfPoint2f approxCurve = new MatOfPoint2f();
    MatOfPoint2f contour2f = new MatOfPoint2f(contours.get(c).toArray());
    double approxDistance = Imgproc.arcLength(contour2f, true) * 0.02;
    Imgproc.approxPolyDP(contour2f, approxCurve, approxDistance, true);

    if (approxCurve.toArray().length == 4) {
        MatOfPoint points = new MatOfPoint(approxCurve.toArray());
        // Get bounding rect of contour
        List<MatOfPoint> contourTemp = new ArrayList<>();
        contourTemp.add(points);
        Rect rect = Imgproc.boundingRect(points);
        Mat destImage = new Mat(rect.height, rect.width, src.type());
        Mat src1 = new MatOfPoint2f(new Point(points.toList().get(0).x, points.toList().get(0).y),
            new Point(points.toList().get(3).x, points.toList().get(3).y), new Point(points.toList().get(2).x, points.toList().get(2).y), new Point(points.toList().get(1).x, points.toList().get(1).y));
        Mat dst = new MatOfPoint2f(new Point(0, 0), new Point(destImage.width() , 0), new Point(destImage.width() , destImage.height() ), new Point(0, destImage.height() ));
        Mat transform = Imgproc.getPerspectiveTransform(src1, dst);
        Imgproc.warpPerspective(src2, src, transform, destImage.size());
        break;
    }
}
```

Κώδικας εύρεσης οθόνης

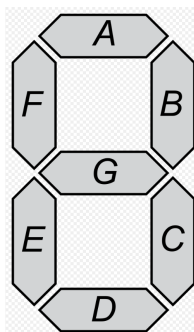
- Τέλος μετατρέπουμε την εικόνα από mat σε bitmap και την παρουσιάζουμε στον χρήστη.

```
original = Bitmap.createBitmap(src.cols(), src.rows(), Bitmap.Config.ARGB_8888);
Utils.matToBitmap(src, original);
```

Ανάθεση νέας επεξεργασμένης εικόνας ως κύρια

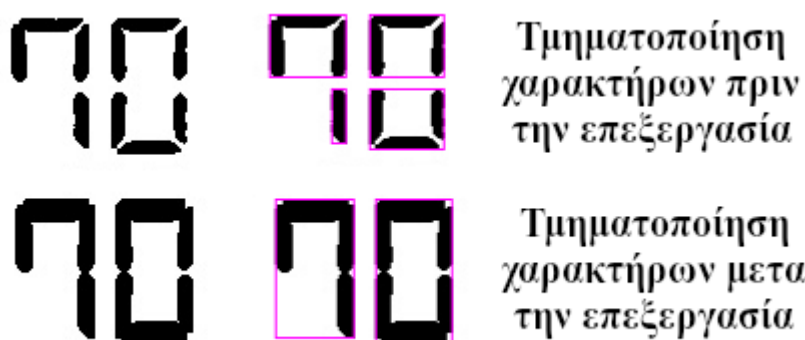
4.3 Επεξεργασία αριθμών οθόνης

Αφού έχουμε εντοπίσει την οθόνη συνεχεία έχει η κατάλληλη επεξεργασία της έτσι ώστε να διασφαλίσουμε ότι ο αλγόριθμος αναγνώρισης θα είναι ακριβής, η επεξεργασία αυτή βοηθάει ιδιαίτερα τον αλγόριθμο αφού έχει παρατηρηθεί ότι παράγει καλύτερα αποτελέσματα όταν χρησιμοποιούνται συγκεκριμένα φίλτρα (όπως για παράδειγμα μαύρα νούμερα σε λευκό φόντο από ότι το αντίθετο). Στην συγκεκριμένη εφαρμογή ωστόσο κρίνεται απαραίτητη καθώς πολλά ιατρικά μηχανήματα χρησιμοποιούν στις οθόνες τους αριθμούς της μορφής 7-segment (τμηματικούς).



Παράδειγμα 7-seg αριθμού

Οι αλγόριθμοι αναγνώρισης θεωρούν ότι οι χαρακτήρες είναι ενιαίοι με αποτέλεσμα όταν καλούνται να αναγνωρίσουν έναν τμηματικό χαρακτήρα να θεωρούν ότι είναι διάφοροι μικρότεροι χαρακτήρες, για αυτό μέσο συναρτήσεων θα καλύψουμε τα καινά των 7-seg χαρακτήρων



Διαφορά αναγνώρισης αριθμών από τον αλγόριθμο

4.3.1 Αλγόριθμος επεξεργασία αριθμών οθόνης

Όπως και στον αλγόριθμο απομόνωσης οθόνης μετατρέπουμε την εικόνα σε αποχρώσεις του γκρι και μειώνουμε την ευκρινών χρησιμοποιώντας τις συναρτήσεις:

```
Imgproc.cvtColor(src, src, Imgproc.COLOR_RGBA2GRAY); και  
Gauss Imgproc.GaussianBlur(src, src, new Size(5, 5), 0);
```



Original

Gray

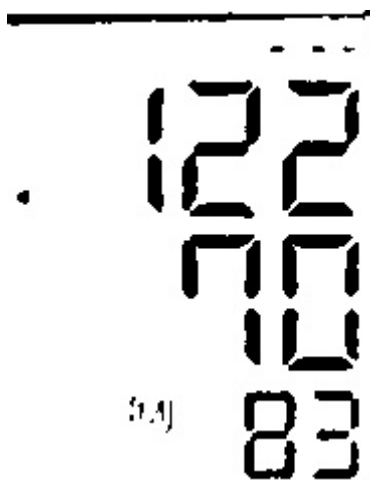
Blur

Επεξεργασία εικόνας οθόνης (COLOR_RGBA2GRAY & GaussianBlur)

Στην συνέχεια μετατρέπουμε την εικόνα σε δυαδική, που έχει μόνο μαύρα και άσπρα pixel χρησιμοποιώντας την συνάρτηση:

```
Imgproc.adaptiveThreshold(src, src, 255, Imgproc.ADAPTIVE_THRESH_GAUSSIAN_C,  
Imgproc.THRESH_BINARY, 37, 9);
```

(όπου blocksize = 37 και c = 9 βρέθηκαν να δίνουν το βέλτιστο αποτέλεσμα)



Χρήση εντολής adaptiveThreshold

Στην συνέχεια θα προσπαθήσουμε να ενώσουμε τα καινά μεταξύ των χαρακτήρων χρησιμοποιώντας την συνάρτηση erode(διαβρώσει). Το όνομα της συνάρτησης μπορεί να παραπέμπει στο αντίθετο από αυτό που προσπαθούμε να πετύχουμε, αλλά ο αλγόριθμος αυτός επεξεργάζεται συνήθως το λευκό κομμάτι της εικόνας, όποτε στην περίπτωση μας θα “διαβρώσει” το λευκό φόντο, με αποτέλεσμα να νούμερα να φαίνονται μεγαλύτερα !

```
Imgproc.erode(src, dst, kernel, new Point(-1, -1), 2);
```



Χρήση εντολής erode

Οι δυο επόμενες συναρτήσεις που θα χρησιμοποιήσουμε, θα πρέπει να επηρεάσουν τα νούμερα στην εικόνα, θα ανατρέψουμε τα χρώματα καθώς όπως αναφέραμε οι αλγοριθμοί επεξεργάζονται το λευκό κομμάτι. Η συνάρτηση για να το κάνουμε αυτό είναι Core.bitwise_not(src, dst);



Χρήση εντολής bitwise_not

Εφαρμόζουμε στην εικόνα μορφολογικό μετασχηματισμό τύπου open, όπου είναι ένας τρόπος διάβρωσης (erode) που ακολουθείται από διαστολή (dilate). Είναι χρήσιμο στην εξάλειψη του θορύβου.

```
Mat kernel1 = Mat.ones(3, 3, CvType.CV_8U);  
Imgproc.morphologyEx(src, dst, Imgproc.MORPH_OPEN, kernel1);
```



Χρήση εντολής morphologyEx

Έχοντας πλέον μια “καθαρή” από θόρυβο εικόνα εφαρμόζουμε διαστολή (dilate) όπου θα διαστέλλει τα λευκά σημεία της εικόνας (νούμερα) με αποτέλεσμα να εξαλείφουν τα καινά .

```
Mat kernel2 = Mat.ones(5, 1, CvType.CV_8U);  
Imgproc.dilate(src, dst, kernel2, new Point(-1, -1), 1);
```



Χρήση εντολής dilate

Ο αλγόριθμος αναγνώρισης χαρακτήρων σε αντίθεση με αυτόν της επεξεργασίας εικόνας θεωρεί πως οι χαρακτήρες στην εικόνα είναι μαύρου χρώματος πάνω σε πιο ανοιχτό φόντο για αυτό και θα ανατρέψουμε για ακόμα μια φορά τα χρώματα της εικόνας.

```
Core.bitwise_not(src, dst);
```



Χρήση εντολής bitwise_not

Τέλος θα προσθέσουμε ένα λευκό καινό στα όρια της οθόνης έτσι ώστε να διευκολύνουμε τον αλγόριθμο αναγνώρισης στην περίπτωση που τα νούμερα εκτείνονται μέχρι τα όρια της εικόνας.



Τελική εικόνα προς αναγνώριση

4.3.2 Αναγνώριση κειμένου

Το τελευταίο βήμα για την ολοκλήρωση της εφαρμογής είναι η αναγνώριση του κειμένου από την επεξεργασμένη εικόνα με την χρήση του μηχανισμού Tesseract. Για να επιτευχθεί το καλύτερο αποτέλεσμα χρησιμοποιείτε διαφορετικό traineddata αναλόγως με το αν οι αριθμοί είναι της μορφής 7-seg η όχι (traineddata είναι το γλωσσικό μοντέλο σύμφωνα με το οποίο αναγνωρίζεται το κείμενο). Ο χρήστης ορίζει στην εικόνα σε ποιο σημείο βρίσκεται η κάθε τιμή αναλόγως με το όργανο το οποίο χρησιμοποιεί.

```
mTess = new TessBaseAPI();
datapath = getFilesDir()+ "/tesseract/";
checkFile(new File( pathname: datapath + "tessdata/"));
mTess.setPageSegMode(TessBaseAPI.PageSegMode.PSM_SINGLE_BLOCK);
mTess.init(datapath, language: "7-num");
```

Κώδικας αναγνώρισης κειμένου

Όπως φαίνεται στον κώδικα καθορίζεται η τοποθεσία του γλωσσικού μοντέλου, καθορίζεται το PageSegMode (αντιπροσωπεύει τους πιθανούς τρόπους ανάλυσης της διάταξης της σελίδας.) σε PSM_SINGLE_BLOCK και τέλος ενεργοποιείτε μηχανή αναγνώρισης με τις παραπάνω μεταβλητές .

Στην συνέχεια αφού κάνει την αναγνώριση των τιμών έχει την δυνατότητα να προσθέσει σχόλια και να αποθηκεύσει τις μετρήσεις για μελλοντική στον online ηλεκτρονικό του φάκελο οποι υπάρχει το ιστορικό του .

4.3.3 Αλγόριθμος αυτόματης αναγνώρισης

Ο χρήστης αφού αναγνωρίσει την συσκευή που θέλει έχει την δυνατότητα να αποθηκεύσει το μοντέλο της συσκευής. Στην ουσία αποθηκεύονται οι συντεταγμένες οπου βρίσκεται η κάθε μέτρηση πάνω στην εικόνα , έτσι ώστε την επόμενη φορά που θα χρειαστεί να αναγνώριση το ίδιο μοντέλο ο αλγόριθμος να βρει αυτόματα την τοποθεσία των μετρήσεων πάνω στην εικόνα και να κάνει την αναγνώριση! Κάθε φορά που ο χρήστης ανοίγει την εφαρμογή ανανεώνεται αυτόματα η λίστα με τα αποθηκευμένα μοντέλα.

Κεφάλαιο 6

6.1 Σχεδιασμός βάσης δεδομένων

Σχεδιασμός βάσης δεδομένων

Για τις ανάγκες του project χρειάστηκε να δημιουργήσουμε μια βάση δεδομένων η οποία ονομάστηκε OCR και αποτελείται από 3 πίνακες, και χρησιμοποιείτε για την αποθήκευση όλων των δεδομένων της εφαρμογής. Οι 3 πίνακες από τους οποίους αποτελείται είναι:

Account: Περιλαμβάνει τα δεδομένα για κάθε λογαριασμό. Ποιο συγκεκριμένα το email, password (κρυπτογραφημένο), όνομα, προαιρετικά το token άμα έχει ζητήσει για την επαναφορά κωδικού και το id του λογαριασμού το οποίο χρησιμοποιείτε και σαν primary key.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	email	text latin1_swedish_ci		No				Change Drop More
<input type="checkbox"/>	3	password	text latin1_swedish_ci		No				Change Drop More
<input type="checkbox"/>	4	Name	text latin1_swedish_ci		No				Change Drop More
<input type="checkbox"/>	5	token	text latin1_swedish_ci		No				Change Drop More

↑ Check all With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext

Δομή πίνακα Account

Data: Περιλαμβάνει τις μετρήσεις που έχει αποθηκεύσει ο χρήστης και ποιο συγκεκριμένα το id της μέτρησης (primary key), το email που αντιστοιχεί την μέτρηση με τον χρήστη, την ημερομηνία, τα σχόλια, τις τιμές (αναλόγως με τι είδους μέτρηση είναι) και το είδος της μέτρησης σαν τίτλο (blood pressure, oximeter, glucose meter).

Δομή πίνακα Data

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	title	text utf8_unicode_ci		No				Change Drop More
<input type="checkbox"/>	3	date	text utf8_unicode_ci		No				Change Drop More
<input type="checkbox"/>	4	user_email	text utf8_unicode_ci		No				Change Drop More
<input type="checkbox"/>	5	comments	text utf8_unicode_ci		No				Change Drop More
<input type="checkbox"/>	6	val1	text utf8_unicode_ci		No				Change Drop More
<input type="checkbox"/>	7	val2	text utf8_unicode_ci		No				Change Drop More
<input type="checkbox"/>	8	val3	text utf8_unicode_ci		No				Change Drop More

↑ Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Coordinates: Περιλαμβάνει όλα τα μοντέλα που έχουν αποθηκευτεί και χρησιμοποιούνται για την λειτουργία του αλγορίθμου μηχανικής εκμάθησης. Ποιο συγκεκριμένα το id του μοντέλου (primary key), το όνομα του μοντέλου, ο τύπος του και οι συντεταγμένες των σημείων των τιμών πάνω στην οθόνη.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id		int(11)	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	model_name	utf8_unicode_ci	text	No				Change Drop More
<input type="checkbox"/>	3	type	utf8_unicode_ci	text	No				Change Drop More
<input type="checkbox"/>	4	x1_1		int(11)	No	None			Change Drop More
<input type="checkbox"/>	5	x2_1		int(11)	No	None			Change Drop More
<input type="checkbox"/>	6	y1_1		int(11)	No	None			Change Drop More
<input type="checkbox"/>	7	y2_1		int(11)	No	None			Change Drop More
<input type="checkbox"/>	8	x1_2		int(11)	No	None			Change Drop More
<input type="checkbox"/>	9	x2_2		int(11)	No	None			Change Drop More
<input type="checkbox"/>	10	y1_2		int(11)	No	None			Change Drop More
<input type="checkbox"/>	11	y2_2		int(11)	No	None			Change Drop More
<input type="checkbox"/>	12	x1_3		int(11)	No	None			Change Drop More
<input type="checkbox"/>	13	x2_3		int(11)	No	None			Change Drop More
<input type="checkbox"/>	14	y1_3		int(11)	No	None			Change Drop More
<input type="checkbox"/>	15	y2_3		int(11)	No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Δομή πίνακα Coordinates

Αποστολή δεδομένων από την εφαρμογή

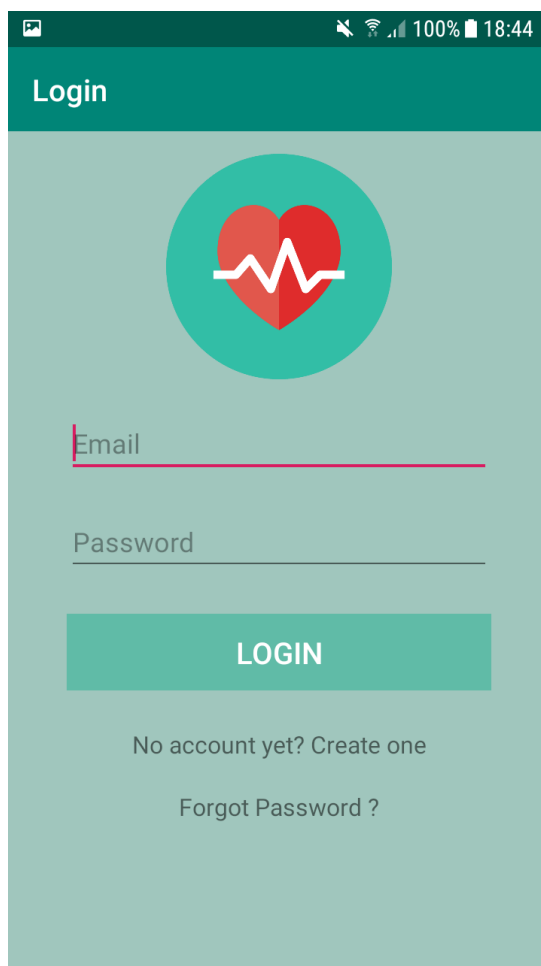
Η αποθήκευση των δεδομένων από την εφαρμογή γίνονται απευθείας στην βάση δεδομένων στον server, έτσι ώστε να μην υπάρχει ο κίνδυνος να χαθούν οι μετρήσεις άμα υπάρξει κάποιο πρόβλημα και τερματιστεί απροσδόκητα η εφαρμογή (crash), και για να μην υπάρχει ένα επιπλέον βήμα, η αποθήκευση στην συσκευή, κάτι που θα αύξανε τον χρόνο που θα χρειαζόταν η κάθε μέτρηση και την πολυπλοκότητα της εφαρμογής προς τον χρήστη.

Επίσης επειδή η εφαρμογή είναι είδη απαιτητική ως προς τους πόρους της συσκευής (επεξεργασία εικόνων, οπτική αναγνώριση, αποθηκευτικός χώρος για τις βιβλιοθήκες), και κύριος στόχος μας είναι να δουλεύει σε όσες περισσότερες συσκευές γίνεται για αυτό επιλέχτηκε να μην δημιουργείτε βάση δεδομένων στο κινητό αλλά να αποθηκεύονται τα δεδομένα προσωρινά στις μεταβλητές μέχρι να αποθηκευτούν στον μειώνοντας έτσι τις απαιτήσεις της εφαρμογής (επεξεργαστική ισχύ, χώρος αποθήκευσης).

Έτσι ο χρήστης για να αποθηκεύσει κάποια μέτρηση, το μόνο που έχει να κάνει είναι μόλις ολοκληρωθεί η μέτρηση να επιλέξει αποθήκευση και η εφαρμογή στέλνοντας ένα post request στον server με τις αναγνωρισμένες τιμές, τις οποίες αποθηκεύει στον λογαριασμό του.

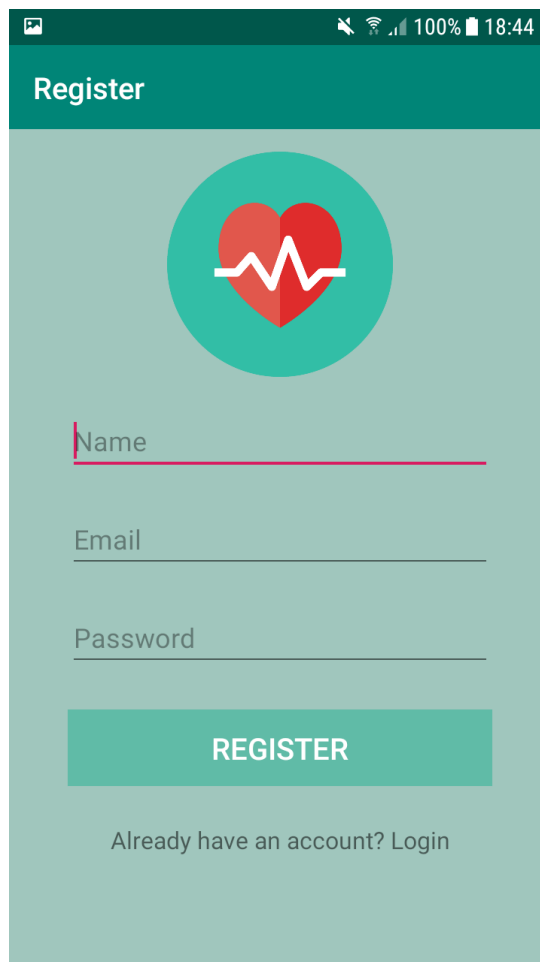
6.2 Λειτουργίες της εφαρμογής

activity_login: Η αρχική οθόνη της εφαρμογής η οποία επιτρέπει στον χρήστη να συνδεθεί χρησιμοποιώντας τα στοιχεία του, να επαναφέρει τον κωδικό του η να δημιουργήσει νέο λογαριασμό !




Οθόνη login

activity_register: Σε αυτή την οθόνη ο χρήστης μπορεί να δημιουργήσει έναν νέο λογαριασμό για να συνδεθεί στην εφαρμογή (το πεδίο email υπάρχει έτσι ώστε να μπορεί να επαναφέρει τον κωδικό του αν χρειαστεί)



Register



Name

Email

Password

REGISTER

Already have an account? [Login](#)

Οθόνη Register

forgot_pass: Ο χρήστης έχει την δυνατότητα να στείλει έναν μοναδικό κωδικό (token) στο email του. Στην συνέχεια εισάγοντας το email του και το token μπορεί να αλλάξει τον κωδικό του.

Password reset

Email

SEND TOKEN

Token

Password

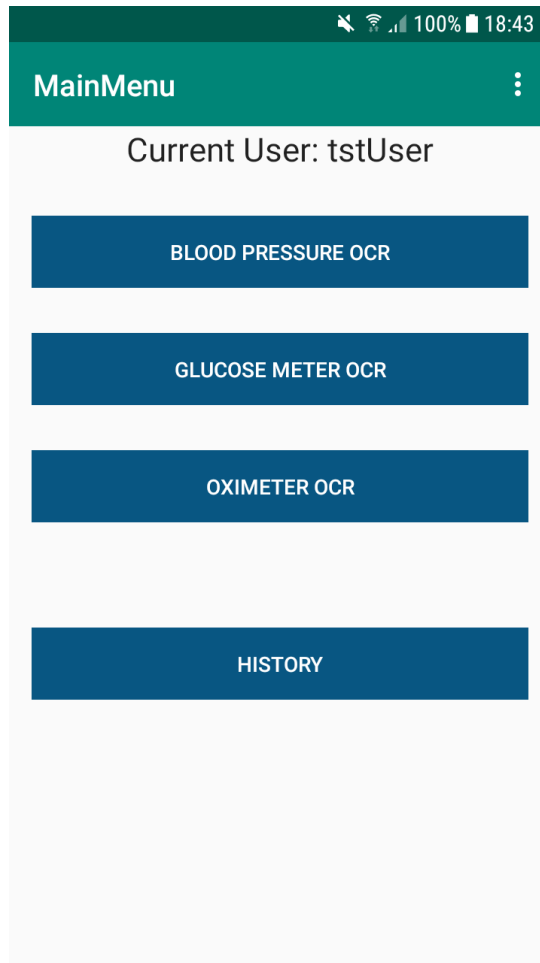
Confirm Password

CHANGE PASSWORD

Go to login page !

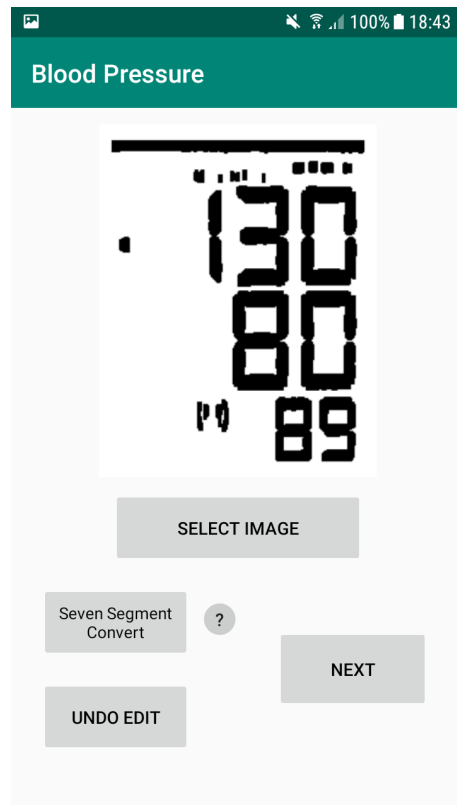
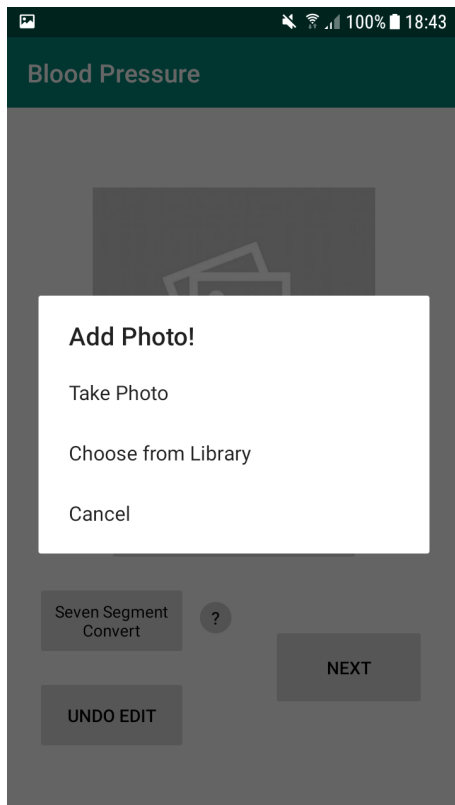
Οθόνη reset password

activity_menu: Είναι η κεντρική οθόνη της εφαρμογής, ο χρήστης έχει την δυνατότητα να εισάγει μια νέα μέτρηση, να δει το ιστορικό του ή να αποσυνδεθεί από τον λογαριασμό του.



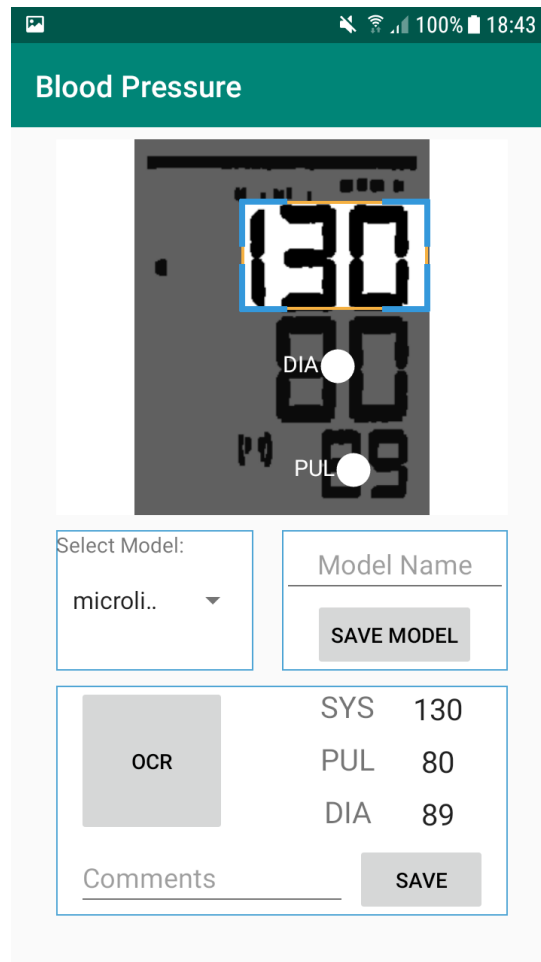
Κύρια οθόνη εφαρμογής

activity_select_image: Στην περίπτωση που επιλέξει να εισάγει μια νέα μέτρηση μεταβαίνει στην οθόνη επιλογής εικόνας. Εδώ ο χρήστης πατώντας το “select image” έχει την δυνατότητα να εισάγει μια νέα εικόνα είτε από την συσκευή ή κάνοντας λήψη από την κάμερα. Στην συνέχεια η εφαρμογή απομονώνει αυτόματα την οθόνη και δίνεται η δυνατότητα να απαλειφθούν όλες οι επεξεργασίες που έγιναν στην εικόνα άλλα και η επιλογή “seven segment convert” η οποία μετατρέπει τα τμηματικά νούμερα (7-segment) σε πιο ευανάγνωστα. Όταν τελειώσει με την επεξεργασία μπορεί να πατήσει next και να μεταφερθεί στην επόμενη οθόνη, όπου θα γίνει η οπτική αναγνώριση.



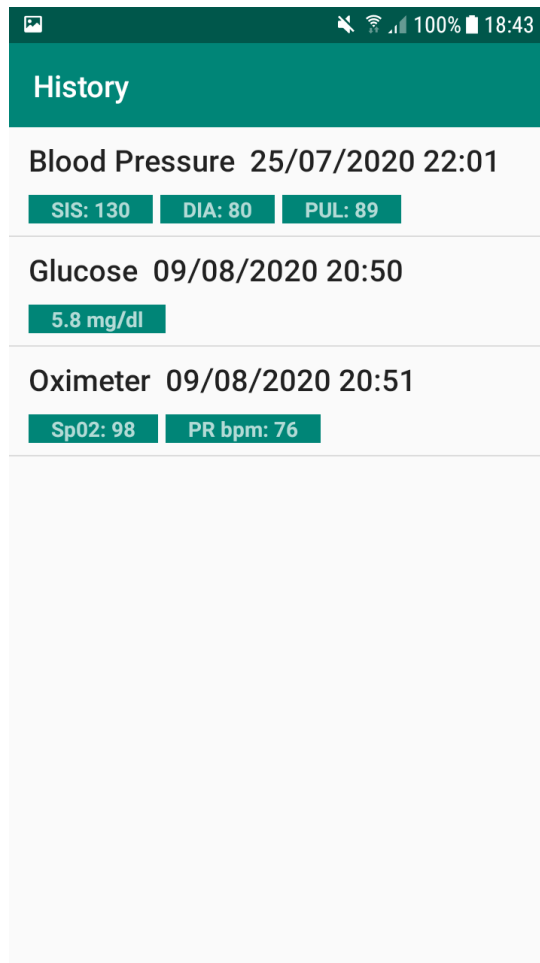
Οθόνη επιλογής και επεξεργασίας εικόνας

activity_ocr: Στην οθόνη αυτή η εφαρμογή αναγνωρίζει τις τιμές από την ιατρική συσκευή. Στην περίπτωση που το μοντέλο υπάρχει στην βάση δεδομένων εντοπίζεται αυτόματα σε ποια σημεία βρίσκονται οι μετρήσεις ενώ διαφορετικά ο χρήστης μπορεί να υποδείξει και να αποθηκεύσει το μοντέλο στην βάση δεδομένων. Επίσης μπορεί να καταχωρίσει σχόλια και να τα αποθηκεύσει στο online ιστορικό του μαζί με τις μετρήσεις.



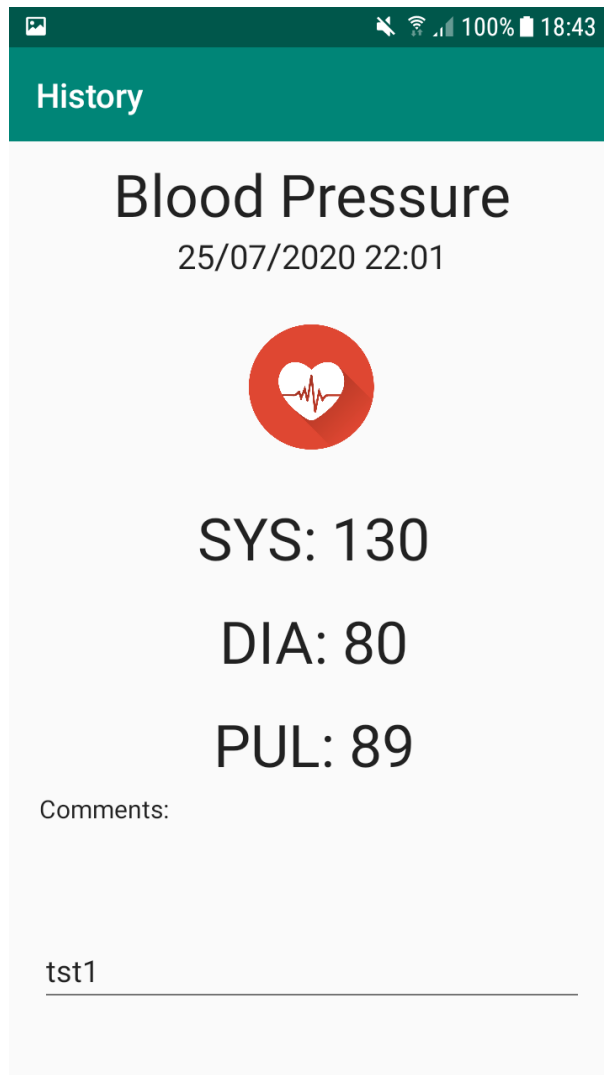
Οθόνη OCR

activity_history: Ο χρήστης μπορεί να δει το ιστορικό των μετρήσεων, οι τιμές σε κάθε μέτρηση, η ημερομηνία και ώρα που αποθηκεύτηκαν.



Οθόνη ιστορίας μετρήσεων

activity_history_item: Επιλέγοντας μια μέτρηση εισέρχεται σε αυτήν την οθόνη, όπου εκτός από το είδος της μέτρησης, τις τιμές και την ημερομηνία έχει πρόσβαση και στα σχόλια τα οποία είχε αποθηκεύσει κατά την μέτρηση!



Οθόνη μέτρησης

6.3 Σύνδεση εφαρμογής με την βάση δεδομένων

Για την επικοινωνία της εφαρμογής με την βάση δεδομένων θα χρησιμοποιήσουμε rest api. Η δουλειά του είναι να παίρνει αιτήματα από τον χρήστη, να επικοινωνεί με την βάση δεδομένων και να απαντάει στον χρήστη. Έτσι, εκτός από την επικοινωνία με τον server, επιτυγχάνουμε και την ασφάλεια των δεδομένων επικοινωνίας, αφού ο server είναι αυτός που αποφασίζει για τις απαντήσεις του. Για αυτό τον σκοπό, θα δημιουργήσουμε ένα PHP, MySQL API.

Το API μας είναι υπεύθυνο για:

- Να αποδεχτεί τα αιτήματα σε μεθόδους POST/GET.
- Να αλληλεπιδράσει με την βάση δεδομένων κάνοντας εισαγωγή ή ανάκτηση δεδομένων.
- Να δώσει απάντηση σε μορφή JSON.

Τα php αρχεία λοιπόν που έχουμε δημιουργήσει στον server για να επιτευχθεί η επικοινωνία είναι τα εξής:

adConnect.php: Ένα βασικό αρχείο που επιτυγχάνει την σύνδεση με την βάση δεδομένων και πρέπει να υπάρχει σε κάθε αρχείο που θέλουμε να αλληλεπιδράσει με αυτή.

```
dbConnect.php > ...
1  <?php
2      define('HOST','localhost');
3      define('USER','');
4      define('PASS','');
5      define('DB','');
6
7      $con = mysqli_connect(HOST,USER,PASS,DB) or die('Unable to Connect');
8  ?>
```

Κώδικας σύνδεσης με db

Login.php: Δέχεται σαν στοιχεία το email και το password ελέγχει άμα αντιστοιχεί λογαριασμός και αν ναι επιστρέφει το όνομα του δικαιούχου, αλλιώς επιστρέφει μήνυμα αποτυχίας.

```
login.php > ...
1  <?php
2      if($_SERVER['REQUEST_METHOD']=='POST'){
3          require_once('dbConnect.php');
4
5          $username = $_POST['email'];
6          $passwords = $_POST['password'];
7
8          $sql = $con->prepare("SELECT Name,password FROM account WHERE email='$username'");
9          $sql->bind_result($Name,$password);
10         $sql->execute();
11         $row = $sql->fetch();
12         if ($row > 0) {
13             if (password_verify($_POST['password'], $password)){
14                 echo "$Name";
15             }
16             // invalid password
17             else{
18                 echo "failure";
19             }
20         }
21         // invalid username
22         else{
23             echo "failure";
24         }
25         mysqli_close($con);
26     }
27     ?>
```

Κώδικας σύνδεσης χρήστη

Τα υπόλοιπα αρχεία είναι τα:

register.php (Δημιουργεί νέο λογαριασμό)

addHistory.php (αποθηκευη την μέτρηση που έκανε ο χρήστης στην βάση δεδομένων)

getHistory.php (Επιστρέφει στον χρήστη το ιστορικό με της μετρήσεις που αντιστοιχούν στο email του)

forgotPass.php (στελνει στο email του χρήστη ένα μοναδικό κωδικό (token) για να επαναφέρει τον κωδικό του και εισάγει τον κωδικό αυτό στην βάση δεδομένων)

changePass.php (Αλλάζει τον κωδικό του χρήστη)

addCoordinates.php (αποθηκεύει το νέο μοντέλο,και τα σημεία στα οποία υπάρχουν τιμές)

getCoordinates.php (Ανάλογα με την μέτρηση που κάνει ο χρήστης (blood pressure, oximeter, glucose) επιστρέφει στον χρήστη όλα τα αποθηκευμένα μοντέλα)

Παράδειγμα αιτηματος post απο την εφαρμογη προς τον server, για να πραγματοποιηθεί login, όπου κάνει έτοιμα με τα στοιχεία email και password του χρήστη και δέχεται απάντηση το όνομα του χρήστη αν είναι επιτυχημένο ή failure αν απέτυχε η σύνδεση και δειχνει το ανάλογο μηνυμα

```
//Creating a string request
StringRequest stringRequest = new StringRequest(Request.Method.POST, Config.LOGIN_URL,
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            //If we are getting success from server
            if(!response.trim().equalsIgnoreCase( anotherString: "failure")){
                //Starting profile activity
                Intent intent = new Intent( packageContext Login.this, MainMenu.class);
                startActivity(intent);
                finish();
            }else{
                //If the server response is not success
                Log.d( tag: "myTag", response.trim());
                Toast.makeText( context Login.this, text: "Invalid username or password", Toast.LENGTH_LONG).show();
            }
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            if (error != null)
                Log.d( tag: "myTag", error.toString());
        }
    }){
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String,String> params = new HashMap<>();
        //Adding parameters to request
        params.put(Config.KEY_EMAIL, email);
        params.put(Config.KEY_PASSWORD, password);
        return params;
    }
};
```

Αίτημα Post και απάντηση από τον Server

Στην εφαρμογή εγκαταστάθηκε η βιβλιοθήκη volley (implementation 'com.android.volley:volley:1.1.0') για να γίνεται η επικοινωνία με τον server ευκολότερα, αλλά κύριος γρηγορότερα.

Για να είναι ευκολότερο να παραμετροποιήσουμε την εφαρμογή έχει δημιουργηθεί config το αρχείο όπου περιέχει όλα τα link που χρειάζεται η εφαρμογή, έτσι ώστε σε περίπτωση αλλαγής του server να μην χρειαστεί να αναζητήσει τα links σε όλο τον κώδικα. Επίσης υποθηκεύονται στην συσκευή τα στοιχεία email, name σαν sharedPreferences όταν ο χρήστης συνδεθεί στον λογαριασμό του έτσι ώστε να μην χρειάζεται να τα ξανα γράφει κάθε φορά που ανοίγει την εφαρμογή αλλά να μένει συνδεδεμένος.

```
public class Config {
    private static String site = "kagialikos.com/ocr";

    public static final String LOGIN_URL = "https://" + site + "/login.php";
    public static final String REGISTER_URL = "https://" + site + "/register.php";
    public static final String AddHistory_URL = "https://" + site + "/addHistory.php";
    public static final String GetHistory_URL = "http://" + site + "/getHistory.php";
    public static final String GetCoordinates_URL = "http://" + site + "/getCoordinates.php";
    public static final String AddCoordinates_URL = "https://" + site + "/addCoordinates.php";
    public static final String token_send_URL = "https://" + site + "/forgotPass.php";
    public static final String changepass_URL = "https://" + site + "/changePass.php";

    public static final String KEY_EMAIL = "email";
    public static final String KEY_PASSWORD = "password";
    public static final String LOGIN_SUCCESS = "success";

    public static final String SHARED_PREF_NAME = "loginApp";
    public static final String EMAIL_SHARED_PREF = "email";
    public static final String NAME_SHARED_PREF = "name";
    public static final String LOGGEDIN_SHARED_PREF = "loggedin";
}
```

Αρχείο config

Για να μπορούμε να έχουμε μια επιτυχή σύνδεση στο διαδίκτυο, πρόσβαση στην κάμερα αλλά και στις αποθηκευμένες εικόνες πρέπει να επιτρέψουμε στην εφαρμογή μας τις λειτουργίες αυτές. Κάτι τέτοιο επιτυγχάνετε με την χορήγηση αδειών στο AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CAMERA"/>
```

Χορήγηση αδειών στην εφαρμογή

Κεφάλαιο 7

7.1 Συμπεράσματα

Είναι αναμφισβήτητο ότι η ηλεκτρονική υγείας τα τελευταία χρόνια έχει αναπτυχθεί σημαντικά ενώ προβλέπεται ότι στο μέλλον η διάδοση της θα είναι αρκετά μεγαλύτερη. Αυτή η εφαρμογή στοχεύει να βοηθήσει σε αυτόν τον τομέα, προσφέροντας στους χρήστες έναν εύκολο, οικονομικό και ασφαλή τρόπο για να καταγράψουν και να αποθηκεύουν online της μετρήσεις από τα ιατρικά όργανα, καθιστώντας έτσι δυνατόν πραγματοποίησή μετρήσεων σε καθημερινή βάση, ή και συχνότερα, αφού ο χρόνος που χρειάζεται είναι ελάχιστος.

Σημαντικές προκλήσεις που παρουσιάστηκαν κατά την δημιουργία της εφαρμογής ήταν η εύρεση των καταλληλότερων τιμών για την επεξεργασία εικόνας, έτσι ώστε να έχουμε το ιδανικότερο αποτέλεσμα, αλλά και η αναγνώριση αριθμών από την οθόνη οι οποίοι δεν ήταν ψηφιακής μορφής όπως θα ήταν αυτοί από ένα σκαναρισμένο κείμενο. Στην συνέχεια έπρεπε να όλες οι απαραίτητες συναρτήσεις από διαφορετικές βιβλιοθήκες να συνεργαστούν στο περιβάλλον της εφαρμογής android, με το αποτέλεσμα όμως να είναι αρκετά ικανοποιητικό .

7.2 Μελλοντικές επεκτάσεις

Στο μέλλον θα μπορούσαν να προστεθούν επιπλέον λειτουργίες στην εφαρμογή, όπως η δυνατότητα για σάρωση πολλών εικόνων που εισάγονται από το κινητό ταυτόχρονα.

Ακόμα θα μπορούσε να στέλνει την φωτογραφία στον server για να γίνει η επεξεργασία της εικόνας εκεί και να μην χρειάζεται να δεσμεύει πόρους της συσκευής.

Θα μπορούσε να δίνει συμβουλές ή και διαγνώσεις με βάση της μετρήσεις που λαμβάνει για να ενημερώνει τον ασθενή για την κατάσταση της υγείας του και να υπενθυμίζει στον ασθενή σε καθημερινή βάση να ελέγχει την υγεία του με νέες μετρήσεις.

Τέλος θα μπορούσε να δημιουργηθεί ίδια εφαρμογή για κινητά iOS έτσι ώστε να έχουν και αυτοί την δυνατότητα να την χρησιμοποιήσουν, η να έχει ο ασθενής πρόσβαση στα αρχεία του αν αλλάξει κινητό .

Βιβλιογραφία

1. Σταυρακούδης Α.. (2010). Βάσεις Δεδομένων και SQL Μια Πρακτική Προσέγγιση. Αθήνα, Κλειδάριθμος.
2. Χατζόπουλος Μ.. (2012). Θεμελιώδης Αρχές Συστημάτων Βάσεων Δεδομένων (6η έκδοση). (R. Elmasri & S.B. Navath.). Αθήνα, Δίαυλος (Ημερομηνία αυθεντικής έκδοσης 2012)
3. Bernd Jähne-Handbook of Computer Vision and Applications
4. James S. Walker-Wavelet-based image processing
5. Gunilla Borgefors. Distance transformations in digital images. Computer vision
6. Stephane Mallat – A wavelet tour of signal processing
7. V.K. Govindan, A.P. Shivaprasad- Character Recognition: A Review
8. J. Mantas- An Overview of Character Recognition Methodologies
9. John Canny. A computational approach to edge detection
10. Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extremas for realtime feature detection and matching. - *Computer Vision–ECCV*
11. *Ray Smith - An overview of the Tesseract OCR Engine*
12. Alexandre Alahi, Raphael Ortiz, and Pierre Vanderghenst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition*
13. Tesseract OCR Engine: What it is, where it came from,where it is going.,Ray Smith, Google Inc,OSCON 2007
14. Kari Pulli, Anatoly Baksheev, Kirill Korniyakov, Victor Eruhimov - Realtime Computer Vision with OpenCV
15. W3schools (1999). PHP5 Tutorial/Easy Learning with “Show PHP”. Retrieved from <http://www.w3schools.com/php/>
16. W3schools (1999). SQL Tutorial. Retrieved from <http://www.w3schools.com/sql/>
17. Opencv - <https://opencv.org>
18. Android Studio - www.developer.android.com/studio/index.html

19. Android - www.developer.android.com/develop/index.html
20. Tesseract - <https://opensource.google/projects/tesseract>
21. PHP login logout code - <http://phplogincode.blogspot.gr/>
22. PHPMailer/PHPMailer. - <https://github.com/PHPMailer/PHPMailer>