



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

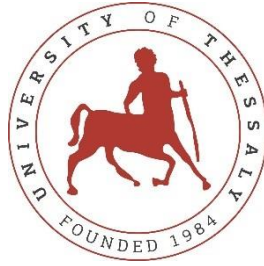
**ΣΧΕΔΙΑΣΜΟΣ ANDROID ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΣΑΡΩΣΗ ΚΑΙ
ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΙΚΟΝΑΣ ΣΕ ΕΓΓΡΑΦΟ**

Διπλωματική Εργασία

Γκεβρέκη Σοφία

Επιβλέπουσα: Τσαλαπάτα Χαρίκλεια

Βόλος 2021



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**ANDROID APPLICATION FOR SCANNING AND OPTIMIZING
IMAGES TO DOCUMENTS**

Diploma Thesis

Gkevreki Sofia

Supervisor: Tsalapata Harikleia

Volos 2021

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πυσα **Τσαλαπάτα Χαρίκλεια**

Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Δασκαλοπούλου Ασπασία**

Επίκουρη Καθηγήτρια, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Φεύγας Αθανάσιος**

Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 24-09-2021

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

Γκεβρέκη Σοφία

24-09-2021

ΠΕΡΙΛΗΨΗ

Τις τελευταίες δεκαετίες, νέες τεχνολογίες έχουν συστηθεί ως ένα σημαντικό και χρήσιμο εργαλείο, που καθιστά την καθημερινότητα των ανθρώπων πιο εύκολη και γρήγορη. Ιδιαίτερα τη σημερινή εποχή, τα έξυπνα κινητά (smartphones) μέσω των εφαρμογών τους έχουν μια συνεχώς αυξανόμενη παρουσία στην προσωπική και επαγγελματική ζωή των ανθρώπων, βελτιώνοντας έτσι την ποιότητα της σε διάφορους τομείς. Η παρούσα διπλωματική εργασία πραγματεύεται την κατασκευή μιας εφαρμογής για κινητές συσκευές, όπου επιτρέπει στους χρήστες της να “σαρώσουν” εικόνες, να τις βελτιστοποιήσουν και να τις μοιράσουν στη μορφή που επιθυμούν οι ίδιοι. Είναι κατανοητό πως μια τέτοια εφαρμογή μπορεί να επιφέρει πολλά χρήσιμα αποτελέσματα στον άνθρωπο, ο οποίος συνεχώς επιδιώκει μείωση χρόνου και αύξηση αποδοτικότητας σημαντικών δραστηριοτήτων. Μια τέτοια πρακτική εφαρμογή μπορεί να αντικαταστήσει τις ηλεκτρονικές συσκευές σάρωσης εικόνας (scanner) σε ένα μεγάλο βαθμό και να επιτυγχάνει γρηγορότερα την ψηφιοποίηση της, διαδικασία αναμφίβολα απαραίτητη την σημερινή εποχή. Η εργασία ολοκληρώθηκε με την χρήση κατάλληλου λογισμικού περιβάλλοντος (Android Studio) και τεχνικών που θα αναλυθούν. Η προγραμματιστική γλώσσα που χρησιμοποιήθηκε είναι η Kotlin.

ABSTRACT

In the last few decades, new technologies have been introduced as a major and useful tool, making people's lives easier and faster. Especially nowadays, smartphones through their applications have an ever-increasing presence in both personal and professional people's lives by enhancing its quality in several ways. This thesis looks into the construction of a mobile application that allows its users to scan images, optimize them and share them in a format of their choice. It is understood that such an application can bring many useful results in regards to reducing the duration and increasing the productivity of important tasks. An application such as this could possibly not only replace the electronic scanners up to a certain point, but also increase the digitalization time of images, something of major importance today. This work was completed with the usage of appropriate software environments (Android Studio) and techniques that will be analyzed here. The programming language used is Kotlin.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	8
ABSTRACT.....	10
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	12
ΚΕΦΑΛΑΙΟ 1.....	16
ΕΙΣΑΓΩΓΗ.....	16
1.1 Η τεχνολογία στη ζωή των ανθρώπων.....	16
1.2 Smartphones και εφαρμογές.....	16
1.3 Ηλεκτρονικός σαρωτής.....	16
1.4 Η ιδέα (Image Scanner).....	17
1.5 Οργάνωση κεφαλαίων.....	17
ΚΕΦΑΛΑΙΟ 2.....	18
ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΙΔΕΑΣ.....	18
2.1 Εφαρμογή Image Scanner.....	18
2.2 Λόγοι που οδήγησαν στην πραγματοποίηση.....	18
2.3 Σύγκριση με παρόμοιες εφαρμογές.....	19
ΚΕΦΑΛΑΙΟ 3.....	22
ΗΛΕΚΤΡΟΝΙΚΟΣ ΣΑΡΩΤΗΣ.....	22
3.1 Ορισμός.....	22
3.2 Ιστορία του σαρωτή.....	23
3.3 Πλεονεκτήματα του σαρωτή.....	23
3.4 Χρήσεις το σαρωτή.....	25
ΚΕΦΑΛΑΙΟ 4.....	26
ΤΕΧΝΟΛΟΓΙΚΗ ΕΠΙΣΚΟΠΗΣΗ.....	26
4.1 Εφαρμογές για κινητές συσκευές.....	26
4.1.1 Ορισμός.....	26
4.1.2 Android λογισμικό.....	26
4.2 Android Studio.....	27
4.2.1 Ορισμός.....	27

4.2.2 Προγραμματιστική γλώσσα Kotlin.....	29
4.3 Το περιβάλλον του Android Studio.....	29
4.3.1 Εισαγωγή.....	29
4.3.2 Activities.....	30
4.3.3 Fragments.....	31
4.3.4 Layout - Αρχεία XML.....	32
4.3.5 Android Manifest.....	33
4.3.6 Gradle.....	34
4.4 Τύποι αρχείων	34
4.4.1 Εικόνα JPG (ή JPEG).....	34
4.4.2 Έγγραφο PDF	34
4.5 Έλεγχος εκδόσεων (version control).....	35
4.5.1 Χρησιμότητα.....	35
4.5.2 Git.....	36
4.5.3 Github.....	36
ΚΕΦΑΛΑΙΟ 5.....	38
ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ.....	38
5.1 Εισαγωγή.....	38
5.2 Αλγόριθμος έξυπνης περικοπής.....	38
5.2.1 Εισαγωγή.....	38
5.2.2 Ανίχνευση ακρών.....	39
5.2.3 Περικοπή εικόνας.....	39
5.3 Βελτιστοποίηση εικόνας.....	39
5.3.1 Εισαγωγή.....	39
5.3.2 Κατωφλίωση (thresholding) και μονοχρωματισμός (grayscale).....	40
5.3.3 Αφαίρεση θορύβων (noise removal).....	41
5.3.4 Παράδειγμα.....	41
ΚΕΦΑΛΑΙΟ 6.....	43
ΤΟ IMAGE SCANNER ΑΝΑΛΥΤΙΚΑ.....	43

6.1 Εισαγωγή.....	43
6.2 Εμπειρία χρήσης.....	43
6.2.1 Υπηρεσίες εφαρμογής.....	43
6.2.2 Παρουσίαση εφαρμογής.....	44
6.2.3 Αποτελέσματα εφαρμογής.....	48
6.2.4 Χρησιμότητα εφαρμογής.....	52
6.3 Τεχνικό κομμάτι.....	52
6.3.1 Περιεχόμενα της εφαρμογής στο <i>Android Studio</i>	52
6.3.2 Ανάλυση του κώδικα και των αλγορίθμων.....	55
6.3.3 Επιπρόσθετες τεχνικές και έννοιες.....	70
6.4 Μελλοντικές επεκτάσεις.....	75
ΚΕΦΑΛΑΙΟ 7.....	77
ΕΠΙΛΟΓΟΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.....	77
ΠΑΡΑΡΤΗΜΑ.....	78
ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΜΕΛΕΤΗ ΤΟΥ ΚΩΔΙΚΑ.....	78
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	79

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Η τεχνολογία στη ζωή των ανθρώπων

Η τεχνολογία με τη ραγδαία εξέλιξη που έχει γνωρίσει τις τελευταία χρόνια, έχει τεράστιες επιπτώσεις στην ζωή των ανθρώπων. Από την αρχή των επιστημονικών ανακαλύψεων μέχρι σήμερα, το ανθρώπινο γένος έχει δει αμέτρητες αλλαγές στη ζωή του και στο περιβάλλον του.

Στη σημερινή κοινωνία, μια εφεύρεση που έχει μεγαλύτερο αντίκτυπο στην κοινωνία και που άλλαξε εντελώς τον τρόπο που ο άνθρωπος εργάζεται, σκέφτεται και επεξεργάζεται είναι η τεχνολογία των υπολογιστών και το διαδίκτυο. Οι υπολογιστές έχουν κάνει την επεξεργασία απλή και γρήγορη και πλέον, αποτελούν ουσιαστικό μέρος της καθημερινότητας.

1.2 Smartphones και εφαρμογές

Οι υπολογιστές έρχονται σε πολλά σχήματα και μεγέθη. Χαρακτηριστικό παράδειγμα μικρού μεγέθους υπολογιστή αποτελούν τα κινητά τηλέφωνα. Ιδιαίτερα, τα “έξυπνα” τηλέφωνα (Smartphones), είναι κινητά τηλέφωνα βασισμένα σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερο προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα. Τα smartphones είναι χρήσιμες, πρακτικές και βολικές συσκευές που μπορούν όλοι να χρησιμοποιήσουν για να διευκολύνουν τις καθημερινές τους δραστηριότητες. Με το λειτουργικό σύστημα που διαθέτουν, δίνουν τη δυνατότητα στους χρήστες να εγκαταστήσουν εφαρμογές σε αυτά, οι οποίες διευρύνουν τις ικανότητες τους.

Πλέον, υπάρχουν πάνω από δύο εκατομμύρια εφαρμογές όπου οι χρήστες μπορούν να κατεβάσουν και κάθε μια από αυτές εξυπηρετεί και διαφορετικό σκοπό. Υπάρχουν εφαρμογές παραγωγικότητας, που επικεντρώνονται στη βελτίωση της αποτελεσματικότητας των επιχειρήσεων, εκπαιδευτικές και ταξιδιωτικές εφαρμογές, εφαρμογές που προσφέρουν σημαντικά εργαλεία, εφαρμογές κοινωνικής δικτύωσης, παιχνίδια που διασκεδάζουν τον χρήστη καθώς και πολλά άλλα. Όλες, έχουν σαν κοινό στοιχείο την βελτίωση της ανθρώπινης ζωής με τον δικό τους μοναδικό τρόπο.

1.3 Ηλεκτρονικός σαρωτής

Ο σαρωτής εικόνας (scanner) είναι μια συσκευή που σαρώνει οπτικά εικόνες, τυπωμένο κείμενο, χειρόγραφο, φωτογραφικές εκτυπώσεις, αφίσες, σελίδες περιοδικών και άλλα αντικείμενα και τα μετατρέπει σε ψηφιακή εικόνα. Η διαφορά ενός σαρωτή με μία εικόνα ψηφιακής φωτογραφικής κάμερας, είναι ότι αυτή υπόκειται σε ένα βαθμό παραμόρφωσης, αντανάκλασεων, σκιών, χαμηλής αντίθεσης και θολούρας, κάτι που πολλές φορές αποφεύγεται.

Ανάλογα με τις επιθυμίες του χρήστη, ο σαρωτής παρέχει διάφορες χρήσιμες λειτουργίες όπως αντιγραφή χειρόγραφου ή εκτυπωμένου κειμένου, αρχειοθέτηση σημαντικών εγγράφων, σάρωση βιβλίων ή πληροφοριών που απαιτούνται σε έρευνες, εύκολη αποστολή αρχείων και φωτογραφιών κ.ο.κ.

1.4 Η ιδέα (Image Scanner)

Η εφαρμογή που κατασκευάστηκε για χάρη της παρούσας διπλωματικής εργασίας είναι μια εφαρμογή για Android κινητές συσκευές που “σαρώνει” εικόνες και τις μετατρέπει σε ευανάγνωστα έγγραφα. Το Image Scanner αποτελεί ένα χρήσιμο εργαλείο της σημερινής εποχής που μπορεί να χρησιμοποιηθεί καθημερινά και να ικανοποιήσει διάφορες ανάγκες.

1.5 Οργάνωση κεφαλαίων

Στο δεύτερο Κεφάλαιο, περιγράφεται η ιδέα και οι λόγοι που οδήγησαν στην υλοποίηση της. Παράλληλα, περιλαμβάνεται και μια σύντομη επισκόπηση παρόμοιων εφαρμογών.

Στο Κεφάλαιο 3, αναλύονται οι ηλεκτρονικοί σαρωτές και η χρήση τους και περιγράφονται οι λόγοι που μπορεί να χρησιμοποιείσαι κάποιος ένα scanner.

Στο επόμενο κεφάλαιο, γίνεται μια τεχνολογική επισκόπηση διάφορων τεχνολογιών. Πιο συγκεκριμένα, δίνεται ο ορισμός των εφαρμογών για κινητές συσκευές, περιγράφεται το περιβάλλον Android Studio, η γλώσσα προγραμματισμού Kotlin και ορίζονται οι τύποι αρχείων JPG και PDF.

Στο πέμπτο κεφάλαιο παρουσιάζεται η σημαντικότητα της επεξεργασίας εικόνας και αναλύονται οι αλγόριθμοι και τα φίλτρα που χρησιμοποιήθηκαν στην εφαρμογή.

Στη συνέχεια, στο Κεφάλαιο 6 δίνεται η αναλυτική περιγραφή της εφαρμογής Image Scanner, από την εμπειρία χρήσης της μέχρι την μελέτη του κώδικα. Σε αυτό το κεφάλαιο περιλαμβάνονται επίσης διάφορα παραδείγματα από αποτελέσματα της εφαρμογής και προτάσεις για μελλοντική εξέλιξη.

Τέλος, στο έβδομο και τελευταίο κεφάλαιο παρουσιάζονται τα κυριότερα συμπεράσματα που εξάχθηκαν κατά την υλοποίηση της εργασίας.

ΚΕΦΑΛΑΙΟ 2

ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΙΔΕΑΣ

2.1 Εφαρμογή Image Scanner

Το Image Scanner είναι η εφαρμογή που υλοποιήθηκε και πραγματοποιήθηκε για αυτή την διπλωματική εργασία. Αποτελεί μια Android εφαρμογή κατά την οποία γίνεται εύκολα σάρωση εικόνων και παράλληλα η ωραιοποίηση τους.

Ο χρήστης έχει την επιλογή είτε να τραβήξει με την κάμερα του κινητού του μια εικόνα, είτε να επιλέξει μια υπάρχουσα από τα αρχεία της συσκευής του. Η εφαρμογή αναλύοντας αυτόματα το περιεχόμενο προτείνει έναν έξυπνο τρόπο περικοπής, απαλείφοντας έτσι άσκοπα κενά και δυσαναλογίες. Έπειτα, η εικόνα περνάει στη διαδικασία της βελτιστοποίησης, μέσω της οποίας το γραπτό της φωτογραφίας περνάει από ειδικά φίλτρα και γίνεται πιο “καθαρό” και εύκολα αναγνωρίσιμο. Τέλος, ο χρήστης έχει την δυνατότητα να αποθηκεύσει την φωτογραφία στο κινητό του και να την στείλει εύκολα σε όποιο κοινωνικό μέσο θέλει και στη μορφή που επιθυμεί (φωτογραφία ή έγγραφο). Εκτός από την διαδικασία της βελτιστοποίησης, η εφαρμογή προσφέρει και έναν μετατροπέα εγγράφων, δηλαδή χρήστης μπορεί να μετατρέψει οποιαδήποτε εικόνα της συσκευής του σε έγγραφο και να το μοιράσει σε άλλους ή να το αποθηκεύσει.

Η ύπαρξη μιας τέτοιας εφαρμογής έχει πολλά πλεονεκτήματα και χρήσεις στην ανθρώπινη ζωή και προσφέρει άμεσα και γρήγορα αποτελέσματα για την κάλυψη καθημερινών αναγκών.

2.2 Λόγοι που οδήγησαν στην πραγματοποίηση

Στην σημερινή καθημερινότητα, η χρήση ενός σαρωτή είναι μια χρήσιμη διαδικασία που πολλές φορές μπορεί να γίνει και απαραίτητη.

Για παράδειγμα, είναι συχνοί οι λόγοι για τους οποίους χρειάζονται πολλά αντίγραφα μιας εικόνας ή ενός αρχείου τόσο σε επαγγελματικούς χώρους, όσο και προσωπικούς. Επίσης, ένα χαρτί εκτυπωμένο είναι αρκετά εύκολο να καταστραφεί ή να αλλοιωθεί και έτσι να γεννιέται η ανάγκη για ένα ψηφιακό αντίγραφο που μπορεί να διατηρηθεί καλύτερα. Πολλές φορές χρειάζεται να έχει κάποιος κρατημένα σημαντικά αρχεία, φωτογραφίες, σημειώσεις στην κινητή του συσκευή. Ακόμα, το μοίρασμα εικόνων και εγγράφων είναι μια διαδικασία που σε αυτή την εποχή της τεχνολογίας αποτελεί σχεδόν καθημερινή ενέργεια. Από φοιτητές που πρέπει να στείλουν θέματα και εργασίες από μαθήματα στους καθηγητές, μέχρι και στις δημόσιες υπηρεσίες για αποστολή συμπληρωμένων δηλώσεων και σημαντικών εγγράφων. Παράλληλα, είναι χρήσιμη η αποθήκευση και η οργάνωση αποδείξεων, εισιτηρίων, σελίδων από περιοδικά ή βιβλία, αφισών κ.ο.κ.

Η χρήση του ηλεκτρονικού σαρωτή πολλές φορές μπορεί να είναι μια διαδικασία χρονοβόρα ή ακριβή. Οι άνθρωποι πλέον επιζητούν άμεσα αποτελέσματα με την ευκολία του σπιτιού τους.

Παράλληλα, η χρήση φωτογραφικής κάμερας δεν είναι πάντα η καλύτερη λύση διότι η εικόνα που τραβιέται μπορεί να έχει δυσαναλογίες και περιττά κενά, και η τελική εικόνα να εμφανίζεται δυσανάγνωστη και κουραστική.

Τα τελευταία χρόνια πολλές χρονοβόρες ενέργειες έχουν αντικατασταθεί από την χρήση εφαρμογών που προσφέρουν οι “έξυπνες” κινητές συσκευές. Υπάρχουν χιλιάδες χρήσιμες εφαρμογές που γλιτώνουν χρήμα και χρόνο και χρησιμοποιούνται συνεχώς.

Με βάση τα παραπάνω, και με την χρήση ήδη υπάρχων εφαρμογών σάρωσης, πραγματοποιήθηκε η υλοποίηση μιας εφαρμογής που λειτουργεί ως “σκάνερ” και κάνει την διαδικασία εύκολη και γρήγορη. Η ιδέα συνδυάζει πραγματοποίηση εφαρμογής Android και βελτιστοποίηση/επεξεργασία εικόνας, δυο κλάδοι αρκετά σημαντικοί και διαδομένοι στο χώρο της επιστήμης υπολογιστών.

2.3 Σύγκριση με παρόμοιες εφαρμογές

Στην αγορά εφαρμογών Android, υπάρχουν μερικές εφαρμογές που έχουν παρόμοια χρήση με την εφαρμογή Image Scanner.

Η πιο δημοφιλής από αυτές είναι το CamScanner. Το CamScanner είναι μια εφαρμογή σάρωσης η οποία έχει εγκατασταθεί σε πάνω από 500 εκατομμύρια συσκευές σε περισσότερες από 200 χώρες σε όλο τον κόσμο.

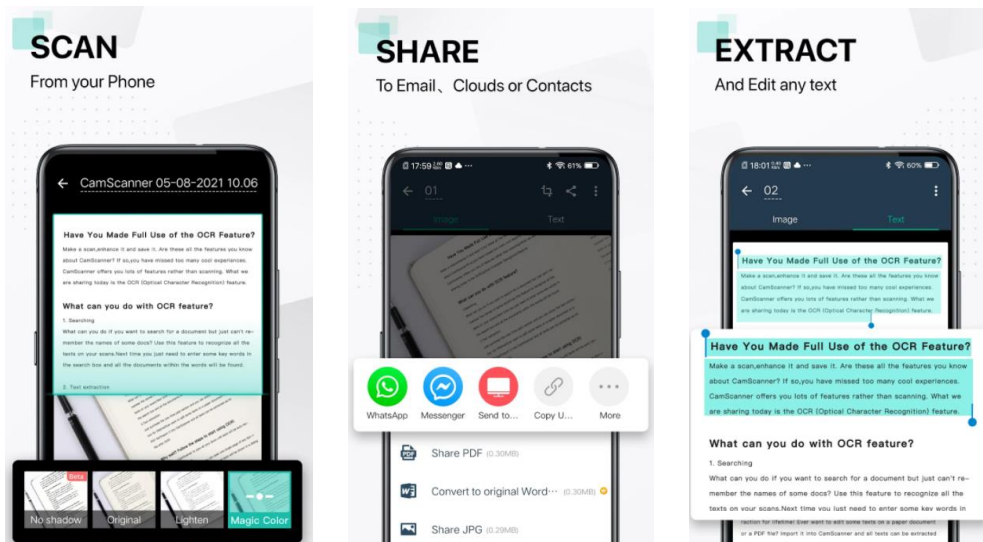


Εικόνα 2.1 Το λογότυπο της εφαρμογής CamScanner [1]

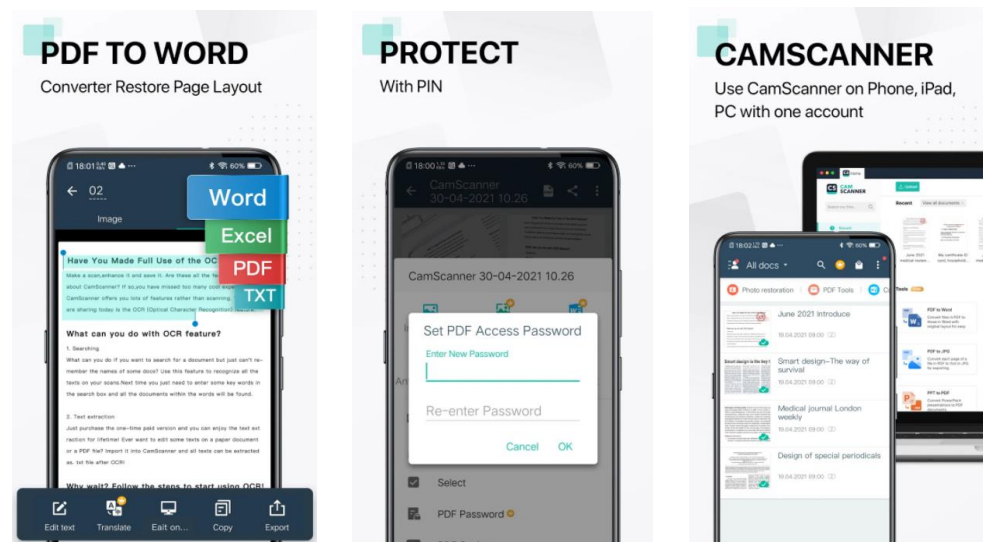
Οι λειτουργίες που προσφέρει είναι οι εξής: [1]

- Γρήγορη ψηφιοποίηση εγγράφου
- Βελτιστοποίηση ποιότητας σάρωσης
- Εξαγωγή κειμένων από την εικόνα με χρήση OCR (Optical character recognition)
- Κοινή χρήση αρχείων PDF/JPEG
- Επεξεργασία εγγράφων με προσθήκη σχολιασμών

- Άμεση εκτύπωση εγγράφων με κοντινό εκτυπωτή
- Γρήγορη αναζήτηση
- Ασφάλεια σημαντικών εγγράφων
- Συγχρονισμός με άλλες συσκευές



Εικόνα 2.2 (α, β, γ) Κάποιες λειτουργίες του CamScanner [1]

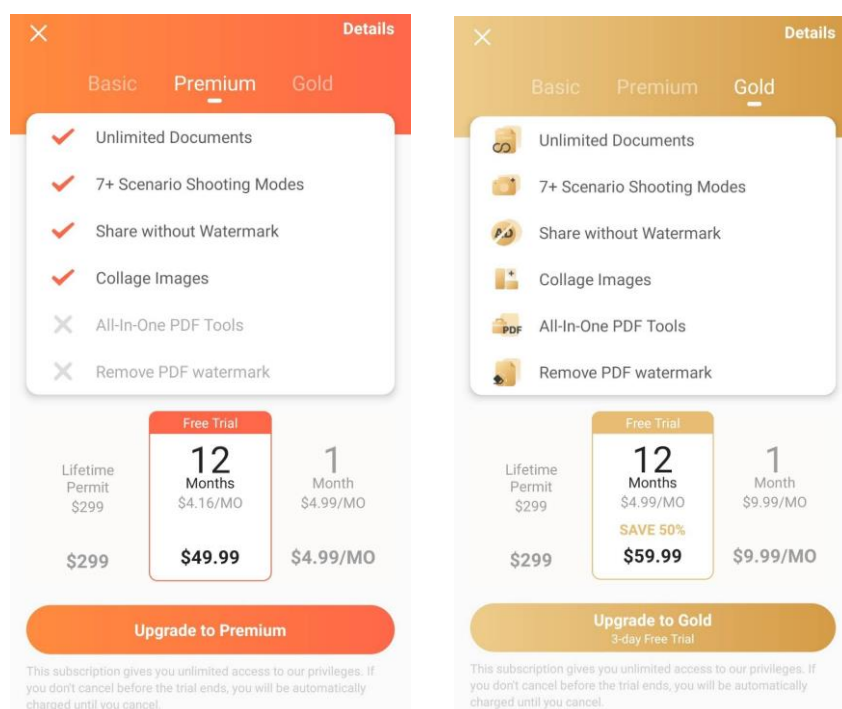


Εικόνα 2.2 (δ, ε, στ) Κάποιες λειτουργίες του CamScanner [1]

Η εφαρμογή προσφέρει κάποιες βασικές λειτουργίες δωρεάν (σάρωση αρχείων, βελτιστοποίηση και αποστολή), ενώ για τα υπόλοιπα χαρακτηριστικά θα πρέπει να γίνει αναβάθμιση σε είτε σε Premium λογαριασμό, είτε σε Gold λογαριασμό. Οι τιμές και οι λειτουργίες τους απεικονίζονται στην Εικόνα 2.3.

Το CamScanner είναι αναμφισβήτητα μια πολύ χρήσιμη εφαρμογή, φιλική ως προς τον χρήστη και εξαιρετικά αποδοτική. Πίσω από την πραγματοποίηση της βρίσκεται μια ολόκληρη ομάδα προγραμματιστών και σχεδιαστών που ασχολούνται επαγγελματικά με την υλοποίηση της εφαρμογής.

Η εφαρμογή Image Scanner προσφέρει βασικές λειτουργίες που προσφέρει και η δωρεάν έκδοση του CamScanner. Μια διαφορά είναι πως στο Image Scanner δεν είναι απαραίτητη η παρουσία του υδατόσημου (watermark) της εφαρμογής στο τέλος των εγγράφων, χαρακτηριστικό που πολλές φορές μπορεί να θεωρηθεί ανεπιθύμητο. Επίσης, η δωρεάν έκδοση του CamScanner εμφανίζει πολλές και μεγάλες σε χρόνο διαφημίσεις σε διάφορα σημεία, επιβραδύνοντας έτσι την όλη διαδικασία.



Εικόνα 2.3 (α, β) Τα χαρακτηριστικά και οι τιμές των εκδόσεων CamScanner επί πληρωμή

Άλλες παρόμοιες εφαρμογές είναι οι εξής:

- Adobe Scan
- Scanner App To PDF – TapScanner
- Simple Scan
- PDF Scanner App - Free Document Scanner & Scan PDF

Όλες παρέχουν τις βασικές λειτουργίες που χρειάζονται για την σάρωση εικόνων και τη μετατροπή τους σε έγγραφο.

ΚΕΦΑΛΑΙΟ 3

ΗΛΕΚΤΡΟΝΙΚΟΣ ΣΑΡΩΤΗΣ

3.1 Ορισμός

Ο ηλεκτρονικός σαρωτής, η κοινώς *scanner* είναι μια συσκευή που “διαβάζει” οπτικά μια εικόνα και τη μετατρέπει σε ψηφιακό σήμα. Μια τέτοια εικόνα μπορεί να είναι μια εκτυπωμένη φωτογραφία, ένα χειρόγραφο κείμενο, μια αφίσα, μια ζωγραφιά, ένα βιβλίο κ.ο.κ. Η σαρωμένη εικόνα μπορεί να χρησιμοποιηθεί έπειτα για επεξεργασία στον υπολογιστή.

Παραδοσιακά, ο σαρωτής χρησιμοποιείται στα γραφεία όπου το έγγραφο τοποθετείται σε γυάλινο παράθυρο και ετοιμάζεται για σάρωση. Υπάρχουν όμως διάφοροι άλλοι τύποι σαρωτών που χρησιμοποιούνται καθημερινά. Τα τρία βασικά είδη σαρωτών είναι τα εξής: [2]

- i. Οι **επίπεδοι** (flatbed), οι οποίοι παρέχουν μια επίπεδη γυάλινη επιφάνεια και η κεφαλή σάρωσης μετακινείται κάτω από αυτή (συνήθως χρησιμοποιούνται για σάρωση πολλών φύλλων τη φορά) (Εικόνα 3.1.1)
- ii. Οι **χειροκίνητοι** (handheld), μικρές φορητές συσκευές με τις οποίες ο χρήστης μετακινεί χειροκίνητα την κεφαλή
- iii. Οι **σαρωτές έλξης** (sheet-fed), που χρησιμοποιούνται μόνο για σάρωση χαρτιού μέσω μιας ακίνητης σταθερής κεφαλής.



Εικόνα 3.1.1 Ένας τυπικός σαρωτής τύπου flatbed [32]

Τα scanners λειτουργούν χρησιμοποιώντας την ανάκλαση φωτός που οι ίδιοι εκπέμπουν. Δηλαδή, φωτίζουν το αντικείμενο που ψηφιοποιείται και κατευθύνουν το ανακλώμενο φως σε ένα φωτοευαίσθητο στοιχείο. Στους περισσότερους σαρωτές, το μέσο ανίχνευσης είναι ένα ηλεκτρονικό, ολοκληρωμένο κύκλωμα ανίχνευσης φωτός γνωστό ως Φορτισμένη Συζευγμένη Συσκευή (CCD). Οι φωτοευαίσθητοι αισθητήρες τοποθετημένοι κατά μήκος του CCD μετατρέπουν τα επίπεδα φωτεινότητας σε ηλεκτρονικά σήματα που στη συνέχεια επεξεργάζονται σε ψηφιακή εικόνα μέσω ενός Αναλογικού Ψηφιακού Μετατροπέα (ADC). [3]

3.2 Ιστορία του σαρωτή

Οι σύγχρονοι σαρωτές θεωρούνται οι διάδοχοι των πρώιμων τηλεφωτογραφιών και συσκευών εισαγωγής φαξ. Πριν από 50 περίπου χρόνια, ο Rudolf Hell σχεδίασε μια συσκευή φαξ που χρησιμοποιήθηκε για τη μετάδοση πληροφοριών στην βιομηχανία των εφημερίδων και ήταν σε θέση να μεταδώσει έγγραφα με τη μορφή εικόνας που έχει υποστεί ραστεροποίηση σε γραμμές και εικονοστοιχεία. Αυτή η δαπανηρή και περίπλοκη συσκευή αναπτύχθηκε αργότερα στον επίπεδο σαρωτή, ο οποίος χρησιμοποίησε μια διαφορετική, πιο οικονομική τεχνολογία, το στοιχείο CCD. Περαιτέρω ανάπτυξη σαρωτή πραγματοποιήθηκε λόγω των διαφορετικών εργασιών που απαιτούνται. Αργότερα, η γραμμή CCD αντικαταστάθηκε από το τσιπ CCD (το τσιπ Bayer), το οποίο είναι σε θέση να διαβάσει ένα έγχρωμο έγγραφο σε κλάσματα του δευτερολέπτου διαβάζοντας τη μήτρα RGB-Bayer.

3.3 Πλεονεκτήματα του σαρωτή

Οι σαρωτές σήμερα παρέχουν υψηλή ποιότητα ως προς την ανάλυση, τόσο για έγχρωμα όσο και για ασπρόμαυρα έγγραφα. Οι σύγχρονοι πολυλειτουργικοί σαρωτές παρέχουν αρκετή λεπτομέρεια και ανάλυση για τον χειρισμό εικόνων, φωτογραφιών, αντικειμένων κ.τ.λ. , κάνοντας τους πιο πολύτιμους από ποτέ.

Αρχικά, θα μπορούσε κανείς εύλογα να πει πως η χρήση ενός σαρωτή είναι αρκετά φιλική προς το περιβάλλον. Τα ψηφιακά αντίγραφα που δημιουργούν μειώνουν σημαντικά την ποσότητα χαρτιού που χρησιμοποιείται.

Ταυτόχρονα, δίνεται η δυνατότητα σε πολλά άτομα από διαφορετικές τοποθεσίες να έχουν πρόσβαση και να βλέπουν το ίδιο ηλεκτρονικό έγγραφο ταυτόχρονα. Η ευκολία της αποστολής ηλεκτρονικών εγγράφων μέσω ηλεκτρονικού ταχυδρομείου διευκολύνει την πιο έγκαιρη επικοινωνία.

Παράλληλα, ελαττώνεται η συσσώρευση χαρτιών και εγγράφων, που πολλές φορές μπορεί να δημιουργήσει άσκοπο όγκο στο γραφείο. Πολλές φορές θεωρείται πιο εύκολο να οργανώσεις και να ευρετηριάσεις ψηφιακά αρχεία. Εκτός από την εξοικονόμηση χώρου που προσφέρουν, κάνουν και πολύ πιο εύκολη την προσβασιμότητα και την αναζήτηση τους.

Επίσης, είναι ένα πολύ σημαντικό εργαλείο για να διατηρηθούν διάφορα σημαντικά έγγραφα και εικόνες. Ένα αρχείο που έχει υποστεί σάρωση, απελευθερώνεται από τον κίνδυνο να αλλοιωθεί ή να καταστραφεί και επίσης δημιουργείται πιο εύκολα το ενδεχόμενο να βελτιωθεί η αναγνωσιμότητά του. Ακόμα και αν η φωτογραφία έχει ήδη υποστεί κάποια καταστροφή, είναι δυνατό να πραγματοποιηθεί η αποκατάστασή της μέσω κάποιου προγράμματος επεξεργασίας εικόνας. Για παράδειγμα, στην Εικόνα 3.3.1 φαίνεται το αποτέλεσμα σάρωσης μια κατεστραμμένης παλιάς φωτογραφίας που ύστερα υπέστη επεξεργασία και βελτιώθηκε σημαντικά. Αυτό φυσικά καθιστά τα scanners ένα πολύτιμο μέσο και για τη διατήρηση σημαντικών ιστορικών εγγράφων.



Εικόνα 3.3.1 Η χρήση του σαρωτή για επανόρθωση κατεστραμμένης εικόνας [33]

3.4 Χρήσεις του σαρωτή

Κάποιοι ενδεικτικοί λόγοι να χρησιμοποιήσει κάποιος έναν σαρωτή είναι οι εξής: [4]

- Δημιουργία αντιγράφων
- Ψηφιακή αρχειοθέτηση
- Συλλογή πληροφοριών από δανεισμένα βιβλία βιβλιοθήκης ή άλλες ιδιωτικές πηγές
- Εύκολο και γρήγορο μοίρασμα φωτογραφιών και αρχείων
- Επεξεργασία σαρωμένης εικόνας/έγγραφου
- Προστασία έντυπων πληροφοριών

Το scanner είναι μια συσκευή που χρησιμοποιείται καθημερινά τόσο για επαγγελματικούς λόγους, όσο και προσωπικούς. Στα γραφεία και στις επιχειρήσεις αποτελεί απαραίτητο

εργαλείο, όπως επίσης και σε όλους τους δημόσιους οργανισμούς, σχολεία και πανεπιστήμια. Πολλές φορές θεωρείται αναγκαίο μηχάνημα για επιστημονικές έρευνες και ιστορικές μελέτες. Τέλος, αποτελεί ένα πολύ συνηθισμένο εργαλείο για το σπίτι καθώς επεκτείνεται και σε πολλές οικιακές και καθημερινές χρήσεις.

ΚΕΦΑΛΑΙΟ 4

ΤΕΧΝΟΛΟΓΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

4.1 Εφαρμογές για κινητές συσκευές

4.1.1 Ορισμός

Ο γρήγορος ρυθμός εξέλιξης της σύγχρονης τεχνολογίας έχει αυξήσει παράλληλα και την ταχύτητα προόδου του προγραμματισμού στις κινητές συσκευές. Μια *εφαρμογή* για κινητά είναι ένα πρόγραμμα υπολογιστή ή μια εφαρμογή λογισμικού που έχει σχεδιαστεί για να λειτουργεί σε μια κινητή έξυπνη συσκευή όπως τηλέφωνο ή tablet. Οι εφαρμογές είναι συνήθως μικρές, μεμονωμένες μονάδες λογισμικού με περιορισμένη λειτουργία. [5]

4.1.2 Android λογισμικό

Το Android ορίζεται ως ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας. Ο σχεδιασμός του που αναπτύχθηκε από τη Google, επιτρέπει στους χρήστες να χειρίζονται τις κινητές συσκευές διαισθητικά, με κινήσεις των δακτύλων. Η Google χρησιμοποιεί επίσης λογισμικό Android σε τηλεοράσεις, αυτοκίνητα και ρολόγια χειρός, καθένα από τα οποία διαθέτει μια μοναδική διεπαφή χρήστη. [6]

Οι εφαρμογές, γράφονται χρησιμοποιώντας το Kit Ανάπτυξης Λογισμικού Android (SDK). Το SDK είναι ένα σύνολο εργαλείων και προγραμμάτων λογισμικού που περιλαμβάνουν μια σειρά από εργαλεία όπως βιβλιοθήκες, εγχειρίδια, δείγματα κώδικα, διεργασίες και “οδηγούς” που οι προγραμματιστές μπορούν να χρησιμοποιήσουν και να ενσωματώσουν στις δικές τους εφαρμογές. [7]

Το Google Play Store ορίζεται ως το επίσημο κατάστημα εφαρμογών για τα κινητά με Android λογισμικό, επιτρέποντας στους χρήστες να κατεβάζουν εφαρμογές που έχουν αναπτυχθεί με το SDK και δημοσιεύονται μέσω της Google. Το κατάστημα προσφέρει δωρεάν και επί πληρωμή εφαρμογές.



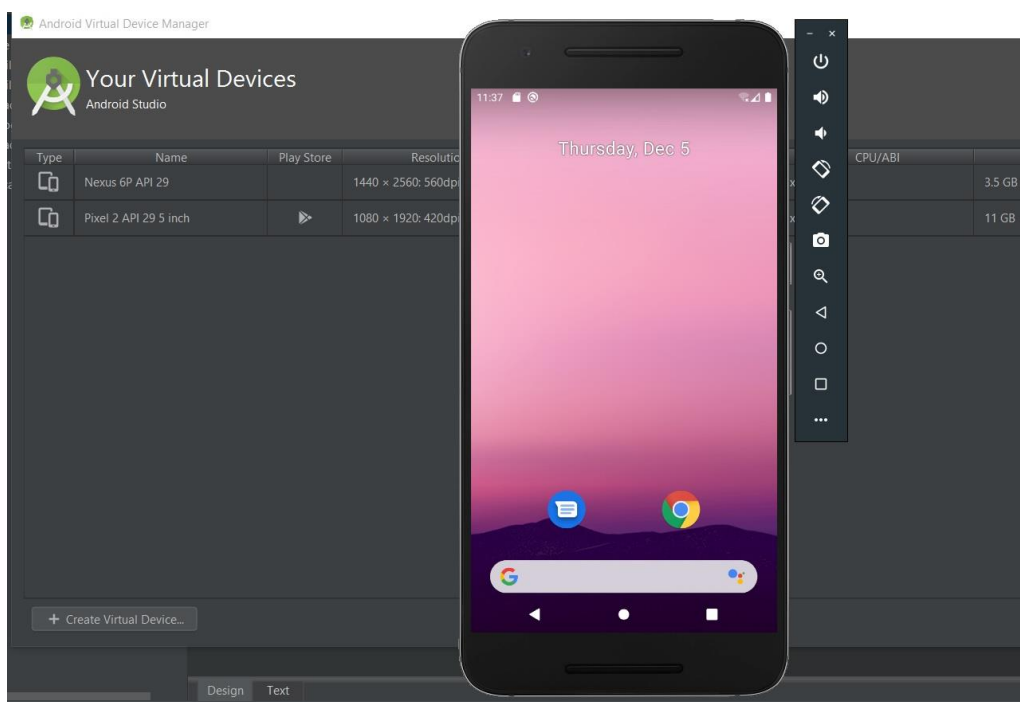
Εικόνα 4.1.2 Το λογότυπο της Google Play [34]

4.2 Android Studio

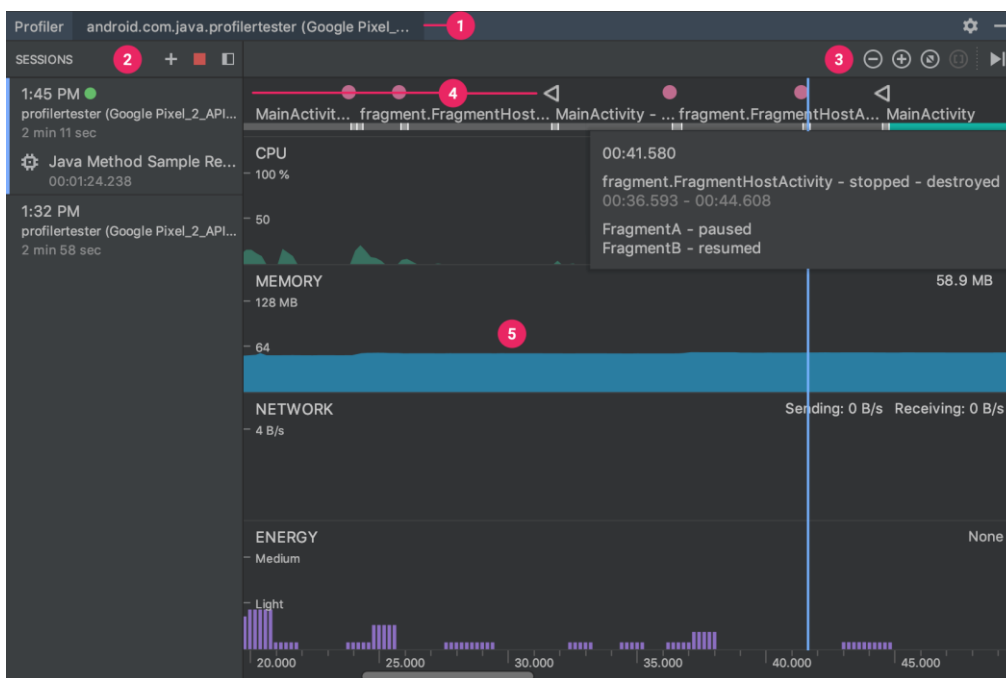
4.2.1 Ορισμός

Το Android Studio είναι το επίσημο Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE) για ανάπτυξη εφαρμογών και βασίζεται στο IntelliJ IDEA. Η ιδέα της κατασκευής του αρχικά έλαβε χώρα στις 16 Μαΐου 2013, ενώ η πρώτη σταθερή και ολοκληρωμένη υλοποίηση κυκλοφόρησε τον Δεκέμβριο του 2014, ξεκινώντας από την έκδοση 1.0. [8]

Το περιβάλλον αυτό προσφέρει πολλά χρήσιμα και πρακτικά εργαλεία για προγραμματιστές. Το Android Studio αποτελεί ένα περιβάλλον που επιτρέπει τον γρήγορο προγραμματισμό διευκολύνοντας τους χρήστες του με εξαιρετική ευελιξία ως προς την πραγματοποίηση μικρών αλλαγών στον κώδικα ενώ η εφαρμογή είναι ακόμα σε κατάσταση λειτουργίας. Παράλληλα, συνοδεύεται από ένα γρήγορο εξομοιωτή που μπορεί να προσομοιώσει λειτουργίες μιας φυσικής συσκευής και κάνει την δοκιμή της εφαρμογής εύκολα και αβίαστα. Αυτό είναι ένα πολύ σημαντικό βοήθημα για τον προγραμματιστή, αφού μπορεί να ελέγξει τη συμβατότητα της εφαρμογής του με διάφορα μοντέλα συσκευών χωρίς να τα έχει πραγματικά στην κατοχή του. (Εικόνα 4.2.1) Αξιοσημείωτο είναι επίσης ότι παρέχει ένα ισχυρό εργαλείο επεξεργασίας των αρχείων που αφορούν την διεπαφή του χρήστη με την εφαρμογή (XML αρχεία), κάνοντας την διαδικασία πολύ φιλική ως προς τον προγραμματιστή. Ακόμα, το Android Studio παρέχει πολλές λύσεις γρήγορης διόρθωσης που αφορούν την ασφάλεια και την απόδοση των εφαρμογών όπως επίσης και ένα ενσωματωμένο εργαλείο που εκτιμάει και παρέχει στατιστικά πραγματικού χρόνου που αφορούν τη CPU, τη μνήμη και τη δραστηριότητα της εφαρμογής. (Εικόνα 4.2.2) [9][10]



Εικόνα 4.2.1 Προβολή του εξομοιωτή που προσφέρει το Android Studio [10]



Εικόνα 4.2.2 Προβολή χρονοδιαγράμματος με στατιστικά στοιχεία πραγματικού χρόνου που αφορούν την συσκευή [10]

4.2.2 Προγραμματιστική γλώσσα Kotlin

Ενώ η προτεινόμενη γλώσσα προγραμματισμού για την σχεδίαση εφαρμογών Android ήταν η Java, τον Μάιο του 2019 αυτή αντικαταστάθηκε από την Kotlin. Έχει σχεδιαστεί για να συμβαδίζει πλήρως με την Java, αλλά ο συμπερασμός τύπων της (type inference) επιτρέπει τη σύνθεσή της να είναι πιο συνοπτική. [11] Η Kotlin έχει αποκτήσει το ενδιαφέρον των προγραμματιστών τα τελευταία τέσσερα χρόνια καθώς τα ισχυρά χαρακτηριστικά που προσφέρει καθιστούν τον χειρισμό της εύχρηστο και τον κώδικα σύντομο και ευανάγνωστο.

Κάποιοι ενδεικτικοί λόγοι να χρησιμοποιήσει κανείς Kotlin είναι οι εξής: [12] [13]

- Συνοπτικός αλλά παράλληλα εύκολος στην ανάγνωση κώδικας
- Εύκολη μετάφραση άλλων προγραμματιστικών γλωσσών σε αυτή
- Είναι πλήρως συμβατή με την Java
- Εξοικονομεί χρόνο λόγω της διαισθητικής σύνταξης και του συνολικού σχεδιασμού της
- Η στάνταρ βιβλιοθήκη της είναι μικρή και δεν επιφέρει μεγάλη επιβάρυνση χρόνου εκτέλεσης
- Προσφέρει μεγάλη ασφάλεια καθώς αποτρέπει από το να συμβούν πολλά κοινά λάθη που πραγματοποιούνται κατά την εκτέλεση του προγράμματος
- Απλοποιεί τον ασύγχρονο προγραμματισμό καθιστώντας κοινές διεργασίες όπως κλήσεις δικτύου και ενημερώσεις βάσεων δεδομένων απλές και αποδοτικές

- Είναι πολύ εύκολο να κάνεις αλλαγές στον κώδικα
- Συνοδεύεται από έναν έξυπνο μεταγλωττιστή που ανιχνεύει σφάλματα κατά τη μεταγλώττιση και όχι κατά τη διάρκεια του χρόνου εκτέλεσης



```

1 //file.kt
2 fun main(args: Array<String>) {
3     println("Hello, World!")
4 }

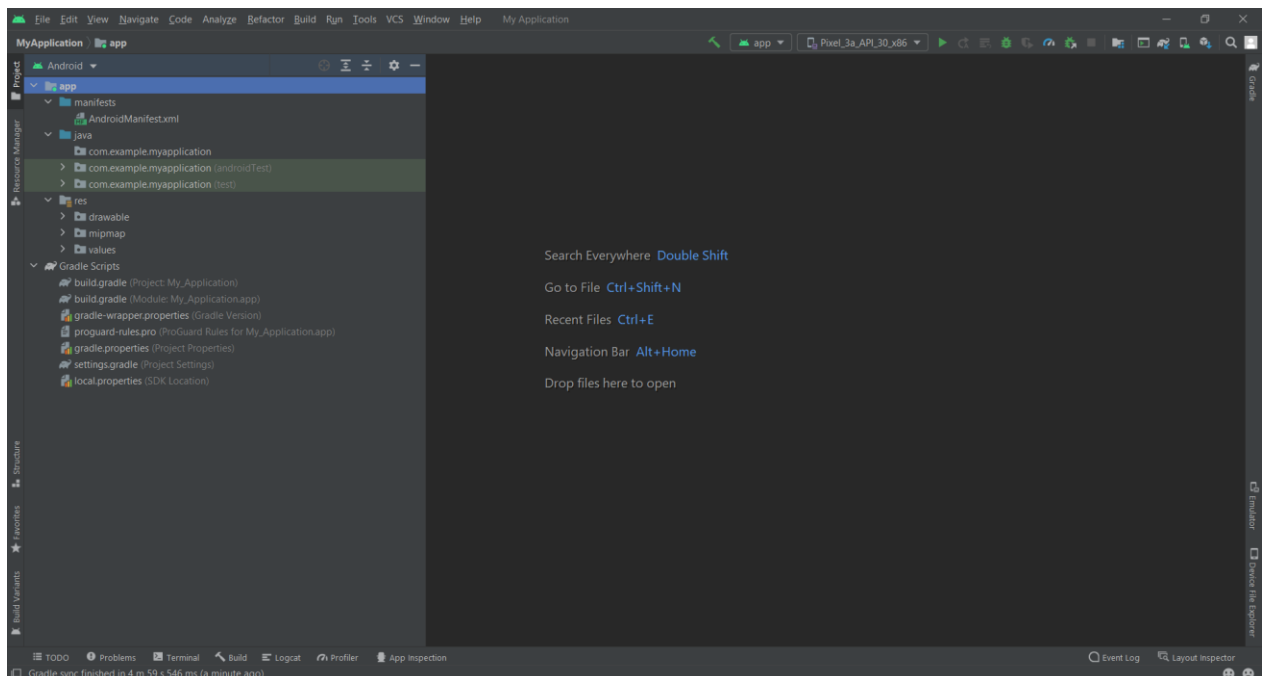
```

Εικόνα 4.2.3 Η χαρακτηριστική συνάρτηση “Hello world” σε γλώσσα Kotlin [34]

4.3 Το περιβάλλον του Android Studio

4.3.1 Εισαγωγή

Παρακάτω θα αναλυθεί το περιβάλλον του Android Studio για να κατανοηθεί περεταίρω η λειτουργία του και οι υπηρεσίες που προσφέρει.

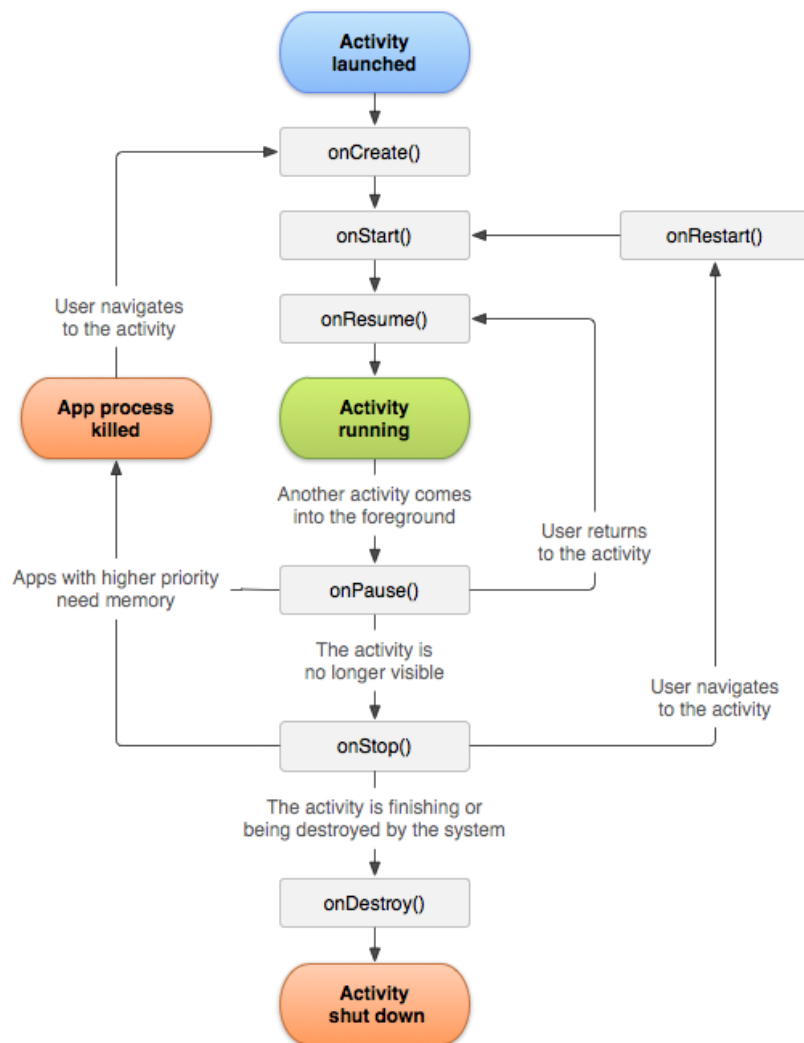


Εικόνα 4.3.1 Η αρχική οθόνη της πλατφόρμας Android Studio

4.3.2 Activities

Η κλάση Activity έχει απαραίτητο και βασικό ρόλο σε μια εφαρμογή Android και ο τρόπος που συνδυάζονται οι δραστηριότητες αυτές αποτελεί θεμελιώδες μέρος του μοντέλου εφαρμογής της πλατφόρμας.

Γενικά, ένα Activity αντιστοιχεί μία ενέργεια σε κάθε οθόνη που βλέπει ο χρήστης. Οι περισσότερες εφαρμογές περιέχουν πολλές οθόνες και κατά συνέπεια πολλαπλά Activities. Συνήθως, σε μια εφαρμογή ορίζεται η “κύρια δραστηριότητα”, η οποία είναι η πρώτη οθόνη που εμφανίζεται όταν ο χρήστης εκκινήσει την εφαρμογή. Κάθε Activity μπορεί στη συνέχεια να ξεκινήσει ένα άλλο Activity προκειμένου να εκτελέσει διαφορετικές ενέργειες. [14] Για παράδειγμα, σε μια εφαρμογή που χρησιμοποιεί κάμερα, ένα Activity μπορεί να είναι το τράβηγμα της φωτογραφίας ή οποία στη συνέχεια θα ξεκινήσει μια άλλη δραστηριότητα που πιθανώς να είναι η επισκόπηση της φωτογραφίας που τραβήχτηκε.



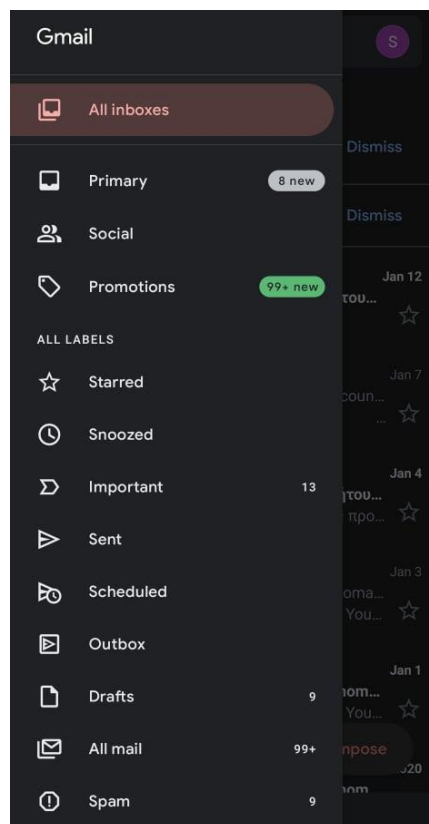
Εικόνα 4.3.2 Ο κύκλος ζωής ενός Activity [15]

Κάθε κλάση Activity χαρακτηρίζεται από έναν συγκεκριμένο κύκλο ζωής η διαχείριση του οποίου είναι πολύ σημαντική για την δημιουργία μιας ολοκληρωμένης εφαρμογής. Υπάρχουν τέσσερις καταστάσεις ενός Activity: [15]

- Ενεργή κατάσταση (συνήθως συμβαίνει όταν ο χρήστης αλληλεπιδράει μαζί του)
- Ορατή κατάσταση (εξακολουθεί να παρουσιάζεται στον χρήστη αλλά αυτός εστιάζει σε ένα άλλο)
- Σε παύση (έχει διακοπεί αλλά είναι ακόμα ορατό, αν όμως χρειαστεί μήμη θα διαγραφεί)
- Σταματημένη κατάσταση (έχει καταστραφεί είτε από το σύστημα είτε από τον ίδιο τον χρήστη)

4.3.3 Fragments

Το Fragment αντιπροσωπεύει ένα επαναχρησιμοποιήσιμο τμήμα της διεπαφής του χρήστη σε ένα Activity. Καθένα έχει τον δικό του κύκλο ζωής και μπορεί να προστεθεί ή να αφαιρεθεί ενώ εκτελείται η δραστηριότητα. Ένα Fragment δεν μπορεί να ζήσει από μόνο του, καθώς πρέπει να φιλοξενείται είτε από ένα Activity είτε από ένα άλλο Fragment. [16] Ο κύκλος ζωής του σχετίζεται στενά με τον κύκλο ζωής του Activity που τον φιλοξενεί, δηλαδή όταν σταματήσει η δραστηριότητα, όλα τα Fragments που περιλαμβάνονται στη δραστηριότητα θα σταματήσουν επίσης.



Εικόνα 4.3.3 Παράδειγμα με Fragments στην εφαρμογή Gmail

Για παράδειγμα, Στην εφαρμογή Gmail κάθε φορά που ο χρήστης πατάει κάτι από την αριστερή λίστα, εμφανίζεται μπροστά του ένα Fragment, που ανήκει όμως στο ίδιο Activity με όλα τα υπόλοιπα. (Εικόνα 4.3.3)

4.3.4 Layout- Αρχεία XML

Το Layout της εφαρμογής είναι ουσιαστικά το σχέδιο που καθορίζει τη δομή για τη διεπαφή του χρήστη με αυτή. Όλα τα στοιχεία στο Layout είναι χτισμένα χρησιμοποιώντας μια ιεραρχία αντικειμένων View και ViewGroup. Το View είναι υπεύθυνο για το σχεδιασμό συμβάντων μέσα από την ικανότητα του να χειρίζεται διαδραστικά και γραφικά στοιχεία όπως κουμπιά, πεδία κειμένου κ.τ.λ. Το ViewGroup ορίζεται ως μια ομάδα που περιλαμβάνει άλλα Views και καθορίζει τις ιδιότητες τους στο Layout. Κάθε Activity περιέχει πολλά στοιχεία View και ViewGroup χτισμένα χρησιμοποιώντας μια ιεραρχία αντικειμένων. [17]

Το Layout της κάθε Activity είναι γραμμένο σε eXtensible Markup Language, γνωστή και ως XML. Η XML είναι μια γλώσσα σήμανσης για την ηλεκτρονική κωδικοποίηση κειμένων. Κάθε Activity και Fragment αντιστοιχούνται σε ένα XML αρχείο το οποίο είναι υπεύθυνο για το πως θα παρουσιάζεται στο μάτι του χρήστη η εφαρμογή.

Κάθε View και ViewGroup έχει κάποια XML χαρακτηριστικά όπως ID (“χαρακτηριστικό” ψευδώνυμο), ύψος, φάρδος, μέγεθος, χρώμα κ.ο.κ. οπού ο προγραμματιστής εφαρμόζει ανάλογα ώστε να ή τελική εφαρμογή να είναι φιλική προς τον χρήστη.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Εικόνα 4.3.4 Παράδειγμα κώδικα γραμμένο σε XML που περιλαμβάνει μια ιεραρχία από Views και ViewGroup [17]

4.3.5 Android Manifest

Κάθε εφαρμογή πρέπει να έχει ένα αρχείο που να ονομάζεται ακριβώς AndroidManifest.xml στη ρίζα του κώδικα του πρότζεκτ. Το αρχείο αυτό περιγράφει βασικές πληροφορίες που σχετίζονται με την εφαρμογή.

Στο Manifest είναι απαραίτητο να έχουν δηλωθεί τα εξής:

- Το όνομα πακέτου της εφαρμογής
- Τα μέρη της εφαρμογής όπως τα Activities και άλλες υπηρεσίες
- Τα δικαιώματα που χρειάζεται η εφαρμογή σε περίπτωση που χρειάζεται πρόσβαση σε προστατευμένα τμήματα της συσκευής που την τρέχει
- Τις δυνατότητες υλικού και λογισμικού που απαιτεί η εφαρμογή, οι οποίες επηρεάζουν ποιες συσκευές μπορούν να εγκαταστήσουν την εφαρμογή από το Google Play

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication"
    android:versionCode="1"
    android:versionName="1.0" >
    ...
</manifest>
```

Εικόνα 4.3.5 Παράδειγμα κώδικα γραμμένο σε XML που περιλαμβάνεται στο αρχείο AndroidManifest.xml [18]

Το Android Studio αναλαμβάνει μόνο του την δημιουργία του αρχείου Manifest και τα περισσότερα από τα απαραίτητα στοιχεία συμπληρώνονται αυτόματα. [18]

4.3.6 Gradle

Το σύστημα δημιουργίας Android (Android build system) είναι υπεύθυνο για τη μεταγλώττιση του κώδικα και τη μετατροπή του σε πακέτο εφαρμογής Android (APK). Το Android Studio χρησιμοποιεί το Gradle, μια προηγμένη εργαλειοθήκη που διαχειρίζεται την παραπάνω διαδικασία ενώ επιτρέπει στον προγραμματιστή να διαμορφώσει και να προσαρμόσει το build system. [19]

4.4 Τύποι αρχείων

4.4.1 Εικόνα JPG (ή JPEG)

Το JPG ή JPEG είναι μια κοινώς χρησιμοποιούμενη μέθοδος συμπίεσης ψηφιακής εικόνας, ιδιαίτερα για εικόνες που παράγονται από ψηφιακή φωτογραφική μηχανή. Το JPG επιτυγχάνει συνήθως συμπίεση 10: 1 με μικρή αισθητή απώλεια στην ποιότητα της εικόνας. Είναι το πιο διαδεδομένο πρότυπο συμπίεσης εικόνας στον κόσμο καθώς και η πιο γνωστή μορφή ψηφιακής εικόνας. Τα αρχεία αυτά έχουν συνήθως επέκταση ονόματος αρχείου .jpg ή .jpeg.

Ο όρος "JPG" είναι το ακρωνύμιο για την Joint Photographic Experts Group, η οποία δημιούργησε το πρότυπο το 1992. Ο αλγόριθμος συμπίεσης λειτουργεί καλύτερα σε φωτογραφίες και ζωγραφιές με ομαλές παραλλαγές τόνου και χρώματος. [20]

4.4.2 Έγγραφο PDF

Το PDF (Portable Document Format), είναι μια μορφή αρχείου που αναπτύχθηκε από την Adobe το 1993 για την παρουσίαση εγγράφων. Κάθε αρχείο PDF εμπεριέχει την πλήρη περιγραφή ενός εγγράφου συμπεριλαμβανομένου του κειμένου, των γραμματοσειρών, των γραφικών, των εικόνων, διάφορων διαδραστικών στοιχείων και άλλων πληροφοριών που απαιτούνται για την εμφάνισή του. Το PDF περιλαμβάνει ένα υποσύνολο της γλώσσας προγραμματισμού PostScript, για τη δημιουργία της εμφάνισης του εγγράφου και των γραφικών. [21]

Υπάρχουν πολλοί λόγοι να προτιμήσει κανείς αυτή τη μορφή αρχείων:

- Είναι ένα πρότυπο που χρησιμοποιείται σχεδόν από κάθε βιομηχανία και μπορεί να ανοίξει σε κάθε σύγχρονη συσκευή
- Το PDF διασφαλίζει ότι τα αρχεία που εξάγονται έχουν μορφοποιηθεί σωστά και θα φαίνονται το ίδιο, ανεξάρτητα από το πρόγραμμα το οποίο τα δημιούργησε
- Τα έγγραφα PDF έχουν τη δυνατότητα να διατηρούν υψηλή ποιότητα κρατώντας το μέγεθος του αρχείου χαμηλό, καθιστώντας το PDF την ιδανική μορφή για αποστολή αρχείων

4.5 Έλεγχος εκδόσεων (version control)

4.5.1 Χρησιμότητα

Σήμερα στη βιομηχανία τόσο ο όγκος όσο και η πολυπλοκότητα του κώδικα που δημιουργείται αυξάνονται εκθετικά. Στην προσπάθεια να διαχειριστούν όσο πιο αποδοτικά γίνεται οι μεγάλοι αυτοί όγκοι κώδικα δημιουργήθηκαν διάφορα λογισμικά ελέγχου εκδόσεων.

Ένα τέτοιο λογισμικό επιτρέπει στον προγραμματιστή και την ομάδα του να έχουν μία κοινή διαδικτυακή βάση με όλο το δημιούργημά τους, να την κρατούν ενημερωμένη σε σχέση με τις αλλαγές που πραγματοποιούνται στα διάφορα σημεία του εγχειρήματος αλλά και ίσως σημαντικότερα να έχουν πρόσβαση από οποιοδήποτε υπολογιστή, αρκεί να είναι συνδεδεμένος στο ίντερνετ.

Στην περίπτωση της παρούσας εργασίας χρησιμοποιήθηκε ο έλεγχος εκδόσεων για να διευκολύνει την πρόσβαση στα δεδομένα από τους διάφορους υπολογιστές που χρησιμοποιήθηκαν, για να καθίσταται δυνατή η δοκιμή της εφαρμογής σε διάφορες πραγματικές κινητές συσκευές και έτσι να προληφθούν προβλήματα που έχουν να κάνουν με τις διάφορες λεπτομέρειες των εκδόσεων Android αλλά και επειδή είναι σωστή πρακτική για οποιοδήποτε τρίτο ενδιαφερόμενο πρόσωπο θα θελήσει να εμβαθύνει στον κώδικα.

4.5.2 Git

Κατά την δημιουργία του ImageScanner θεωρήθηκε ωφέλιμο να αποθηκεύεται η πρόοδος σε ένα σύστημα ελέγχου εκδόσεων. Το σύστημα που χρησιμοποιήθηκε είναι το γνωστό git.

Το git είναι το πιο διάσημο σύστημα ελέγχου εκδόσεων και σε χρησιμοποιείται σε συνδυασμό με το Github που θα αναλυθεί στη συνέχεια.

Το git δουλεύει τοπικά σε κάθε υπολογιστή και δεν έχει από μόνο του εξάρτηση από το ίντερνετ. Αρχικά δημιουργεί ένα φάκελο στο σύστημα αρχείων του υπολογιστή και το αρχικοποιεί κατάλληλα. Αφού δημιουργηθούν τα απαραίτητα αρχεία για τη διατήρηση του ιστορικού της εργασίας (.git, .gitignore κλπ.) ο προγραμματιστής μπορεί να αρχίσει να προσθέτει τα δικά του αρχεία.

Μέσω μίας διαδικασίας που αποτελείται σε γενικές γραμμές από τις πιο απλές εντολές που προσφέρει το git ο προγραμματιστής μπορεί να ενημερώνει την κατάσταση του κώδικα κρατώντας πάντα ένα εκτενές και ακριβές ιστορικό της κατάστασης που βρισκόταν το πρόγραμμα σε όλες τις παρελθοντικές στιγμές. Οι εντολές αυτές είναι:

- Git add <ΑΡΧΕΙΟ>
- Git commit <ΑΡΧΕΙΟ> -m "ΣΧΟΛΙΑΣΜΟΣ"
- Git status

Η πρώτη εντολή “μαρκάρει” ένα αρχείο για το σύστημα διαχείρισης δείχνοντας πως αυτό το αρχείο θα πρέπει να περιληφθεί στην επόμενη έκδοση του κώδικα. Ο προγραμματιστής οφείλει να αναφέρει στο git όλα τα αρχεία που έχουν υποστεί αλλαγές που θέλει να περαστούν στο σύστημα. Να σημειωθεί ότι δε χρειάζεται να προστεθούν όλα τα αλλαγμένα αρχεία σε μία έκδοση του κώδικα.

Αφού όλα τα αλλαγμένα αρχεία έχουν προστεθεί με την εντολή git add τότε εκτελείται η εντολή git commit -m "". Η εντολή αυτή διακρίνει τις αλλαγές στα αρχεία και είναι υπεύθυνη να τις συνενώσει τις δύο εκδόσεις που υπάρχουν αυτή τη στιγμή στο δίσκο. Η εντολή commit απαιτεί ο χρήστης να δώσει μαζί με το commit και ένα σχόλιο που να περιγράφει τις αλλαγές που πραγματοποιήθηκαν. Αυτό είναι κομβικής σημασίας τόσο επειδή κάνει την επιστροφή σε σημεία της εξέλιξης του εγχειρήματος πιο κατανοητή - πόσο περισσότερο μάλλον σε ένα πρόγραμμα με πολλούς προγραμματιστές, όσο και επειδή δείχνει τον προσανατολισμό του git προς τη σαφήνεια και ευκολία κατανόησης της πορείας που έχει πάρει η εκάστοτε εφαρμογή.

4.5.3 Github

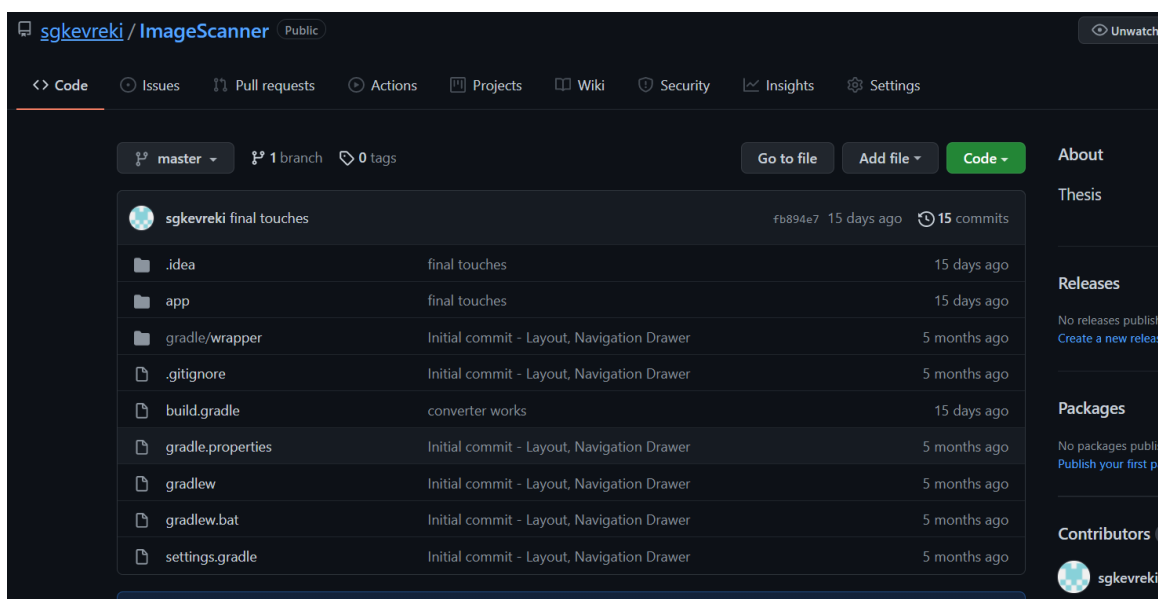
Όσο σημαντικό είναι το git για τη σωστή αποθήκευση του κώδικα, τόσο σημαντικό είναι το Github για την εύκολη και γρήγορη πρόσβαση σε αυτόν.

Το Github είναι πλήρως συμβατό με το git και επιτρέπει την εναπόθεση του κώδικα σε προκαθορισμένο χώρο στο διαδίκτυο. Σε συνέχεια των παραπάνω το git διαθέτει τις εξής εντολές για την αλληλεπίδραση με τον απομακρυσμένο χώρο αποθήκευσης:

- Git clone
- Git push
- Git pull

Το git clone χρησιμοποιείται για το κατέβασμα και τη σωστή διαμόρφωση του κώδικα σε ένα νέο φάκελο τοπικά στον υπολογιστή. Από εκεί και πέρα ακολουθείται η διαδικασία που περιγράφηκε παραπάνω. Όταν ο προγραμματιστής αλλάξει πράγματα στον κώδικα, αυτά αποθηκεύονται αρχικά πάντα τοπικά. Μόλις θελήσει να ενημερώσει την απομακρυσμένη βάση για τις εντολές, τότε εκτελεί τη δεύτερη εντολή git push. Από την άλλη πλευρά αντίστοιχα, όταν θελήσει να κατεβάσει τη νέα έκδοση κώδικα από την απομακρυσμένη βάση τοπικά εκτελεί git pull.

Κατά κανόνα οι εντολές αυτές δίνονται από ένα τερματικό στον υπολογιστή. Το περιβάλλον του Android Studio παρόλα αυτά διαθέτει κάποιες ενσωματωμένες λειτουργίες που κάνουν αυτή την απλή διαδικασία ακόμα ευκολότερη.



Εικόνα 4.5.1 Η αρχική οθόνη του πρότζεκ στο Github

ΚΕΦΑΛΑΙΟ 5

ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

5.1 Εισαγωγή

Η επεξεργασία εικόνας ορίζεται ως η χρήση ψηφιακού υπολογιστή για την επεξεργασία ψηφιακών εικόνων μέσω αλγορίθμων. Η ψηφιακή επεξεργασία εικόνας επιτρέπει την εφαρμογή μεγάλου εύρους αλγορίθμων στα δεδομένα εισόδου και μπορεί να αποφύγει προβλήματα όπως η συσσώρευση θορύβου και η παραμόρφωση. [22]

Η δημιουργία της ψηφιακής επεξεργασίας εικόνας, αλλά και η ανάπτυξη της, σχετίζονται άμεσα με την ανάπτυξη των υπολογιστών και των μαθηματικών και είναι σημαντικό βοήθημα για ένα μεγάλο φάσμα εφαρμογών όπως το περιβάλλον, τη γεωργία, τη βιομηχανία, την ιατρική επιστήμη κ.ο.κ.

Στην παρούσα διπλωματική εργασία εφαρμόστηκαν δύο βασικές τεχνικές που αφορούν την επεξεργασία εικόνας στην εφαρμογή.

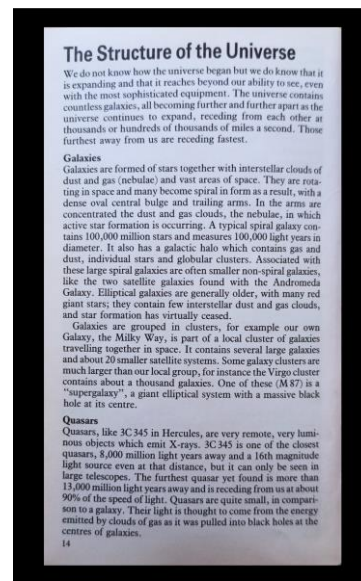
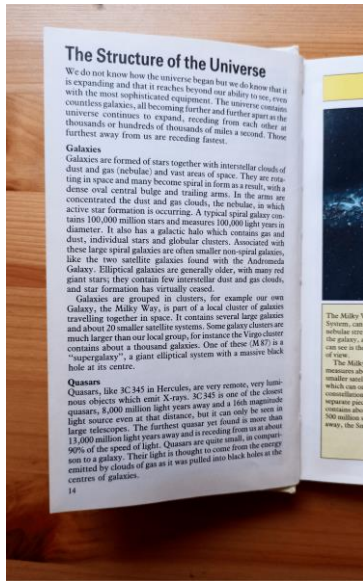
5.2 Αλγόριθμος έξυπνης περικοπής

5.2.1 Εισαγωγή

Ο αλγόριθμος έξυπνης περικοπής είναι η διαδικασία κατά την οποία γίνεται αυτόματα και με επιθυμητό τρόπο η περικοπή μιας φωτογραφίας που συνήθως απεικονίζει ένα κομμάτι χαρτί ή κάποιο πλαίσιο. Ο αλγόριθμος υλοποιείται με σκοπό η τελική εικόνα να μην περιέχει αχρείαστα κενά και δυσαναλογίες, αλλά το αποτέλεσμα να είναι ευανάγνωστο και εύκολο στο μάτι. Για παράδειγμα, η έξυπνη περικοπή μπορεί να εφαρμοστεί σε έγγραφα, αφίσες, ταυτότητες, αποδείξεις, σελίδες βιβλίων, εκτυπωμένες φωτογραφίες.

5.2.2 Ανίχνευση άκρων (Edge Detection)

Η ανίχνευση άκρων περιλαμβάνει μια ποικιλία μαθηματικών μεθόδων που στοχεύουν στον εντοπισμό σημείων σε μια ψηφιακή εικόνα στα οποία η φωτεινότητα της εικόνας αλλάζει απότομα ή έχει ασυνέχειες. Είναι ένα θεμελιώδες εργαλείο στην επεξεργασία εικόνας. Στην ιδανική περίπτωση, το αποτέλεσμα εφαρμογής ενός τέτοιου αλγόριθμου σε μια εικόνα μπορεί να οδηγήσει σε ένα σύνολο συνδεδεμένων καμπυλών που σχηματίζουν τα όρια των αντικειμένων που απεικονίζονται. Για παράδειγμα, ένα τυπικό άκρο σε μια εικόνα μπορεί να είναι το όριο μεταξύ ενός μπλοκ κόκκινου χρώματος και ενός κίτρινου. [23]



Εικόνα 5.2.1 (α, β, γ): Η διαδικασία της έξυπνης περικοπής

Οι αλγόριθμοι για την ανίχνευση άκρων περιλαμβάνουν τρία βήματα:

- I. Φιλτράρισμα (Filtering): Το φιλτράρισμα χρησιμοποιείται συνήθως για τη βελτίωση της απόδοσης σε σχέση με το θόρυβο
- II. Ενίσχυση (Enhancement): Η ενίσχυση δίνει έμφαση στα γειτονικά πίκσελς ενός σημείου στα οποία υπάρχει σημαντική αλλαγή στις τιμές έντασης
- III. Ανίχνευση (Detecting): Υπάρχουν πολλές μέθοδοι για την ανίχνευση άκρων, αλλά οι περισσότερες από αυτές μπορούν να ομαδοποιηθούν σε δύο κατηγορίες: αναζήτησης και μηδενικής διασταύρωσης

5.2.3 Περικοπή εικόνας

Η περικοπή είναι η αφαίρεση ανεπιθύμητων εξωτερικών περιοχών από μια εικόνα. Η διαδικασία περιλαμβάνει αλλαγή του μεγέθους και των διαστάσεων της φωτογραφίας και συνήθως εφαρμόζεται όταν η φωτογραφία περιέχει περιττές πληροφορίες και χρειάζεται καλύτερο “καδράρισμα”.

Όταν συνδυάζεται με την τεχνική ανίχνευσης ακρών, δημιουργείται μια νέα εικόνα με άκρα τα σημεία που ανιχνεύτηκαν και η τελική εικόνα έχει εντελώς διαφορετικές διαστάσεις, ανάλογα με το επιθυμητό αποτέλεσμα. Όπως φαίνεται και στο σχήμα, τα άκρα της φωτογραφίας ανιχνεύτηκαν στα σημεία που υπήρχε ασυνέχεια στη φωτεινότητα και δημιούργησαν ένα πλαίσιο (Εικόνα 5.2.1 (β)), το οποίο στη συνέχεια αποτέλεσε τις νέες διαστάσεις της φωτογραφίας όταν έγινε η περικοπή. (Εικόνα 5.2.1 (γ))

5.3 Βελτιστοποίηση Εικόνας

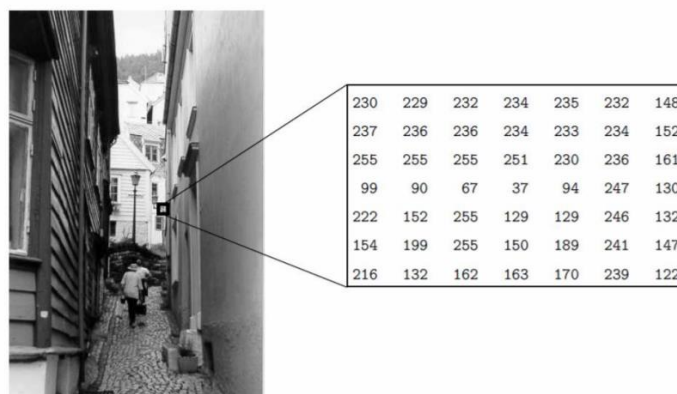
5.3.1 Εισαγωγή

Στην επεξεργασία εικόνας, πολλές φορές εφαρμόζονται φίλτρα που ανάλογα με τον σκοπό τους, αλλάζουν το μέγεθος, τα χρώματα τη σκίαση και άλλα χαρακτηριστικά της. Συνήθως, όταν εφαρμόζεται ένα φίλτρο αλλάζουν οι τιμές των πίξελς της εικόνας και χρησιμοποιείται για την επεξεργασία της αντίθεσης, της φωτεινότητας, της υφής των χρωμάτων κ.ο.κ

5.3.2 Κατωφλίωση (thresholding) και μονοχρωματισμός (Grayscale)

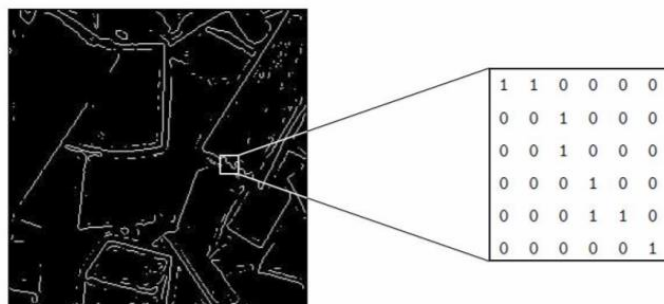
Στην ψηφιακή επεξεργασία εικόνας, η κατωφλίωση είναι η πιο απλή μέθοδος κατακερματισμού, δηλαδή της διαδικασίας διαίρεσης μιας ψηφιακής εικόνας σε πολλά τμήματα (σύνολα πίξελ). Αυτή η διαδικασία μπορεί να βελτιώσει την πληροφορία της εικόνας για είναι πιο λεπτομερής, ευκρινής και καθαρή και να φαίνεται καλύτερη στο ανθρώπινο μάτι.

Οι μονοχρωματικές εικόνες αποτελούνται αποκλειστικά από γκρι αποχρώσεις. Κάθε πίξελ παίρνει μια τιμή γκριζου η οποία κυμαίνεται από μαύρο με την πιο αδύναμη τιμή έντασης (0) έως λευκό με την ισχυρότερη τιμή έντασης (255). (Εικόνα 5.3.1)



Εικόνα 5.3.1 Οι τιμές των πίξελ σε μια μονοχρωματική εικόνα [35]

Αντίθετα, οι δυαδικές εικόνες είναι εικόνες που οι τιμές των πίξελ παίρνουν μόνο τις τιμές 0 και 1.



Εικόνα 5.3.2 Οι τιμές των πίξελ σε μια δυαδική εικόνα [35]

Στην κατωφλίωση, μετατρέπουμε μια μονοχρωματική εικόνα με κλίμακα του γκρι σε δυαδική εικόνα. Στη μέθοδο αυτή αντικαθίσταται κάθε πίξελ από μια εικόνα με ένα μαύρο πίξελ εάν η ένταση της εικόνας είναι μικρότερη από κάποια σταθερή τιμή, ή ένα λευκό ρixel εάν η ένταση της εικόνας είναι μεγαλύτερη από αυτή τη σταθερά. Αυτή η σταθερά ονομάζεται κατώφλι.

5.3.3 Αφαίρεση θορύβων (noise removal)

Όλες οι συσκευές επεξεργασίας σήματος, τόσο αναλογικές όσο και ψηφιακές, έχουν χαρακτηριστικά που τις καθιστούν ευαίσθητες στο θόρυβο. Ο θόρυβος μπορεί να είναι τυχαίος ή λευκός (white noise) με ομοιόμορφη κατανομή συχνότητας ή να είναι εξαρτώμενος από τη συχνότητα που εισάγεται στο μηχανισμό μιας συσκευής. [24] Η αφαίρεση θορύβου είναι μια μέθοδος που μειώνει τον ανεπιθύμητο θόρυβο σε μια εικόνα και έτσι την κάνει πιο “καθαρή” και βελτιωμένη οπτικά.

Για παράδειγμα, ένας αλγόριθμος αφαίρεσης θορύβου αντικαθιστά τις τιμές κάποιων πίξελ μιας εικόνας σε μαύρο ή άσπρο. Τα πίξελ που οι τιμές του κόκκινου, μπλε και πράσινου δεν ξεπερνάνε ταυτόχρονα μια συγκεκριμένη τιμή γίνονται μαύρα, ενώ άμα την ξεπερνάνε γίνονται άσπρα.

5.3.4 Παράδειγμα

Στην Εικόνα 5.3.3 φαίνεται το αποτέλεσμα της επεξεργασίας μιας φωτογραφίας που απεικονίζει μια σελίδα από βιβλίο. Αρχικά, η εικόνα μετατράπηκε σε μονοχρωματική με κλίμακα του γκρι και έπειτα μέσω της μεθόδου της κατωφλίωσης σε δυαδική. Τέλος, χρησιμοποιήθηκε το φίλτρο αφαίρεσης θορύβου για να γίνει πιο ευανάγνωστη. Το τελικό αποτέλεσμα σκοπεύει στο να γίνει η εικόνα πιο αναγνώσιμη και “καθαρή”.

The Structure of the Universe

We do not know how the universe began but we do know that it is expanding and that it reaches beyond our ability to see, even with the most sophisticated equipment. The universe contains countless galaxies, all becoming further and further apart as the universe continues to expand, receding from each other at thousands or hundreds of thousands of miles a second. Those furthest away from us are receding fastest.

Galaxies

Galaxies are formed of stars together with interstellar clouds of dust and gas (nebulae) and vast areas of space. They are rotating in space and many become spiral in form as a result, with a dense oval central bulge and trailing arms. In the arms are concentrated the dust and gas clouds, the nebulae, in which active star formation is occurring. A typical spiral galaxy contains 100,000 million stars and measures 100,000 light years in diameter. It also has a galactic halo which contains gas and dust, individual stars and globular clusters. Associated with these large spiral galaxies are often smaller non-spiral galaxies, like the two satellite galaxies found with the Andromeda Galaxy. Elliptical galaxies are generally older, with many red giant stars; they contain few interstellar dust and gas clouds, and star formation has virtually ceased.

Galaxies are grouped in clusters, for example our own Galaxy, the Milky Way, is part of a local cluster of galaxies travelling together in space. It contains several large galaxies and about 20 smaller satellite systems. Some galaxy clusters are much larger than our local group, for instance the Virgo cluster contains about a thousand galaxies. One of these (M87) is a "supergalaxy", a giant elliptical system with a massive black hole at its centre.

Quasars

Quasars, like 3C 345 in Hercules, are very remote, very luminous objects which emit X-rays. 3C 345 is one of the closest quasars, 8,000 million light years away and a 16th magnitude light source even at that distance, but it can only be seen in large telescopes. The furthest quasar yet found is more than 13,000 million light years away and is receding from us at about 90% of the speed of light. Quasars are quite small, in comparison to a galaxy. Their light is thought to come from the energy emitted by clouds of gas as it was pulled into black holes at the centres of galaxies.

14

Εικόνα 5.3.3 Αποτέλεσμα επεξεργασίας εικόνας μετά από κατωφλίωση (μονοχρωματική σε δυαδική εικόνα) και αφαίρεση θορύβου

ΚΕΦΑΛΑΙΟ 6

ΤΟ IMAGE SCANNER ΑΝΑΛΥΤΙΚΑ

6.1 Εισαγωγή

Στα προηγούμενα κεφάλαια αναλύθηκαν οι λόγοι για τους οποίους είναι σημαντική η χρήση του σαρωτή τη σημερινή εποχή, το μεγάλο ρόλο που παίζουν οι εφαρμογές στην καθημερινότητα, τους λόγους να χρησιμοποιήσει κανείς προηγμένα λειτουργικά περιβάλλοντα όπως το Android Studio καθώς και τη σημαντικότητα αλγορίθμων που συμβάλουν στη βελτιστοποίηση εικόνας. Με βάση τα παραπάνω, δημιουργήθηκε μια εφαρμογή για κινητές Android συσκευές που λειτουργεί σαν ένας σαρωτής εγγράφων, η λειτουργικότητα της οποίας θα αναλυθεί στο συγκεκριμένο κεφάλαιο.



Εικόνα 6.1.1 Το λογότυπο του σχεδιάστηκε για το Image Scanner

6.2 Η εμπειρία χρήσης

6.2.1 Υπηρεσίες εφαρμογής

Η εφαρμογή Image Scanner είναι ένα εργαλείο με το οποίο ο χρήστης μπορεί να σαρώσει έγγραφα, χειρόγραφα ή εκτυπωμένα κείμενα, αποδείξεις, σελίδες από βιβλία ή τετράδια κ.ο.κ.

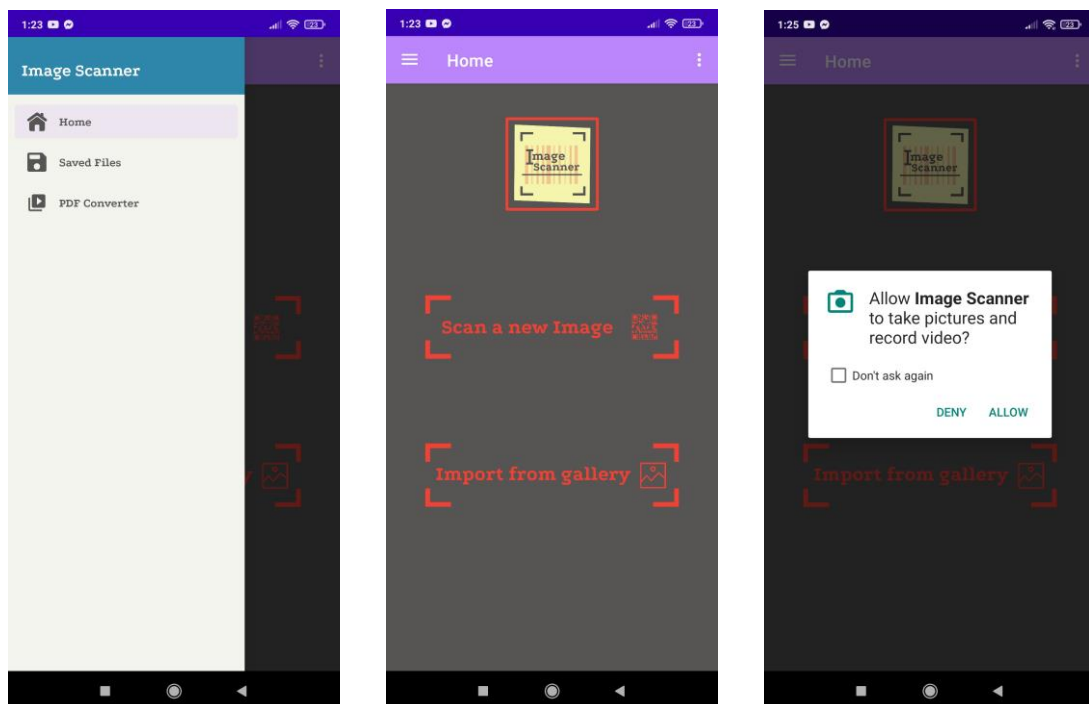
Οι υπηρεσίες που προσφέρει στον χρήστη είναι οι εξής:

- Τράβηγμα φωτογραφίας μέσω της εφαρμογής ή επιλογή εικόνας από τα αρχεία της συσκευής
- Έξυπνη περικοπή εικόνας με αυτόματο τρόπο χρησιμοποιώντας αλγόριθμο ανίχνευσης ακρών
- Δυνατότητα του χρήστη να επιλέξει μόνος του το πλαίσιο σύμφωνα με το οποίο θα γίνει η περικοπή

- Βελτιστοποίηση εικόνας με κατάλληλες τεχνικές που την καθιστούν πιο ευανάγνωστη
- Αυτόματη αποθήκευση της βελτιωμένης φωτογραφίας στη συσκευή
- Προεπισκόπηση όλων των επεξεργασμένων εικόνων που δημιουργήθηκαν μέσω του Image Scanner στη συσκευή
- Δυνατότητα αποστολής/αποθήκευσης της βελτιωμένης εικόνας σε μορφή JPG και PDF
- Δυνατότητα αποστολής/αποθήκευσης πολλών επιλεγμένων εικόνων σε μορφή JPG και ενιαίο PDF
- Δυνατότητα αποστολής σε όλα τα μέσα κοινωνικής δικτύωσης και τα περιβάλλοντα ηλεκτρονικού ταχυδρομείου που είναι εγκαταστημένα στη συσκευή
- Μετατροπές οποιασδήποτε εικόνας σε έγγραφο PDF
- Μετατροπές πολλών εικόνων σε έγγραφο PDF

6.2.2 Παρουσίαση εφαρμογής

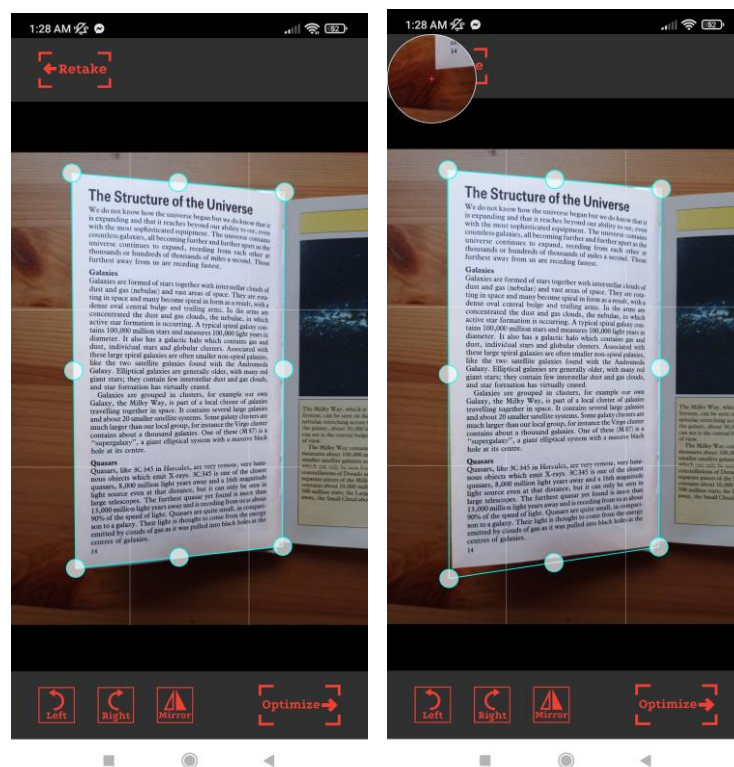
Αρχικά, η κύρια δραστηριότητα (Main Activity) αποτελείται από τρεις καρτέλες/Fragments: την αρχική οθόνη, την καρτέλα που είναι αποθηκευμένες όλες οι επεξεργασμένες φωτογραφίες και την καρτέλα που γίνεται μετατροπή οποιασδήποτε εικόνας σε έγγραφο(pdf). (Εικόνα 6.2.1 (α))



Εικόνα 6.2.1 (α, β, γ) Στιγμιότυπα της εφαρμογής

Στην αρχική καρτέλα της εφαρμογής, δίνεται στον χρήστη η επιλογή να διαλέξει άμα θέλει να τραβήξει μια φωτογραφία χρησιμοποιώντας την κάμερα ή να επιλέξει μια υπάρχουσα από την συσκευή του. (Εικόνα 6.2.2 (β)) Ο χρήστης ανάλογα με την επιλογή του καλείται να δώσει τα κατάλληλα δικαιώματα στην εφαρμογή για να μπορεί αυτή να έχει πρόσβαση στην κάμερα και στα αρχεία. (Εικόνα 6.2.2 (γ))

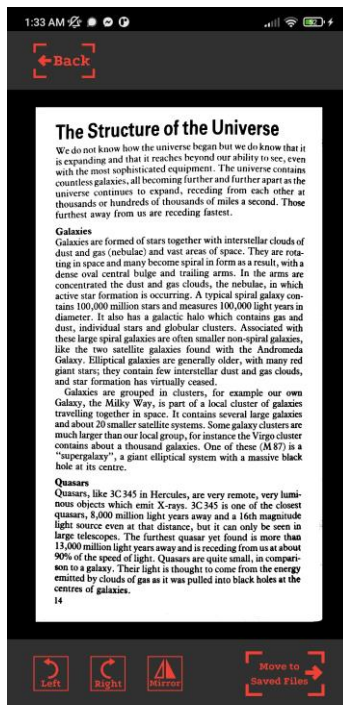
Έπειτα, γίνεται η έξυπνη περικοπή. Το Image Scanner εντοπίζει με τον αλγόριθμο ανίχνευσης ακρών το πλαίσιο που πιθανώς χρειάζεται. (Εικόνα 6.2.3 (α)) Παρ' όλα αυτά, ο χρήστης έχει τη δυνατότητα να το αλλάξει και να το προσαρμόσει όπως επιθυμεί χρησιμοποιώντας τον βοηθητικό μεγεθυντικό φακό. (Εικόνα 6.2.3 (β))



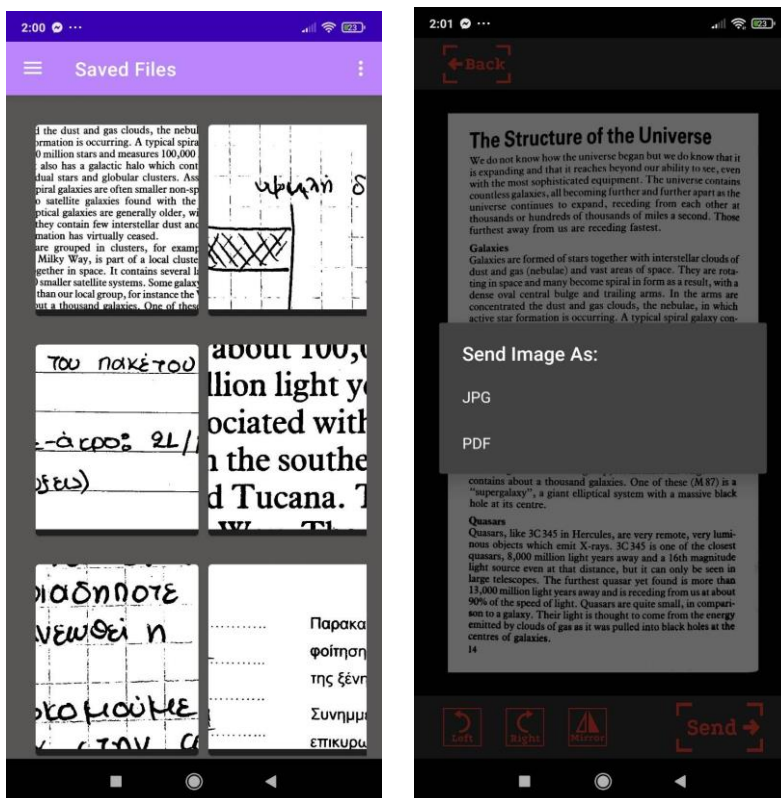
Εικόνα 6.2.3 (α, β) Στιγμιότυπα της Έξυπνης περικοπής

Στη συνέχεια, ο χρήστης οδηγείται στην δραστηριότητα που γίνεται η βελτιστοποίηση. Εκεί, εφαρμόζονται τα φίλτρα που βελτιώνουν την όψη και την αναγνωσιμότητα της εικόνας. Η εικόνα γίνεται πιο “καθαρή” και ευανάγνωστη. (Εικόνα 6.2.4) Η τελική φωτογραφία αποθηκεύεται αυτόματα στα αρχεία της συσκευής.

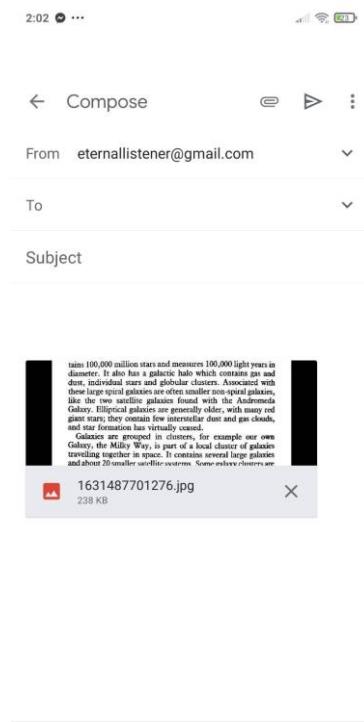
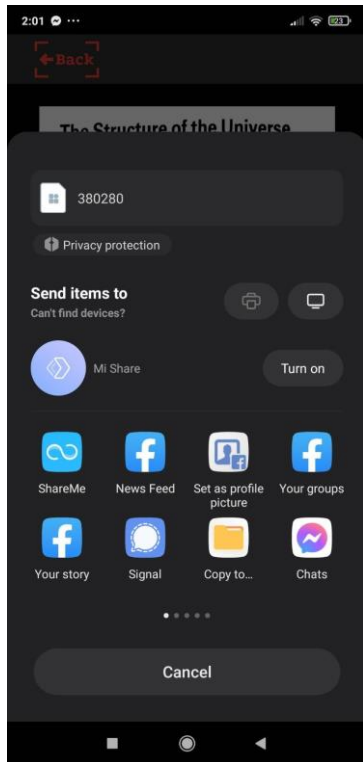
Στη δεύτερη καρτέλα εμφανίζονται όλες οι φωτογραφίες που έχουν επεξεργαστεί από την εφαρμογή. (Εικόνα 6.2.5(α)) Πατώντας πάνω στις φωτογραφίες αυτές εμφανίζονται προς προεπισκόπηση και υπάρχει η δυνατότητα να σταλούν στη μορφή που επιλέγει ο χρήστης (JPG ή PDF). (Εικόνα 6.2.5 (β)) Οι εικόνες μπορούν να σταλούν σε οποιαδήποτε πλατφόρμα κοινωνικής δικτύωσης ή να αποθηκευτούν στη κινητή συσκευή. (Εικόνα 6.2.6 (α, β)) Παράλληλα, υπάρχει η δυνατότητα να επιλεγθούν πολλές φωτογραφίες προς αποστολή. (Εικόνα 6.2.7)



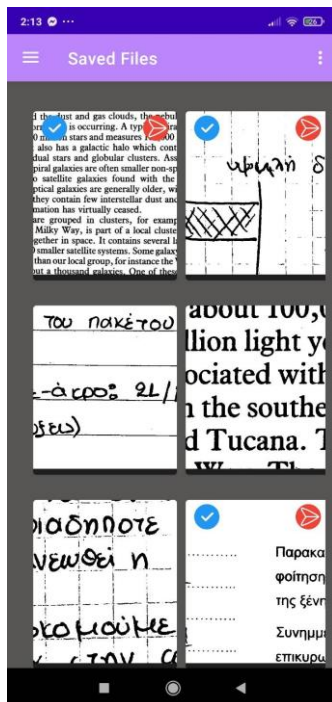
Εικόνα 6.2.4 Η βελτιστοποιημένη εικόνα



Εικόνα 6.2.5 (α, β) Οι αποθηκευμένες εικόνες στην δεύτερη καρτέλα και η προεπισκόπηση/αποστολή τους

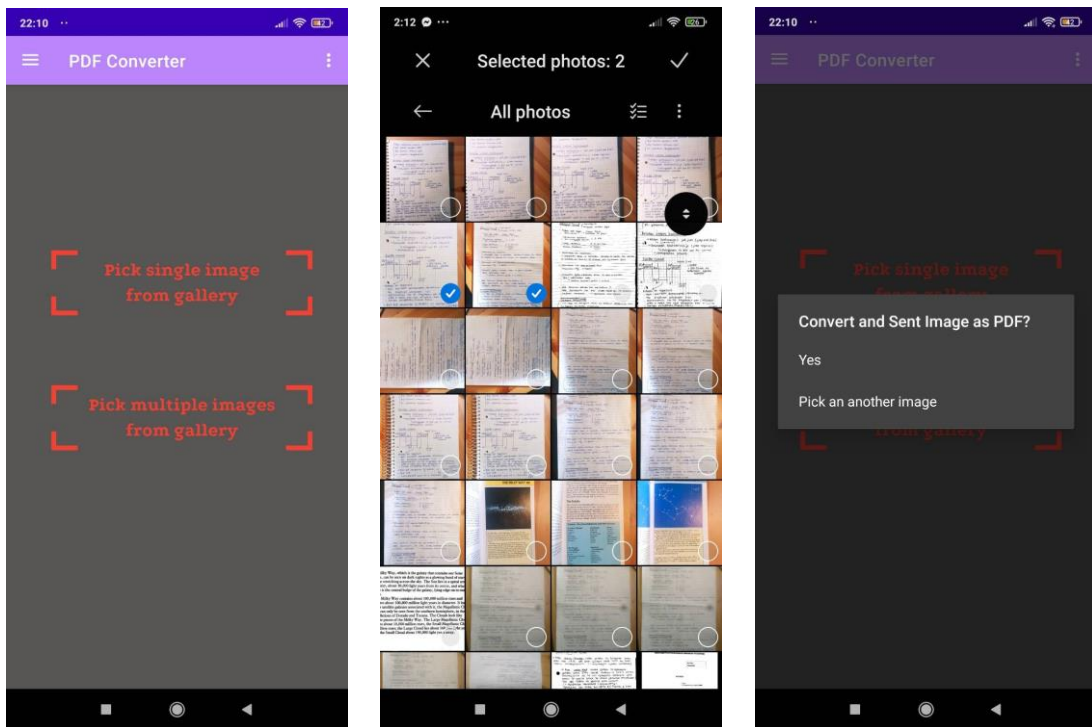


Εικόνα 6.2.6 (α, β) Δυνατότητες αποστολής εικόνων



Εικόνα 6.2.7 Επιλογή και “μαρκάρισμα” πολλών εικόνων προς αποστολή

Τέλος, στην καρτέλα που γίνεται η μετατροπή της μορφής αρχείων, ο χρήστης μπορεί να επιλέξει μια ή παραπάνω φωτογραφίες για να μετατραπούν σε PDF και αντίστοιχα να τις στείλει ή να τις αποθηκεύσει στη συσκευή του. (Εικόνα 6.2.8)



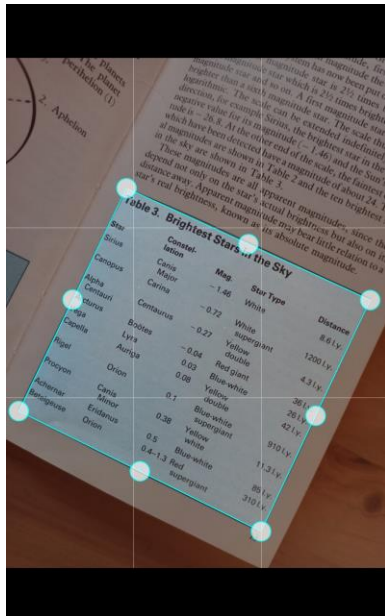
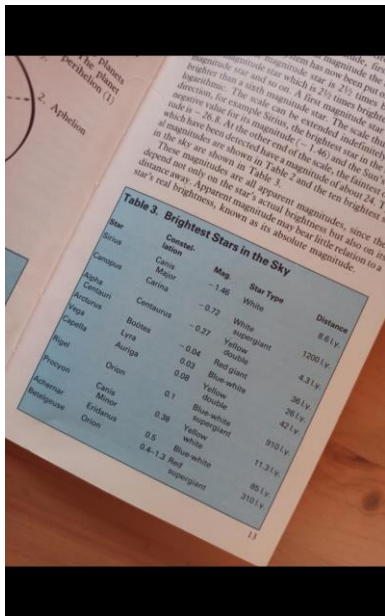
Εικόνα 6.2.8 Η τρίτη καρτέλα που μετατρέπει εικόνες σε έγγραφα

5.2.3 Αποτελέσματα εφαρμογής

Μερικά αποτελέσματα σάρωσης και βελτιστοποίησης της εφαρμογής Image Scanner είναι τα εξής:

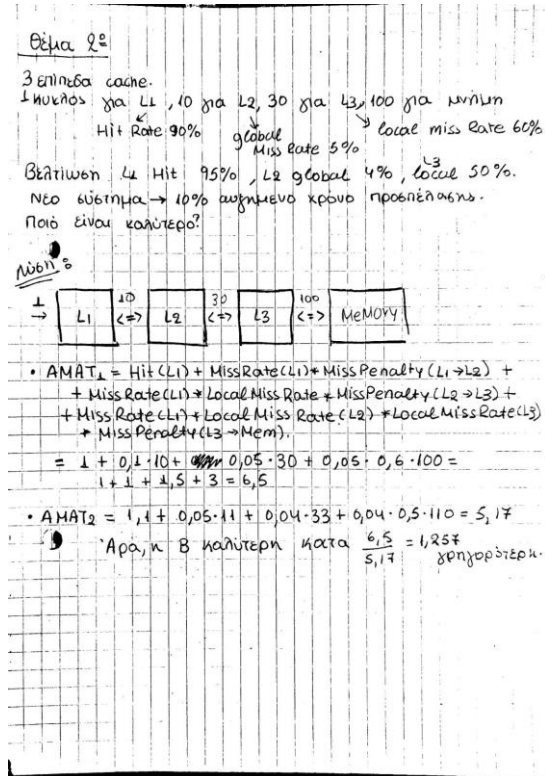
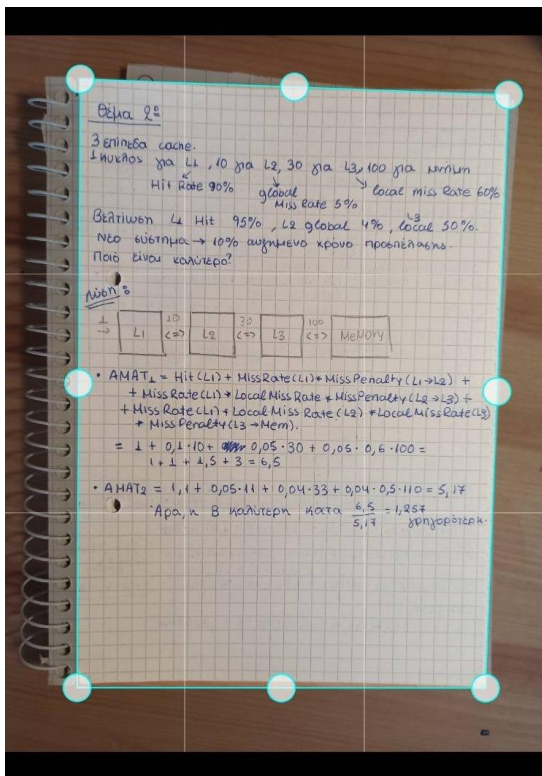
- Παράδειγμα 1: Πίνακας από βιβλίο (Εικόνα 6.2.9)
- Παράδειγμα 2: Σημειώσεις από μάθημα (Εικόνα 6.2.10)
- Παράδειγμα 3: Συνταγή από περιοδικό (Εικόνα 6.2.11)
- Παράδειγμα 4: Ημερήσια Διατροφή (Εικόνα 6.2.12)
- Παράδειγμα 5: Εγχειρίδιο Οδηγιών (Εικόνα 6.2.13)
- Παράδειγμα 6: Σελίδα από βιβλίο (Εικόνα 6.2.14)
- Παράδειγμα 7: Σελίδα από μουσικό βιβλίο (Εικόνα 6.2.15)

Τα συγκεκριμένα παραδείγματα επιλέχθηκαν γιατί μέσα από αυτά φαίνεται η χρησιμότητα της εφαρμογής, καθώς αποτελούν έγγραφα που μπορεί να χρειάζεται κάποιος σε έντυπη μορφή είτε για να μην χάσει (π.χ. εγχειρίδιο οδηγιών), είτε για να έχει πιο εύκολη πρόσβαση (π.χ. συνταγή από περιοδικό), είτε για να μοιράσει εύκολα με άλλους (π.χ. σελίδα από βιβλίο), είτε γιατί χρειάζονται πολλά αντίγραφα για διάφορα άτομα (π.χ. σημειώσεις από μάθημα).



Star	Constellation	Mag.	Star Type	Distance
Sirius	Canis Major	-1.46	White	8.6 l.y.
Canopus	Carina	-0.72	White supergiant	1200 l.y.
Alpha Centauri	Centaurus	-0.27	Yellow double	4.3 l.y.
Arcturus	Boötes	-0.04	Red giant	36 l.y.
Vega	Lyra	0.03	Blue-white	26 l.y.
Capella	Auriga	0.08	Yellow double	42 l.y.
Rigel	Orion	0.1	Blue-white supergiant	910 l.y.
Procyon	Canis Minor	0.38	Yellow white	11.3 l.y.
Achernar	Eridanus	0.5	Blue-white	85 l.y.
Betelgeuse	Orion	0.4-1.3	Red supergiant	310 l.y.

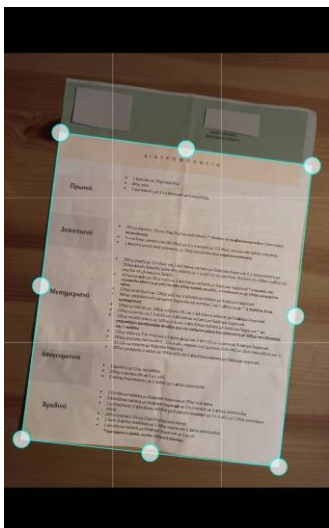
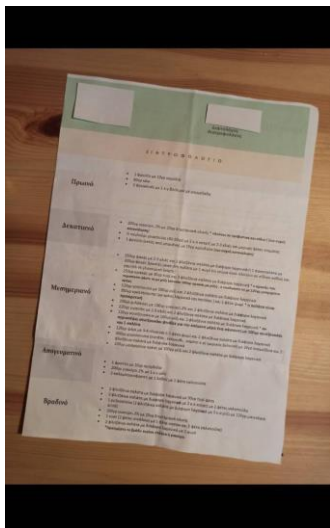
Εικόνα 6.2.9 Παράδειγμα από πίνακα βιβλίου



Εικόνα 6.2.10 Παράδειγμα από σημειώσεις μαθήματος σε τετράδιο



Εικόνα 6.2.11 Συνταγή από περιοδικό



ΔΙΑΤΡΟΦΟΛΟΓΙΟ

Πρωινό

- 1 φρούτο με 10gr κερύθια
- 60gr κρέα
- 2 φρονιμάκια με 1 κγ βούτυρο με μαρμελάδα

Δεκατιανό

- 200gr γαλλικό 2% με 20gr δημητριακά ολικής * κρέατο με παρμεζάνα και ελαιόλ (για κερύθια κρεμμύδι)
- 1% σουλάκι μπαντόνι (40-50gr) με 2 κ σκεπές με 2-3 ελιές και μαρούλι φρέσκιο ντομάτες
- 1 φρούτο (κατά περίπτωση) με 10gr αμυγδαλά (για κερύθια κρεμμύδι)

Μεσημεριανό

- 250gr φακές με 2-3 ελιές και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά ή 1 φακοσάλτσα με 200gr φακές βρετανικές μέσα στη σάλτσα με 1 αυγό (το κρέας είναι πολυκόμο σε σέβηρο κρέας και χυμώλο σε κρεμμύδι κρέας)
- 250gr σπαρακι με 30gr τυρί και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά * η κερύθια κρεμμύδι κρέας με το ρύζι (συνολικά 200gr κερύθια με ρύζι, ο σκεπές και με 100gr αναμεταμω κρέας)
- 120gr κρέας κοτόπουλο με 100gr ρύζι και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά
- 300gr κρεμμύδι (με κρέας λαχανικά και πατάτες) και 1 φέτα ψωμί * η σάλτσα είναι κρεμμύδι
- 200gr γαλλικές με 100gr γαλλικό 2% και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά
- 220gr γαλλικές με 2-3 ελιές και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά
- 120gr σουρωτάκια με 100gr ρύζι και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά * τα παρανοήσιμα σουρωτάκια φρέσκα και στα σπυράκια μέσα έδεσ σκεπές με 100gr σουρωτάκια και 1 σάλτσα
- 120gr φέτα με 3-4 ελιές και 1 φέτα ψωμί και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά
- 300gr μορπούτσου (πατάτες, κολοκύθα, κερύθια κρέας) με 20gr παρμεζάνα με 2 φλιτζάνια σάλτσα με διάφορα λαχανικά
- 120gr μορπούτσου κρέας με 100gr ρύζι και 2 φλιτζάνια σάλτσα με διάφορα λαχανικά

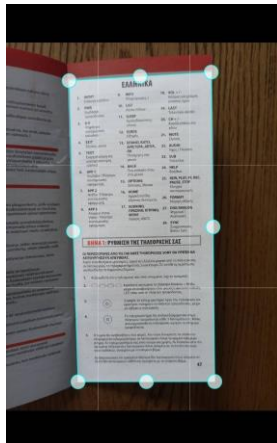
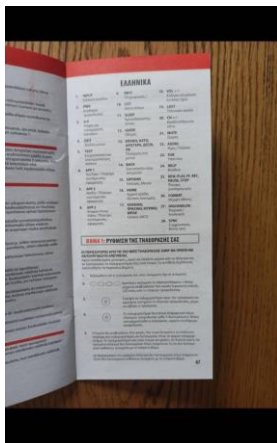
Απογευματινό

- 1 φρούτο με 10gr αμυγδαλά
- 200gr γαλλικό 2% με 1 κγ μέλι
- 2 καλαμποκούφρατες με 1 κρέας με 1 φέτα γαλλικό

Βραδινό

- 2 φλιτζάνια σάλτσα με διάφορα λαχανικά με 30gr τυρί φέτα
- 2 φλιτζάνια σάλτσα με διάφορα λαχανικά με 2 κ σκεπές με 2 φέτες γαλλικό
- 1 ρυσοσάλτσα (2 φλιτζάνια σάλτσα με διάφορα λαχανικά με 3 κ ρύζι με 100gr μανιτάρια φρέσκα)
- 200gr γαλλικό 2% με 25gr δημητριακά ολικής
- 1 ποτήρι (2 φέτες σάλτσα με 1 φέτα κερύθια και 1 φέτα γαλλικό)
- 2 φλιτζάνια σάλτσα με διάφορα λαχανικά με 1 αυγό
- *Προσχημένα το βράδυ κερύθια σάλτσα ή κρέας

Εικόνα 6.2.12 Παράδειγμα από διατροφολόγιο



ΕΛΛΗΝΙΚΑ

1. ΠΡΩΤΗ	5. ΜΕΡΟ	18. ΤΟΠΟΣ
2. ΔΕΥΤΕΡΗ	6. ΠΡΟΤΥΠΟ	19. ΑΝΤΙΣΤΑΣΗ
3. ΤΡΙΤΗ	7. ΠΡΟΤΥΠΗ	20. ΛΑΤΗ
4. ΤΕΤΑΡΤΗ	8. ΠΡΟΤΥΠΗ	21. ΣΥΝΤΑΞΗ
5. ΠΕΝΤΗ	9. ΠΡΟΤΥΠΗ	22. ΣΥΝΤΑΞΗ
6. ΕΚΤΗ	10. ΠΡΟΤΥΠΗ	23. ΣΥΝΤΑΞΗ
7. ΕΠΤΗ	11. ΠΡΟΤΥΠΗ	24. ΣΥΝΤΑΞΗ
8. ΟΚΤΩ	12. ΠΡΟΤΥΠΗ	25. ΣΥΝΤΑΞΗ
9. ΕΝΝΕΑ	13. ΠΡΟΤΥΠΗ	26. ΣΥΝΤΑΞΗ
10. ΔΕΚΑ	14. ΠΡΟΤΥΠΗ	27. ΣΥΝΤΑΞΗ
11. ΕΚΑΤΟΝΤΑ	15. ΠΡΟΤΥΠΗ	28. ΣΥΝΤΑΞΗ
12. ΔΥΟ	16. ΠΡΟΤΥΠΗ	29. ΣΥΝΤΑΞΗ
13. ΤΡΙΑ	17. ΠΡΟΤΥΠΗ	30. ΣΥΝΤΑΞΗ
14. ΤΕΤΡΑ	18. ΠΡΟΤΥΠΗ	31. ΣΥΝΤΑΞΗ
15. ΠΕΝΤΑ	19. ΠΡΟΤΥΠΗ	32. ΣΥΝΤΑΞΗ
16. ΕΚΤΑ	20. ΠΡΟΤΥΠΗ	33. ΣΥΝΤΑΞΗ
17. ΕΠΤΑ	21. ΠΡΟΤΥΠΗ	34. ΣΥΝΤΑΞΗ
18. ΟΚΤΩ	22. ΠΡΟΤΥΠΗ	35. ΣΥΝΤΑΞΗ
19. ΕΝΝΕΑ	23. ΠΡΟΤΥΠΗ	36. ΣΥΝΤΑΞΗ
20. ΔΕΚΑ	24. ΠΡΟΤΥΠΗ	37. ΣΥΝΤΑΞΗ
21. ΕΚΑΤΟΝΤΑ	25. ΠΡΟΤΥΠΗ	38. ΣΥΝΤΑΞΗ
22. ΔΥΟ	26. ΠΡΟΤΥΠΗ	39. ΣΥΝΤΑΞΗ
23. ΤΡΙΑ	27. ΠΡΟΤΥΠΗ	40. ΣΥΝΤΑΞΗ
24. ΤΕΤΡΑ	28. ΠΡΟΤΥΠΗ	41. ΣΥΝΤΑΞΗ
25. ΠΕΝΤΑ	29. ΠΡΟΤΥΠΗ	42. ΣΥΝΤΑΞΗ
26. ΕΚΤΑ	30. ΠΡΟΤΥΠΗ	43. ΣΥΝΤΑΞΗ
27. ΕΠΤΑ	31. ΠΡΟΤΥΠΗ	44. ΣΥΝΤΑΞΗ
28. ΟΚΤΩ	32. ΠΡΟΤΥΠΗ	45. ΣΥΝΤΑΞΗ
29. ΕΝΝΕΑ	33. ΠΡΟΤΥΠΗ	46. ΣΥΝΤΑΞΗ
30. ΔΕΚΑ	34. ΠΡΟΤΥΠΗ	47. ΣΥΝΤΑΞΗ

ΒΗΜΑ 1: ΡΥΘΜΙΣΗ ΤΗΣ ΘΑΥΡΟΣΤΑΣΗΣ

ΘΕΤΡΟΣΤΑΣΗ ΑΥΤΟ ΤΗΝ ΗΛΕΞΕ ΘΑΥΡΟΣΤΑΣΗ ΔΟΥΝ ΟΛΑ ΠΡΕΠΟΥΝ ΑΝΤΙΣΤΑΣΗ ΚΑΤΕΣΧΕΣΕ.

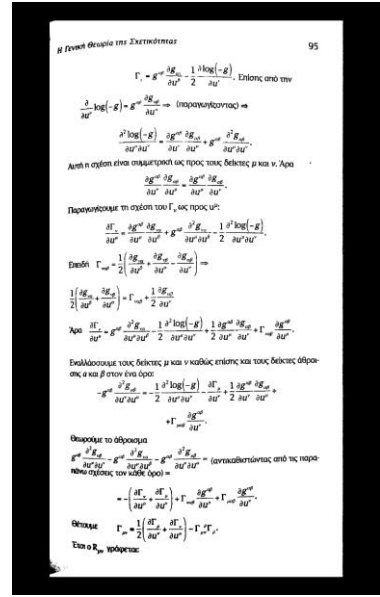
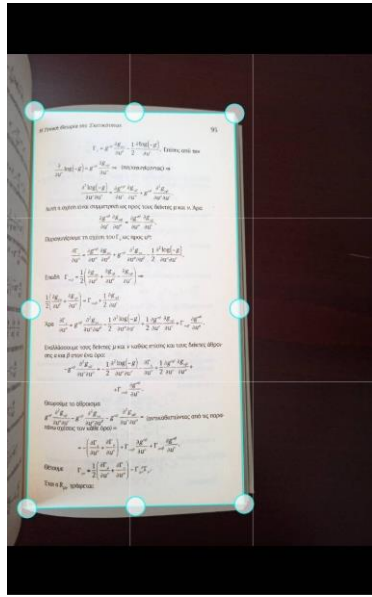
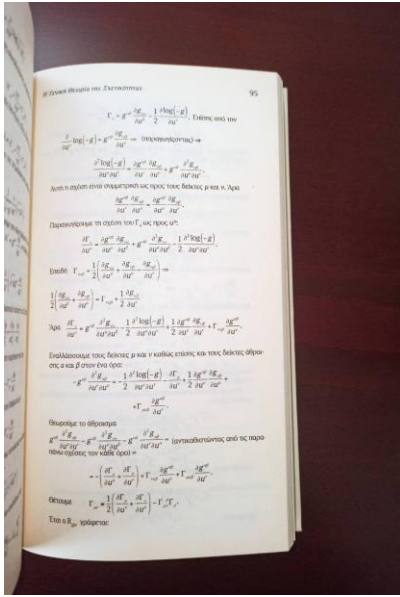
Από τα αποτελέσματα της δοκιμής θα πρέπει να προκύψουν οι ακόλουθοι αριθμοί: 1. Ο αριθμός των κερύθια κρέας που θα χρησιμοποιηθούν. 2. Ο αριθμός των κερύθια κρέας που θα χρησιμοποιηθούν. 3. Ο αριθμός των κερύθια κρέας που θα χρησιμοποιηθούν.

1. Βεβαιωθείτε ότι η ταχυμετρική συσκευή λειτουργεί σωστά.
2. Βεβαιωθείτε ότι η ταχυμετρική συσκευή είναι σωστά ρυθμισμένη.
3. Βεβαιωθείτε ότι η ταχυμετρική συσκευή είναι σωστά ρυθμισμένη.
4. Βεβαιωθείτε ότι η ταχυμετρική συσκευή είναι σωστά ρυθμισμένη.

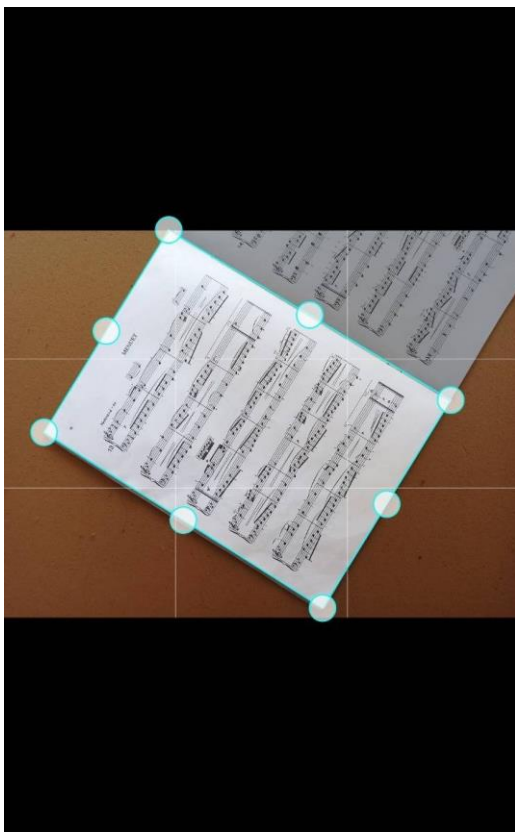
1. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη. 2. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη. 3. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη. 4. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη.

1. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη. 2. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη. 3. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη. 4. Η ταχυμετρική συσκευή πρέπει να είναι σωστά ρυθμισμένη.

Εικόνα 6.2.13 Παράδειγμα από εγχειρίδιο οδηγών



Παράδειγμα 6.2.14 Παράδειγμα από σελίδα βιβλίου



Εικόνα 6.2.15 Παράδειγμα από σελίδα μουσικού βιβλίου

6.2.4 Χρησιμότητα εφαρμογής

Το Image Scanner μπορεί να χρησιμοποιηθεί καθημερινά και να εξυπηρετεί διάφορες ανάγκες. Είναι ένα εργαλείο για δημιουργία ψηφιακών αντίγραφων και εύκολο μοίρασμα αρχείων. Ο χρήστης έχει την δυνατότητα να κρατάει σημαντικές πληροφορίες και έγγραφα στη κινητή του συσκευή και να τα κρατάει ασφαλή. Παράλληλα, μπορεί να οργανώσει καλύτερα τα αρχεία του και να εξοικονομήσει χώρο. Τα ψηφιακά αντίγραφα είναι πιο εύκολο να βρεθούν όταν χρειάζονται και η πρόσβαση των αρχείων μπορεί να γίνει σαφώς πιο γρήγορα ηλεκτρονικά.

Η εφαρμογή μπορεί να αντικαταστήσει σε έναν βαθμό την χρήση του ηλεκτρονικού σαρωτή. Έχει ικανοποιητικά αποτελέσματα στη σάρωση εκτυπωμένων ή χειρόγραφων κειμένων και βελτιώνει την αναγνωσιμότητά τους. Μπορεί να μην έχει πάντα την ίδια λειτουργικότητα με αυτή του σαρωτή, αλλά αποτελεί σίγουρα μια εύκολη και γρήγορη λύση για καθημερινές ανάγκες.

Ο λόγος να χρησιμοποιήσει κανείς το Image Scanner αντί μια απλή φωτογραφική μηχανή είναι ότι το αποτέλεσμα είναι πιο “καθαρό” και εύκολο στο μάτι. Επίσης, απαλείφονται τα περισευούμενα κενά και οι ανεπιθύμητες δυσαναλογίες.

Τη σημερινή εποχή, πολλές διαδικασίες γίνονται από απόσταση και το μοίρασμα ψηφιακών αρχείων πραγματοποιείται καθημερινά. Τόσο για προσωπικούς λόγους, όσο και επαγγελματικούς, το Image Scanner μπορεί να επιταχύνει αυτή την ενέργεια. Για παράδειγμα, μπορεί να το χρησιμοποιήσει ένας φοιτητής που χρειάζεται να στείλει στον καθηγητή του αποτελέσματα ηλεκτρονικής εξέτασης ή κάποιος που χρειάζεται να στείλει άμεσα μια υπεύθυνη δήλωση ή ένα υπογεγραμμένο έγγραφο. Μπορεί κάποιος να μοιράσει συνταγές, αποδείξεις, αναρτημένες ανακοινώσεις, λογαριασμούς, σελίδες από βιβλίο ή περιοδικό, οδηγίες χρήσης κ.ο.κ.

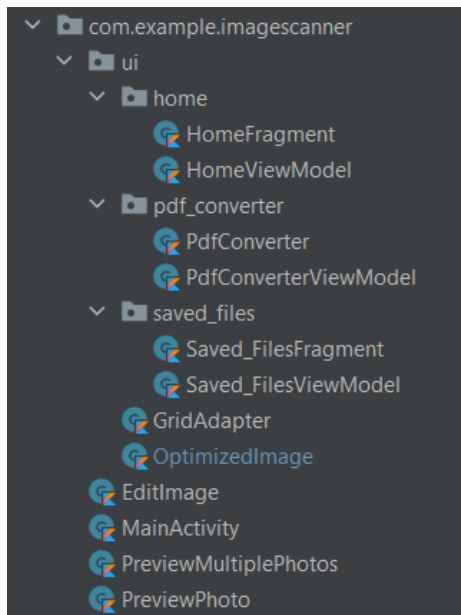
6.3 Τεχνικό κομμάτι

6.3.1 Περιεχόμενα της εφαρμογής στο Android Studio

Το λογισμικό περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση του Image Scanner είναι το Android Studio. Ο κώδικας αποτελείται από τρία Fragments και πέντε Activities. (Εικόνα 6.3.1)

Τα τρία Fragments:

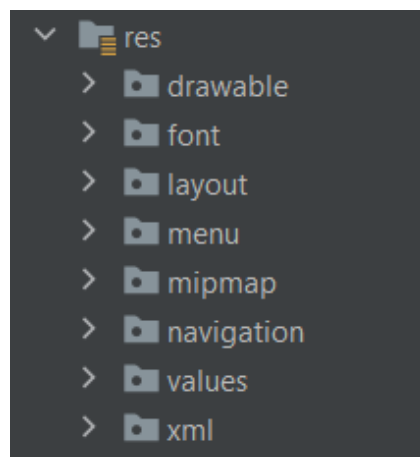
- Home: Είναι το Fragment που αφορά την αρχική καρτέλα. Εκεί ο χρήστης διαλέγει άμα θα χρησιμοποιήσει την κάμερα ή κάποια υπάρχουσα φωτογραφία αποθηκευμένη στη συσκευή του
- Saved Files: Σε αυτό το Fragment προβάλλονται όλα τα αποθηκευμένα αρχεία που έχουν επεξεργαστεί σε αυτή τη συσκευή από το Image Scanner
- PDF Converter: Εμφανίζει στην οθόνη τον μετατροπέα εικόνας σε έγγραφο PDF και ο χρήστης επιλέγει μία ή παραπάνω φωτογραφίες για μετατροπή



Εικόνα 6.3.1 Τα Fragments και τα Activities της εφαρμογής Image Scanner

Τα πέντε Activities:

- **Main Activity:** Είναι η κύρια δραστηριότητα και περιλαμβάνει τα τρία Fragments που εξηγήθηκαν παραπάνω
- **Edit Image:** Εδώ ξεκινάει η επεξεργασία εικονας με την έξυπνη περικοπη
- **Optimized Image:** Σε αυτό το Activity συμβαίνει η βελτιστοποίηση της εικόνας
- **Preview Photo:** Αφορά την περίπτωση που ο χρήστης ενώ βρίσκεται στο Fragment με τα αποθηκευμένα αρχεία, πατάει πάνω σε μια φωτογραφία για προεπισκόπηση και αποστολή
- **Preview Multiple Photos:** Αφορά την περίπτωση που ο χρήστης ενώ βρίσκεται στο Fragment με τα αποθηκευμένα αρχεία, πατήσει πάνω σε πολλές φωτογραφίες για προεπισκόπηση και αποστολή



Εικόνα 6.3.2 Ο φάκελος res στο Android Studio

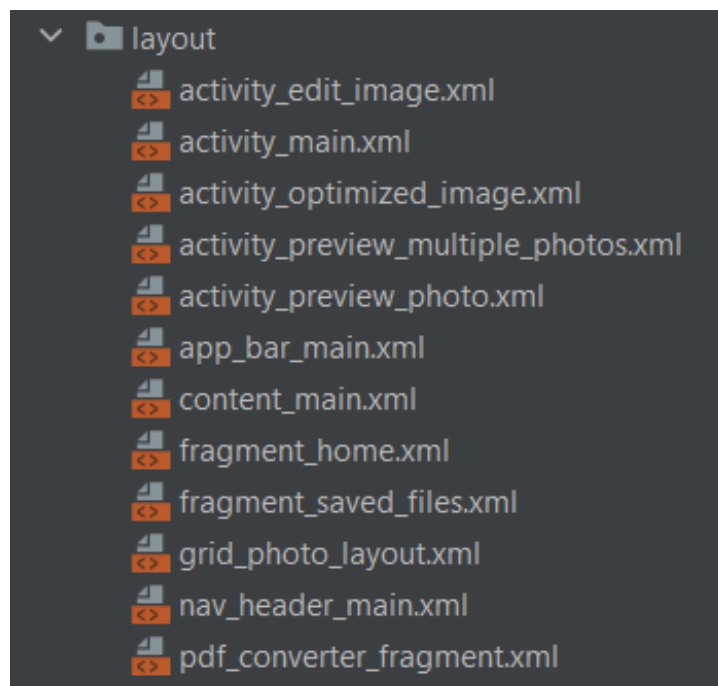
Ο φάκελος `res` αντιπροσωπεύει τη διεπαφή του χρήστη με την εφαρμογή. Χρησιμοποιείται για την αποθήκευση διάφορων τιμών των πόρων (`resources`) που χειρίζεται ο κώδικας της εφαρμογής.

Για παράδειγμα, ο φάκελος `drawable` περιέχει όλες τις εικόνες που χρησιμοποιήθηκαν για το `Image Scanner` όπως κουμπιά, λογότυπο, σύμβολα ενώ ο φάκελος `font` περιέχει την γραμματοσειρά που επιλέχθηκε. (Εικόνα 6.3.3)

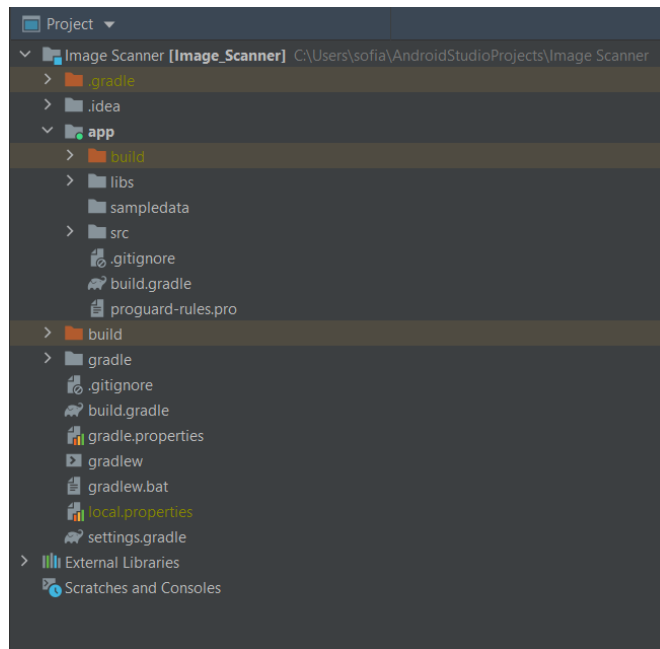


Εικόνα 6.3.3 Εικόνα για κουμπί που χρησιμοποιήθηκε σε διάφορα `Activities`

Ο φάκελος `layout` περιλαμβάνει όλα τα XML αρχεία, τα οποία αντιπροσωπεύουν τον σχεδιασμό της εφαρμογής. Κάθε `Activity` και `Fragment` πρέπει να συσχετίζεται με το αντίστοιχο XML αρχείο. (Εικόνα 6.3.4)



Εικόνα 6.3.4 Ο φάκελος `layout`



Εικόνα 6.3.5 Η ολοκληρωμένη δομή του κώδικα

6.3.2 Ανάλυση του κώδικα και των αλγορίθμων

- Main Activity

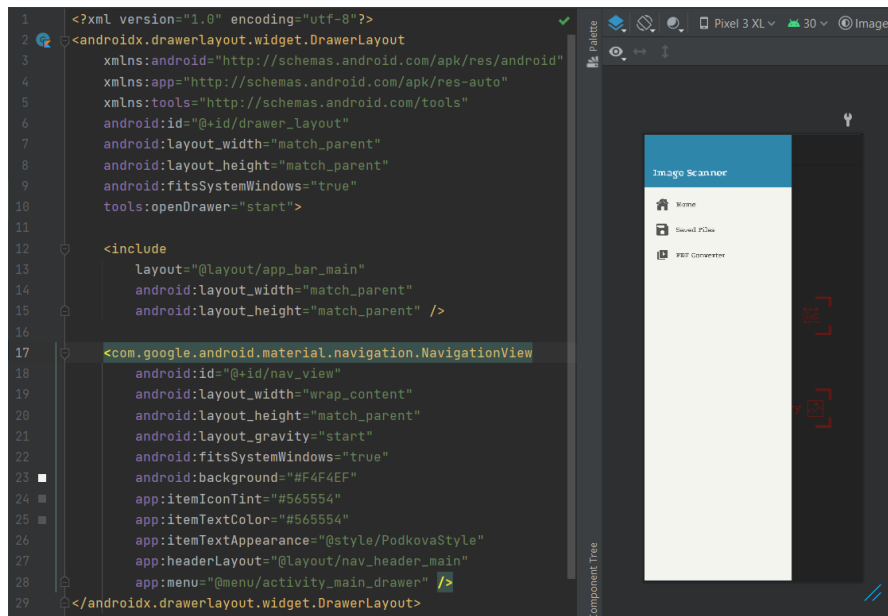
```

1  package com.example.imagescanner
2
3  import ...
14
15
16  class MainActivity : AppCompatActivity() {
17
18      private lateinit var appBarConfiguration: AppBarConfiguration
19
20      private var cameraState = "false"
21      private var test = "false"
22
23
24      override fun onCreate(savedInstanceState: Bundle?) {...}
47
48      override fun onCreateOptionsMenu(menu: Menu): Boolean {...}
53
54      override fun onSupportNavigateUp(): Boolean {...}
58
59      fun getMyData(): String {...}
63
64

```

Εικόνα 6.3.6 Αναλυτικά το Main Activity

Η κύρια δραστηριότητα είναι το Activity που περιλαμβάνει τα τρία Fragments. Η βασική δουλειά της είναι να συνδέσει τα Fragments μαζί της. (Εικόνα 6.3.6)



Εικόνα 6.3.7 Το layout του Main Activity

Όσον αφορά το layout του Main Activity, αυτό αποτελείται από μια “μπάρα” πλοήγησης (Navigation View), η οποία ανάλογα με την επιλογή του χρήστη τον μεταφέρει στο αντίστοιχο Fragment. (Εικόνα 6.3.7)

- Home Fragment

Το Fragment αυτό είναι ουσιαστικά η αρχική οθόνη. Περιλαμβάνει δύο κουμπιά και ο χρήστης επιλέγει άμα θέλει να ανεβάσει μια φωτογραφία από την κάμερα ή από τη συσκευή του. Ο κώδικας περιέχει τις συναρτήσεις που αναλαμβάνουν τα δικαιώματα που πρέπει να δώσει ο χρήστης για να μπορεί να έχει η εφαρμογή πρόσβαση στα αρχεία του. Όταν αυτά έχουν δοθεί όπως πρέπει, τότε ανοίγει είτε ένα νέο Activity κάμερας είτε ένα με την συλλογή φωτογραφιών της συσκευής και ανάλογα τρέχει η συνάρτηση με τα αποτελέσματα (OnActivityResult). Αυτή η συνάρτηση, θα οδηγήσει στη συνέχεια στο επόμενο Activity που ξεκινάει την επεξεργασία εικόνας και επομένως θα χρειαστεί να σταλθεί σε αυτό και η εικόνα που επιλέχθηκε.

```

1 package com.example.imagescanner.ui.home
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31 class HomeFragment : Fragment() {
32
33     override fun onCreateView(
34         inflater: LayoutInflater,
35         container: ViewGroup?,
36         savedInstanceState: Bundle?
37     ): View { ... }
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66     private val CAMERA_PERM_CODE = 101
67     private val CAMERA_REQUEST_CODE = 102
68     private val RESULT_LOAD_IMG = 106
69
70     var currentPhotoPath: String? = null
71     private fun askCameraPermissions() { ... }
72
73     override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String?>, grantResults: IntArray) { ... }
74
75
76
77
78     @SuppressWarnings("QueryPermissionsNeeded")
79     /////////////// CAMERA ///////////////
80     private fun dispatchTakePictureIntent() { ... }
81     @Throws(IOException::class)
82     private fun createImageFile(): File? { ... }
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131 /////////////// IMPORT ///////////////
132 private fun galleryIntent() { ... }
133
134
135
136
137
138     override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) { ... }
139
140
141
142
143
144
145
146

```

Εικόνα 6.3.8 Αναλυτικά το Home Fragment

Το layout αυτού του Fragment έχει μόνο τα δύο εικόνες-κουμπιά και το λογότυπο της εφαρμογής. Τα δύο κουμπιά όπως εξηγήθηκε έχουν να κάνουν με την επιλογή του χρήστη για το αν θα ανοίξει την κάμερα ή την συλλογή φωτογραφιών.

- Saved Files Fragment

Σε αυτό το Fragment βρίσκονται όλες οι αποθηκευμένες βελτιστοποιημένες εικόνες. Στον κώδικα υπάρχει μια συνάρτηση που αναζητάει όλες τις εικόνες της συσκευής και εντοπίζει αυτές που έχουν επεξεργαστεί μέσω του Image Scanner. Για την πραγματοποίηση αυτού του αλγόριθμου χρησιμοποιήθηκε SQLite, ένα σύστημα βάσης δεδομένων που χρησιμοποιεί γλώσσα προγραμματισμού SQL.

Επειδή πολλές εικόνες χρειάστηκαν να προβληθούν με τον ίδιο τρόπο δημιουργήθηκε μια δυναμική λίστα ανακύκλωσης (RecyclerView) και ένας “αντάπτορα” (GridAdapter) για να την χειρίζεται. (Εικόνα 6.3.10)

Η βιβλιοθήκη RecyclerView διευκολύνει την αποτελεσματική προβολή μεγάλων συνόλων δεδομένων. Όπως αναφέρει και το όνομα δημιουργεί δυναμικά στοιχεία όταν χρειάζονται και τα ανακυκλώνει. Αυτή η επαναχρησιμοποίηση βελτιώνει σημαντικά την απόδοση, βελτιώνοντας την απόκριση της εφαρμογής και μειώνοντας την κατανάλωση ενέργειας. [25]


```

1 package com.example.imagescanner.ui.saved_files
2
3 import ...
4
18
19 class SavedFilesFragment : Fragment() {
20
21     private var photos: ArrayList<Uri> = ArrayList()
22
23     @RequiresApi(Build.VERSION_CODES.Q)
24     override fun onCreateView(
25         inflater: LayoutInflater,
26         container: ViewGroup?,
27         savedInstanceState: Bundle?
28     ): View? {
29
30         photos = getAllImages()
31
32         val view = inflater.inflate(R.layout.fragment_saved_files, container, attachToRoot: false)
33         val mRecyclerView : RecyclerView = view!!.findViewById(R.id.recyclerview_id)
34         val mAdapter = GridAdapter(photos, requireActivity().applicationContext)
35         val layoutManager = GridLayoutManager(requireActivity().applicationContext, spanCount: 2 /* Pictures per line */)
36
37         mRecyclerView.layoutManager = layoutManager
38         mRecyclerView.adapter = mAdapter
39
40         return view
41     }
42
43     private fun getAllImages(): ArrayList<Uri> {...}
44 }

```

Εικόνα 6.3.9 Αναλυτικά το Saved Files Fragment

Η κλάση Adapter γράφεται ξεχωριστά και δέχεται σαν όρισμα μια λίστα από εικόνες και ο κώδικας περιγράφει τα γεγονότα που θα συμβούν άμα ο χρήστης πατήσει οποιαδήποτε από αυτές. Όποια εικόνα επιλεγεί θα έχει την ίδια αντιμετώπιση. Για παράδειγμα, άμα ο χρήστης πατήσει μια φωτογραφία για προβολή τότε θα μεταφερθεί σε ένα καινούριο Activity, ενώ άμα πατήσει παρατεταμένα πάνω της θα μπορέσει να τη σημαδέψει και να επιλέξει κι άλλες.

```

1 package com.example.imagescanner.ui
2
3 import ...
4
18
19
20 class GridAdapter(tileList: ArrayList<Uri>, context: Context) :
21     RecyclerView.Adapter<GridAdapter.MyViewHolder>() {
22     private val mTiles: ArrayList<Uri> = tileList
23     private val mContext: Context = context
24     private var mListener: OnItemClickListener? = null
25     private val arraySelectedIDs: MutableList<String> = mutableListOf()
26
27     interface OnItemClickListener {...}
28
29     class MyViewHolder(itemView: View, listener: OnItemClickListener?) :
30         RecyclerView.ViewHolder(itemView) {...}
31
32     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {...}
33
34     override fun onBindViewHolder(holder: MyViewHolder, position: Int) {...}
35
36     override fun getItemCount(): Int {...}
37 }

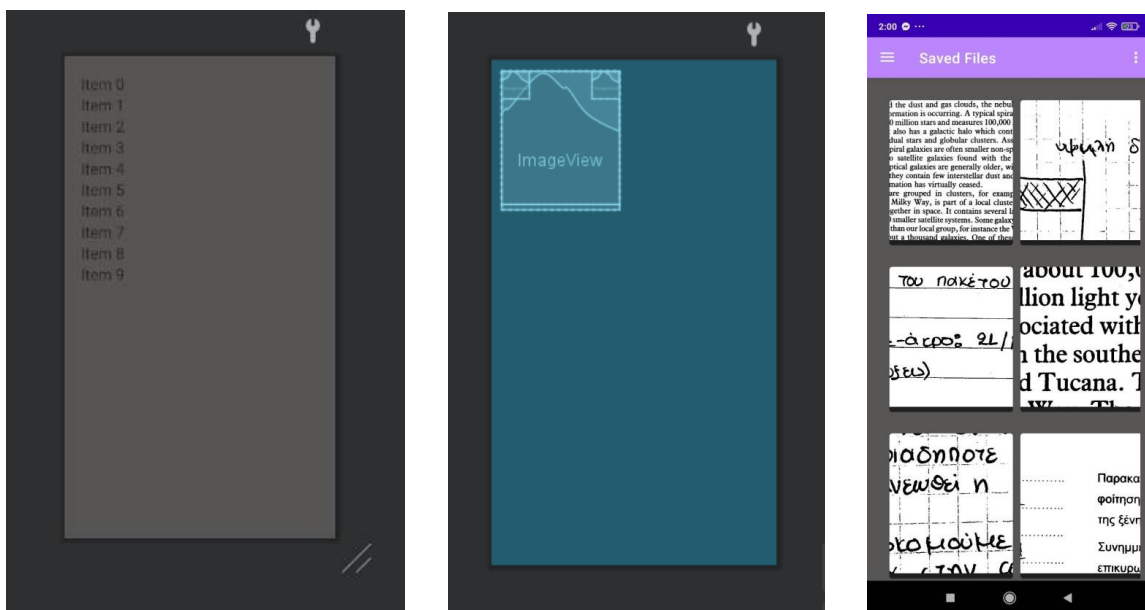
```

Εικόνα 6.3.10 Ο αντάπτορας GridAdapter

Το layout του Saved Files Fragment περιλαμβάνει μόνο τη λίστα με τις φωτογραφίες που στέλνεται στον αντάπτορα. (Εικόνα 6.3.11(α))

Το αρχείο XML που συνδέεται με τον αντάπτορα περιλαμβάνει τον πρότυπο σχεδιασμό της κάθε εικόνας που θα υπάρχει στην λίστα. (Εικόνα 6.3.11(β)) Αυτό αποτελείται από πληροφορίες όπως το μέγεθος και τον προσανατολισμό της, αλλά και από δύο εικονίδια που σηματοδοτούν το “μαρκάρισμα” και την αποστολή. Αυτά τα δυο εμφανίζονται όταν ο χρήστης πατήσει παρατεταμένα πάνω στις φωτογραφίες.

Παρόλο που τα δύο αρχεία XML έχουν ξεχωριστές λειτουργίες, παρατηρείται πως συνδέονται μεταξύ τους και δημιουργούν τελικά μια λίστα με πολλές εικόνες. (Εικόνα 6.3.11(γ))



Εικόνα 6.3.11 (α, β, γ) Τα XML αρχεία του Saved Files Fragment και του Adapter συνδέονται και το τελικό αποτέλεσμα είναι μια λίστα με φωτογραφίες

- PDF Converter Fragment

Το τρίτο Fragment έχει να κάνει με τον μετατροπέα εικόνας σε έγγραφο. Μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εικόνα της συσκευής του χρήστη. Είναι ένα εργαλείο που δημιουργεί γρήγορα PDF αρχεία και τα μοιράζει ή αποθηκεύει. (Εικόνα 6.3.12)

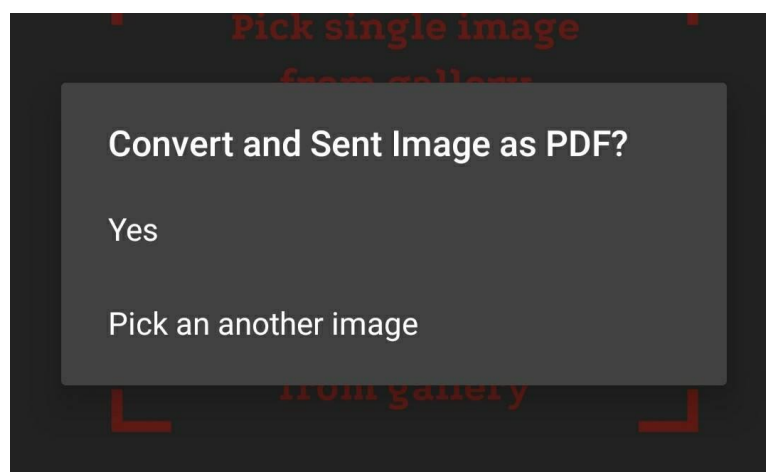
Υπάρχουν οι συναρτήσεις που ανάλογα με το κουμπί που πάτησε ο χρήστης ανοίγει η συλλογή φωτογραφιών της συσκευής και αυτός διαλέγει μια ή παραπάνω εικόνες για μετατροπή. Έπειτα, δημιουργήθηκε η συνάρτηση που διαχειρίζεται το αποτέλεσμα, δηλαδή την εικόνα ή τις εικόνες που επιλέχθηκαν.

Αυτή η συνάρτηση, αφού ελέγξει την επιλογή του χρήστη, κάνει τη μετατροπή. Πριν γίνει η μετατροπή, εμφανίζει στην οθόνη ένα κουτί ειδοποίησης (Alert Dialog).

Ένα Alert Dialog είναι ένα μικρό παράθυρο που ζητάει από τον χρήστη να λάβει μια απόφαση ή να εισαγάγει πρόσθετες πληροφορίες. (Εικόνα 6.3.13) Δεν γεμίζει την οθόνη και συνήθως χρησιμοποιείται για τυπικά συμβάντα που απαιτούν από τους χρήστες να προβούν σε κάποια ενέργεια προτού προχωρήσουν. [26]

```
1 package com.example.imagescanner.ui.pdf_converter
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30 class PdfConverter : Fragment() {
31
32     override fun onCreateView(
33         inflater: LayoutInflater,
34         container: ViewGroup?,
35         savedInstanceState: Bundle?
36     ): View? { ... }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52     private val RESULT_LOAD_IMG = 106
53     private val RESULT_LOAD_MULT_IMG = 102
54
55     private fun galleryIntent() { ... }
56
57
58
59
60
61     private fun galleryMultipleIntent() { ... }
62     private val uriArray: ArrayList<String> = ArrayList()
63
64
65
66
67
68
69     override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) { ... }
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123     var pdfDocument // to create .pdf file
124         : PdfDocument? = null
125
126     private fun sharePDF(imageUri: Uri) { ... }
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159     private fun shareMultPDF(arrayList: ArrayList<String>) { ... }
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222 }
```

Εικόνα 6.3.12 Αναλυτικά το PDF Converter Fragment



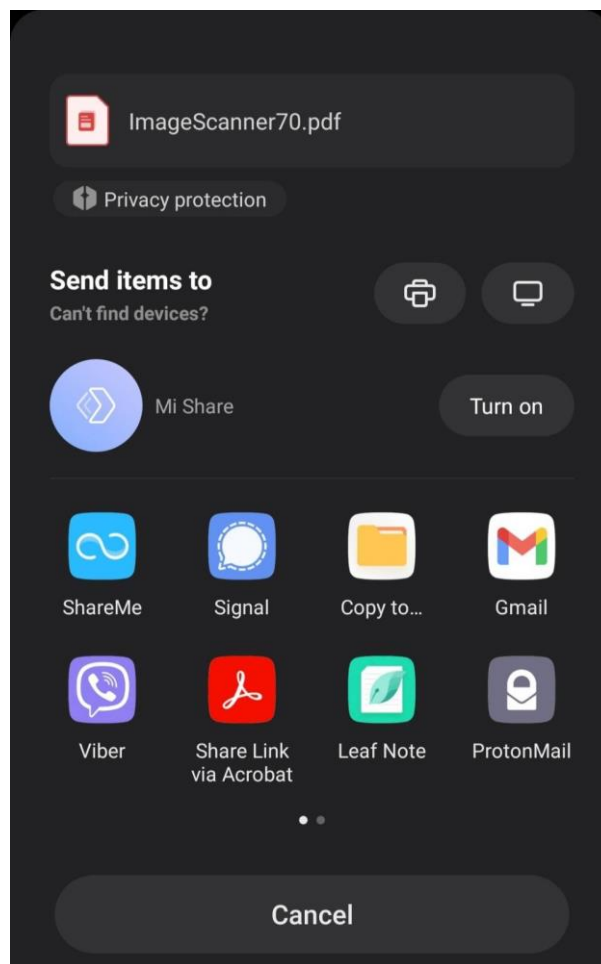
Εικόνα 6.3.13 Το Alert Dialog

Το Alert Dialog σε αυτή την περίπτωση ζητάει από την χρήστη να πάρει την απόφαση αν θέλει όντως να επιλέξει την εικόνα/εικόνες που διάλεξε ή να επιλέξει μια άλλη/άλλες.

Στην περίπτωση που ο χρήστης πατήσει το κουμπί της θετικής απάντησης, καλείται η συνάρτηση που κάνει τη μετατροπή. Η μετατροπή γίνεται με την βοήθεια της κλάσης PdfDocument. Αυτή, δημιουργεί ένα αρχείο PDF από ένα περιεχόμενο Android. Σε αυτή την εφαρμογή, αυτό το περιεχόμενο είναι η εικόνα/εικόνες που έχουν επιλεγεί και έχουν μετατραπεί σε αρχεία bitmap.

Το bitmap είναι ένας τύπος οργάνωσης μνήμης που χρησιμοποιείται για την αποθήκευση ψηφιακών εικόνων και μια διαδεδομένη μορφή αρχείου εικόνας. Ο όρος bitmap σημαίνει ένας “χάρτης” δυαδικών ψηφίων, μια χωροταξικά χαρτογραφημένη συστοιχία δυαδικών ψηφίων. Η μετατροπή σε bitmap είναι σημαντική διαδικασία γιατί διευκολύνει κατά πολύ τη μετέπειτα μετατροπή σε αρχείο PDF.

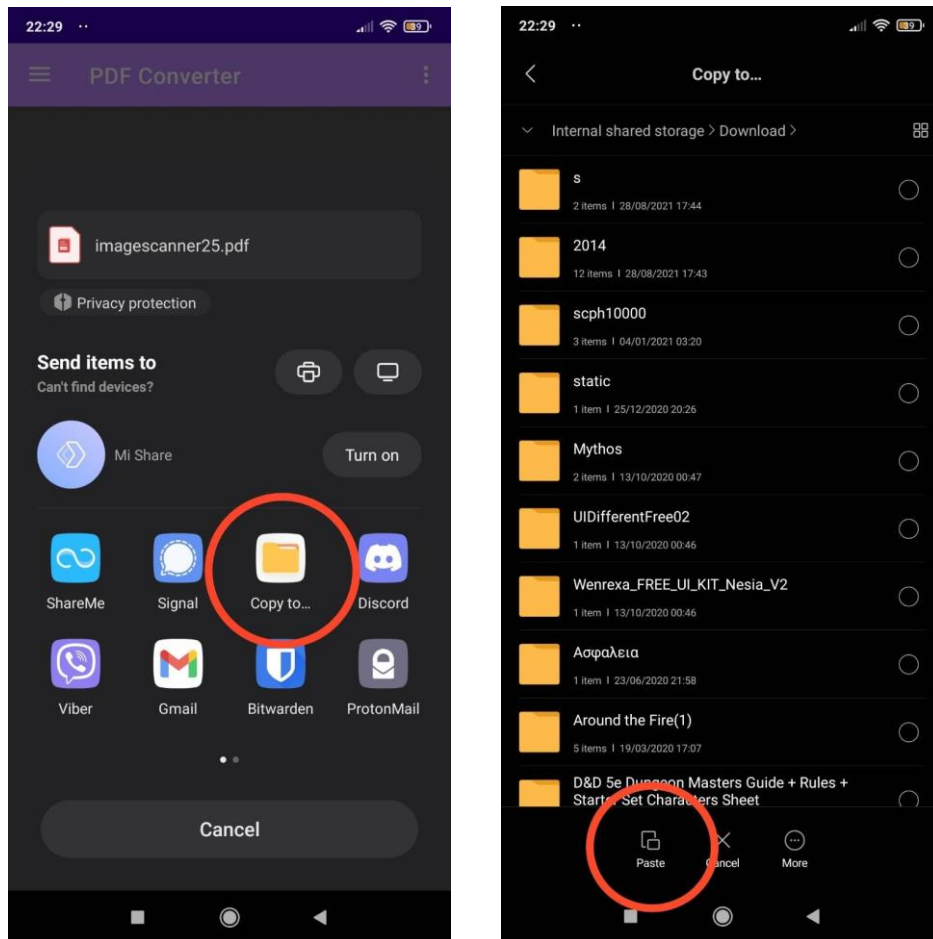
Μόλις δημιουργηθεί το αρχείο της μορφής PDF, αυτόματα βγαίνει ένα παράθυρο που σχετίζεται με την αποστολή του αρχείου. (Εικόνα 6.3.14)



Εικόνα 6.3.14 Παράθυρο αποστολής αρχείου

Αυτό το παράθυρο αποστολής του Android (Android Sharesheet) δίνει στους χρήστες τη δυνατότητα να αποστείλουν τα αρχεία τους σε όλα τα μέσα κοινωνικής δικτύωσης και

περιβάλλοντα ηλεκτρονικού ταχυδρομείου, αλλά και να προβληθούν ή να αποθηκευτούν στις συσκευές. Για παράδειγμα, στην Εικόνα 6.3.15 ένα αρχείο PDF αποθηκεύεται στη συσκευή του κινητού μέσω του παράθυρου αποστολής.



Εικόνα 6.3.15 Αποθήκευση του PDF αρχείου στη συσκευή

Το layout του PDF Converter Fragment περιλαμβάνεται μόνο από δύο κουμπιά. Το πρώτο αφορά το κουμπί για την περίπτωση που ο χρήστης θέλει να επιλέξει μόνο μια φωτογραφία από την συλλογή της συσκευής του. Το δεύτερο έχει να κάνει με την πολλαπλή επιλογή φωτογραφιών όπου ο χρήστης “μαρκάρει” το σύνολο εικόνων που θέλει να μετατρέψει.

- Edit Image Activity

Σε αυτό το Activity ξεκινάει η διαδικασία επεξεργασίας εικόνας με τον αλγόριθμο έξυπνης περικοπής.

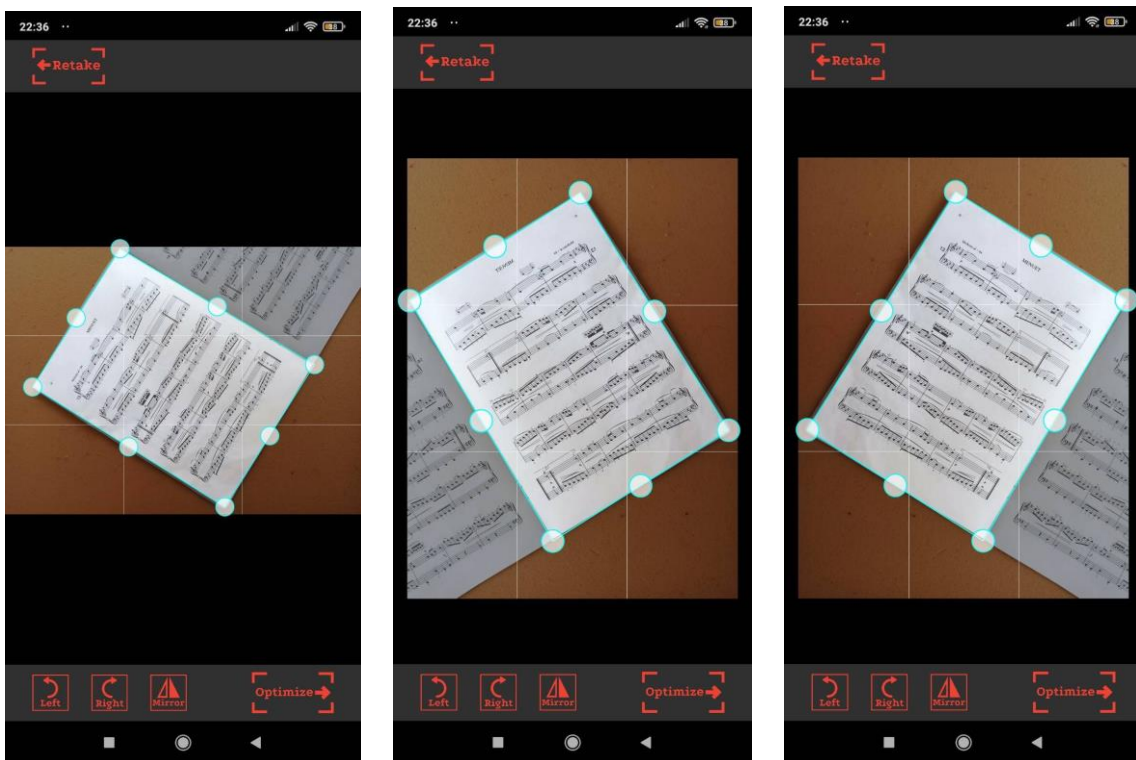
```

1 package com.example.imagescanner
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 class EditImage : AppCompatActivity() {
26     private var mScaleGestureDetector: ScaleGestureDetector? = null
27     private var ivZoom: ImageView? = null
28     var type: String = ""
29
30
31     override fun onCreate(savedInstanceState: Bundle?) {...}
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47     private fun saveImage(bitmap: Bitmap, saveFile: File) {...}
48
49
50
51
52
53
54
55
56
57
58
59     override fun onBackPressed() {...}
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74     override fun onTouchEvent(event: MotionEvent?): Boolean {...}
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159     inner class ScaleListener : ScaleGestureDetector.SimpleOnScaleGestureListener() {...}
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174 }

```

Εικόνα 6.3.16 Αναλυτικά το Edit Image Activity

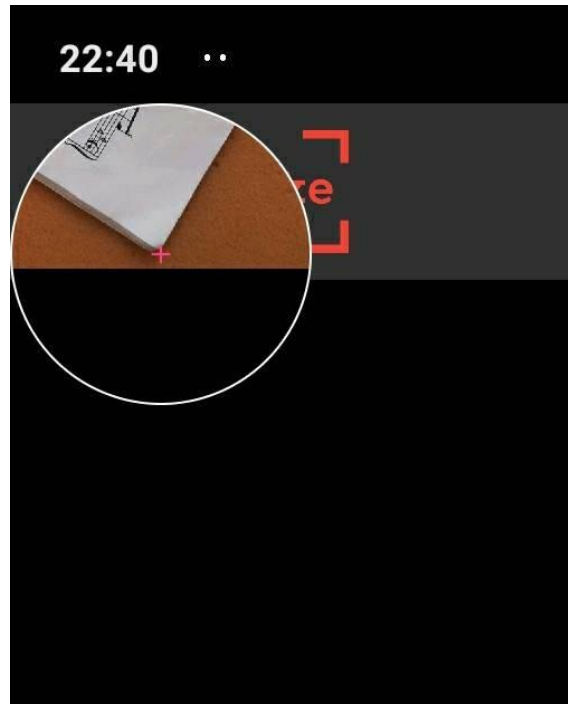
Η δραστηριότητα αυτή καλείται από το Home Fragment. Σε αυτό το Activity συμβαίνει η προεπισκόπηση της εικόνας που επιλέχθηκε από τη συλλογή ή τραβήχτηκε από την κάμερα. Εδώ ο χρήστης μπορεί να γυρίσει τη φορά της εικόνας προς τα δεξιά ή αριστερά καθώς και να αναστρέψει τον προσανατολισμό της “καθρεφτίζοντας” την. (Εικόνα 6.3.17)



Εικόνα 6.3.17 Δυνατότητα αλλαγής προσανατολισμού επιλεγμένης εικόνας

Σε αυτό το σημείο, η εφαρμογή εντοπίζοντας τις άκρες του επιθυμητού πλαισίου προτείνει έναν έξυπνο τρόπο περικοπής. Παρ' όλα αυτά, ο χρήστης έχει την ευχέρεια να αλλάξει το πλαίσιο και να διαλέξει μόνος του τα σημεία περικοπής που επιθυμεί.

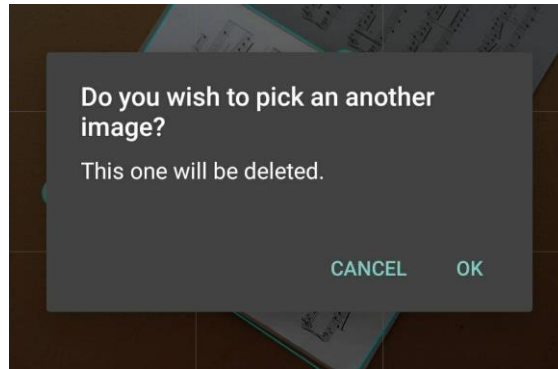
Υπάρχει επίσης η παρουσία ενός βοηθητικού “μεγεθυντικού φακού” που βοηθάει τον χρήστη να εντοπίσει τα σημεία που θα επιλέξει για περικοπή, κάνοντας μεγέθυνση. (Εικόνα 6.3.18)



Εικόνα 6.3.18 Ο βοηθητικός μεγεθυντικός φακός που εμφανίζεται στην άκρη της οθόνης

Στην περίπτωση που ο χρήστης αποφασίσει να αλλάξει την εικόνα μπορεί να πατήσει ή το κατάλληλο κουμπί (αναγράφεται “Retake”) ή το κουμπί της συσκευής του που πάει πίσω. Τότε, εμφανίζεται ένα Alert Dialog που ενημερώνει τον χρήστη για την επιλογή του να πάει πίσω και του ζητάει να επιβεβαιώσει ή να παραμείνει στην ίδια δραστηριότητα. Η παρουσία αυτού του παραθύρου είναι σημαντική γιατί ο χρήστης μπορεί να πατήσει το κουμπί “πίσω” κατά λάθος και στην πραγματικότητα να επιθυμεί να κρατήσει την εικόνα που εμφανίζεται. (Εικόνα 6.3.19)

Άμα πατηθεί το κουμπί “Yes” τότε ο χρήστης μεταφέρεται στο ανάλογο προηγούμενο Activity. Αυτό μπορεί να είναι η κάμερα ή η συλλογή του κινητού με τις φωτογραφίες. Η επιλογή εξαρτάται από το τι είχε αποφασιστεί προηγουμένως, δηλαδή ποιο κουμπί είχε πατήσει αρχικά στο Main Activity και συγκεκριμένα στο Home Fragment.



Εικόνα 6.3.20 Το Alert Dialog για την επιλογή του χρήστη να πάει πίσω

Το layout του Edit Image Activity περιλαμβάνει πέντε κουμπιά: ένα που μεταφέρει πίσω τον χρήστη, ένα για δεξιά περιστροφή της εικόνας, ένα για αριστερή, ένα για κατοπτρισμό του προσανατολισμού της και τέλος, ένα που υποδηλώνει τη μεταφορά του χρήστη στην επόμενη δραστηριότητα. Στη μέση της οθόνης προβάλλεται η φωτογραφία που έχει επιλεγθεί και το πλαίσιο που αφορά την περικοπή

Συνέχεια αυτής της δραστηριότητας αποτελεί η βελτιστοποίηση εικόνας.

- Optimized Image Activity

Σε αυτό το Activity γίνεται η ωραιοποίηση της εικόνας και η βελτιστοποίηση της αναγνωσιμότητας της.

Έχουν γραφτεί δύο συναρτήσεις που συμβάλλουν στο να γίνει η φωτογραφία πιο ευανάγνωστη. Η μια περιλαμβάνει τον αλγόριθμο της κατωφλίωσης που μετατρέπει την εικόνα από μονοχρωματική σε δυαδική και βελτιώνει την ευκρίνεια. Η άλλη έχει να κάνει με τον αλγόριθμο που αφαιρεί τον θόρυβο της φωτογραφίας.

```
1 package com.example.imagescanner.ui
2
3 import ...
33
34
35 class OptimizedImage : AppCompatActivity() {
36     @SuppressWarnings("WrongThread")
37     override fun onCreate(savedInstanceState: Bundle?) {...}
73
74     private fun grayscaleToBin(bitmap: Bitmap): Bitmap {...}
182
183     private fun removeNoise(bitmap: Bitmap): Bitmap {...}
206
207     private fun saveMediaToStorage(bitmap: Bitmap) {...}
235 }
```

Εικόνα 6.3.21 Αναλυτικά το Optimized Image Activity

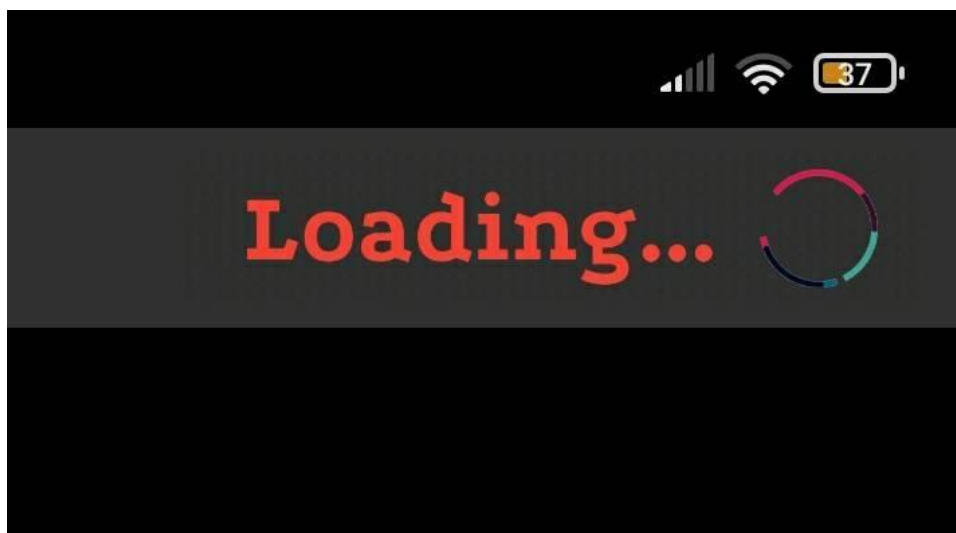
Επίσης, υπάρχει μια συνάρτηση που αποθηκεύει την βελτιστοποιημένη εικόνα στην συσκευή του χρήστη.

Υπάρχει ακόμα η συνάρτηση που αναλαμβάνει την αποθήκευση της εικόνας στην συσκευή του χρήστη.

Στο XML αρχείο της δραστηριότητας αυτής υπάρχουν τα ίδια κουμπιά που εξηγήθηκαν προηγουμένως στο Edit Image Activity αλλά αντί για το κουμπί που μεταφέρει τον χρήστη στο Optimized Image Activity, υπάρχει το κουμπί που ανοίγει το Saved Files Fragment.

Ουσιαστικά, μετά την βελτιστοποίηση ο χρήστης έχει την δυνατότητα να μεταφερθεί εκεί που βρίσκονται και οι υπόλοιπες επεξεργασμένες εικόνες για να μπορεί να προβάλλει μια ή παραπάνω στην οθόνη, να τις μετατρέψει και να τις μοιράσει σε άλλους.

Λόγω της πολυπλοκότητας των αλγορίθμων, υπάρχει μια μικρή καθυστέρηση μερικών δευτερολέπτων στο να εμφανιστεί στην οθόνη η τελική φωτογραφία. Για αυτό το λόγο, μέχρι να ολοκληρωθεί η βελτιστοποίηση εμφανίζεται ένα κινούμενο μήνυμα που ειδοποιεί τον χρήστη να περιμένει. (Εικόνα 6.3.22)



Εικόνα 6.3.22 Το μήνυμα που ειδοποιεί τον χρήστη ότι φορτώνει η εικόνα

Στο Saved Files Fragment δηλαδή, μπορεί ο χρήστης να μεταφερθεί από δικιά του επιλογή πατώντας την κατάλληλη καρτέλα στο Main Activity, είτε να οδηγηθεί από την επεξεργασία εικόνας.

- Preview Photo Activity

Ο ρόλος αυτής της δραστηριότητας είναι να προβάλλει τις εικόνες που έχουν υποστεί βελτιστοποίηση και να επιτρέπει στον χρήστη να τις μοιράσει στη μορφή που επιθυμεί.

```

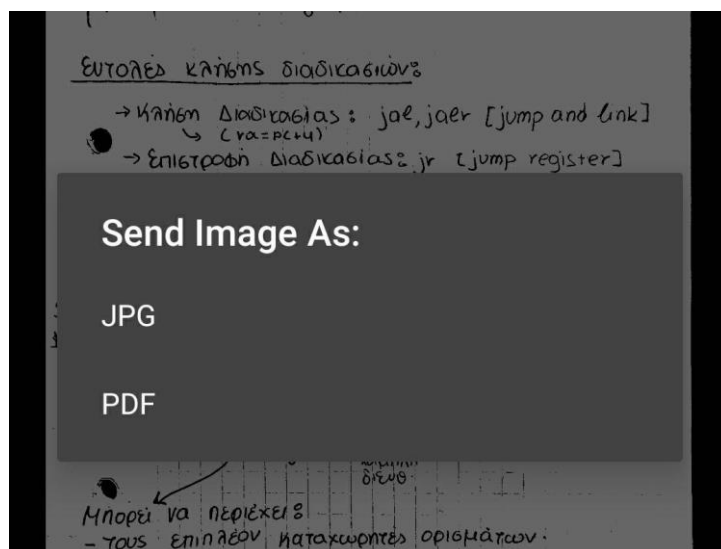
1 package com.example.imagescanner
2
3 import ...
29
30 class PreviewPhoto : AppCompatActivity() {
31     override fun onCreate(savedInstanceState: Bundle?) {...}
64
65     private fun shareJPG(view: View) {...}
78
79     var pdfDocument // to create .pdf file
80         : PdfDocument? = null
81
82     private fun sharePDF(view: View) {...}
129
130     private fun loadBitmapFromView(view: View): Bitmap? {...}
160
161 }

```

Εικόνα 6.3.23 Αναλυτικά το Preview Photo Activity

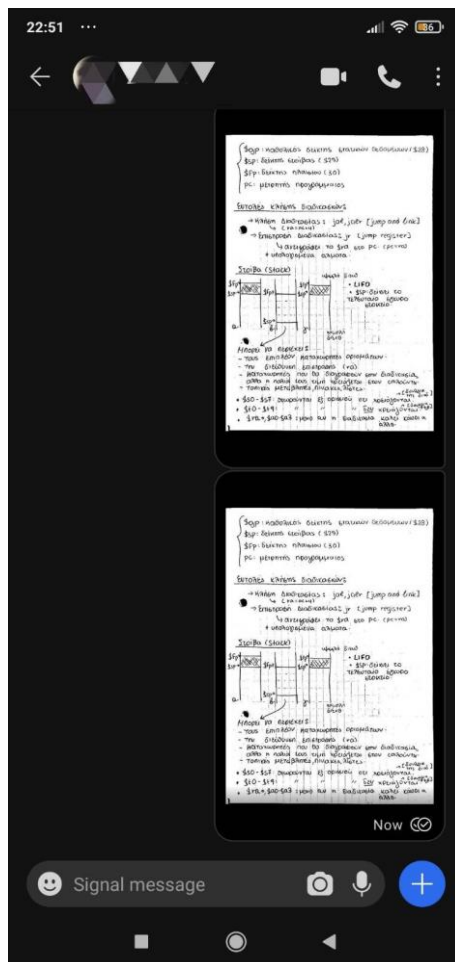
Όταν ο χρήστης βρίσκεται στο Saved Files Fragment βλέπει μπροστά του όλες τις εικόνες που έχουν επεξεργαστεί από την εφαρμογή. Μπορεί να πατήσει πάνω σε μια φωτογραφία και να μεταφερθεί στο Preview Photo Activity.

Σε αυτή τη δραστηριότητα υπάρχει ένα κουμπί που έχει να κάνει με την αποστολή της εικόνας. Όταν πατηθεί, εμφανίζεται ένα Alert Dialog που δίνει την επιλογή στον χρήστη να διαλέξει άμα θέλει να μοιράσει την εικόνα σε μορφή JPG ή σε μορφή PDF. (Εικόνα 6.3.24)



Εικόνα 6.3.24 Το Alert Dialog που εμφανίζει τις επιλογές αποστολής

Όπως αναφέρθηκε προηγουμένως, για την διαδικασία της αποστολής είναι απαραίτητη η μετατροπή της εικόνας σε bitmap. Η μετατροπή σε PDF έγγραφο γίνεται και πάλι με την βοήθεια της κλάσης PdfDocument. Και σε αυτή την περίπτωση εμφανίζονται όλες οι επιλογές για αποστολή σε μέσα κοινωνικής δικτύωσης και ηλεκτρονικού ταχυδρομείου. Για παράδειγμα, στην Εικόνα 6.3.25 μια επεξεργασμένη φωτογραφία στέλνεται σε μήνυμα σε συνομιλία μέσω μιας εφαρμογής κοινωνικής δικτύωσης.



Εικόνα 6.3.25 Αποστολή εικόνας JPG σε μήνυμα συνομιλίας εφαρμογής κοινωνικής δικτύωσης

- Preview Multiple Photos Activity

Αυτό το Activity έχει τις ίδιες ιδιότητες με το Preview Photo Activity αλλά έχει να κάνει με την επιλογή πολλαπλών φωτογραφιών. Όπως εξηγήθηκε προηγουμένως, στο Saved Files Fragment είναι δυνατό το “μαρκάρισμα” πολλών φωτογραφιών. Έτσι, όταν επιλεγθούν πολλές φωτογραφίες γίνεται δυνατό το μείρασμα τους. Μπορούν να σταλθούν είτε σαν εικόνες JPG, είτε σαν ένα ενιαίο PDF. (Εικόνα 6.3.28)

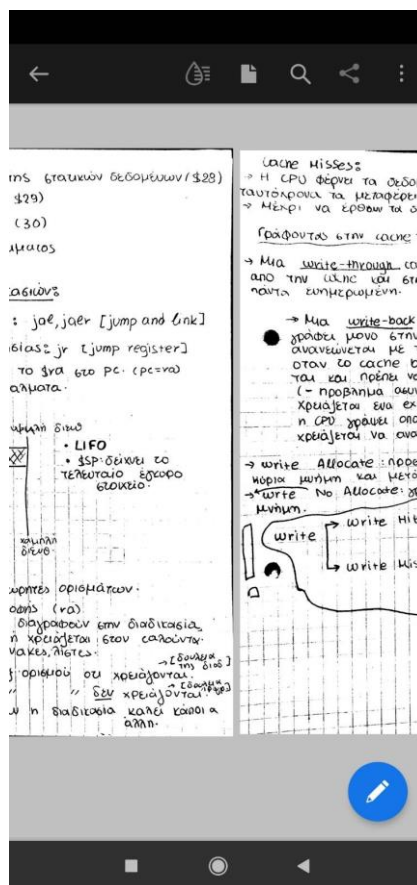
```

1 package com.example.imagescanner
2
3 import ...
20
21
22 class PreviewMultiplePhotos : AppCompatActivity() {
23     override fun onCreate(savedInstanceState: Bundle?) {...}
61
62     private fun shareJPG(arrayList: ArrayList<String>) {...}
76
77     private var pdfDocument // to create .pdf file
78         : PdfDocument? = null
79
80     private fun sharePDF(arrayList: ArrayList<String>) {...}
128
129 }

```

Εικόνα 6.3.27 Αναλυτικά το Preview Multiple Photos Activity

Όπως και με την περίπτωση του Preview Photo, υπάρχει και εδώ η παρουσία του Alert Dialog που δίνει την επιλογή στον χρήστη να διαλέξει σε τι μορφή θέλει να στείλει τα αρχεία του.



Εικόνα 6.3.28 Η δημιουργία του ενιαίου PDF που αποτελείται από πολλές επεξεργασμένες εικόνες

6.3.4 Επιπρόσθετες τεχνικές και έννοιες

Στον κώδικα που αναλύθηκε, χρησιμοποιήθηκαν διάφορες σημαντικές τεχνικές της γλώσσας Kotlin και του περιβάλλοντος Android.

Ειδικότερα:

- Intent

Η Intent είναι μια αφηρημένη περιγραφή μιας πράξης που πρέπει να εκτελεστεί. Η πιο σημαντική χρήση της είναι στην έναρξη δραστηριοτήτων. Είναι δηλαδή η εντολή που ευθύνεται για τη μεταφορά από το ένα Activity σε ένα άλλο και θα μπορούσε να χαρακτηριστεί ως η “κόλλα” μεταξύ των δραστηριοτήτων. [27]

Μέσω της Intent μπορείς να μεταφέρεις διάφορα δεδομένα μεταξύ των Activities, όπως εικόνες, strings, πίνακες κ.ο.κ. Αυτό επιτυγχάνεται μέσω της εντολής `putExtra()` η οποία έχει δύο παραμέτρους, η πρώτη καθορίζει το όνομα ή την ταμπέλα που βάζει ο χρήστης για να ξεχωρίζει τα δεδομένα και η δεύτερη είναι τα ίδια τα δεδομένα. Για παράδειγμα, στο Edit Image Activity, όταν πατηθεί το κουμπί που μεταφέρει στην επόμενη δραστηριότητα (Optimized Image Activity), πρέπει να σταλθεί επίσης η εικόνα που έχει υποστεί περικοπή ώστε να περάσει για τη βελτιστοποίηση. (Εικόνα 6.3.29) Για να λάβει η επόμενη δραστηριότητα τα δεδομένα χρησιμοποιεί το `getStringExtra()`.

```
val intent = Intent( packageContext: this, OptimizedImage::class.java)
intent.putExtra( name: "Cropped Image", mCroppedFile)
startActivity(intent)
```

Εικόνα 6.3.29 Η χρήση της Intent με `putExtra()`

Σαφώς, η Intent χρησιμοποιείται επίσης κάθε φορά που ξεκινάει ένα νέο Activity στην εφαρμογή. Παράλληλα, η εντολή αυτή χρειάζεται και για να ανοίξει η κάμερα ή η συλλογή φωτογραφιών χρησιμοποιώντας την ενέργεια `ACTION_IMAGE_CAPTURE` (ενέργεια για τράβηγμα φωτογραφίας) και `ACTION_PICK` (ενέργεια για διάλεγμα αρχείων) αντίστοιχα. Το αποτέλεσμα αυτής της Intent εφαρμόζεται στην συνάρτηση `OnActivityResult()`, όπου τα δεδομένα που επιλέχθηκαν (στην συγκεκριμένη περίπτωση η εικόνα) ανακτώνται μέσω της εντολής `getData()`.

Τέλος, μια άλλη χρήση που έχει η Intent στην εφαρμογή, είναι για το άνοιγμα του παραθύρου που περιλαμβάνει όλες τις επιλογές για το μοίρασμα αρχείων. (Εικόνα 6.3.14) Αυτό επιτυγχάνεται προσθέτοντας την ενέργεια `ACTION_SEND` (ενέργεια για αποστολή) και την “ταμπέλα” `EXTRA_STREAM` (επιπλέον ροή) στο `putExtra()` και δημιουργώντας ένα παράθυρο με επιλογές (Chooser) που περιλαμβάνει όλες τις εφαρμογές με τη δυνατότητα αποστολής αρχείων.

- Listener

Στο Android υπάρχουν περισσότεροι από ένας τρόποι για να παρακολουθούνται τα γεγονότα από την αλληλεπίδραση του χρήστη με την εφαρμογή. Το πρόγραμμα πρέπει να γνωρίζει πότε ο χρήστης πατάει ένα μέρος της οθόνης (π.χ. ένα κουμπί ή μια εικόνα) ώστε να ξεκινήσει μια ενέργεια ως αποτέλεσμα αυτού. Αυτή η διεπαφή πραγματοποιείται μέσω των EventListeners (ακροατές γεγονότων).

Στην εργασία χρησιμοποιούνται OnClickListener κάθε φορά που πατιέται ένα κουμπί. Για παράδειγμα ένα κουμπί μπορεί να μεταφέρει τον χρήστη σε μια καινούρια ή μια προηγούμενη δραστηριότητα. Σε πολλά Activities του Image Scanner υπάρχουν κουμπιά που όταν πατηθούν ενημερώνεται ο Listener και γίνεται μια ενέργεια όπως περιστροφή της εικόνας. (Εικόνα 6.3.30)

```
rightButton.setOnClickListener() { it: View!
    //rotate right the image
    ivCrop.animate().rotationBy( value: 90f).start()
}
leftButton.setOnClickListener() { it: View!
    //rotate left the image
    ivCrop.animate().rotationBy( value: -90f).start()
}
mirrorButton.setOnClickListener() { it: View!
    //mirror/flip the image
    ivCrop.animate().rotationYBy( value: 180f).start()
}
```

Εικόνα 6.3.30 Κουμπιά που οι Listeners κάνουν μια ενέργεια όταν ο χρήστης πατάει πάνω τους

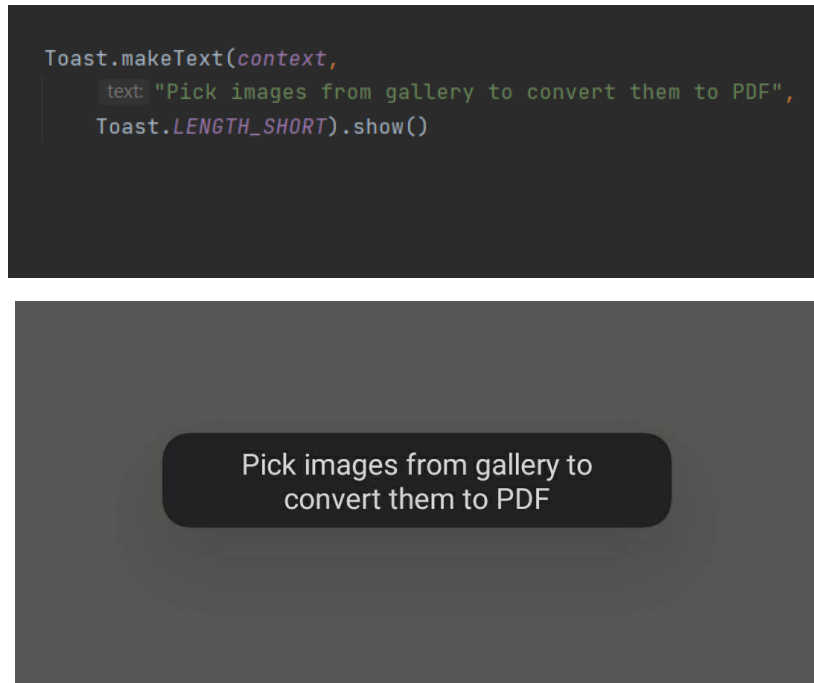
Επίσης, στο Saved Files Fragment που υπάρχει η δυνατότητα του χρήστη να “μαρκάρει” πολλές φωτογραφίες, υπάρχει η παρουσία των OnLongClickListeners. Δηλαδή, κάθε φορά που μια εικόνα πατιέται παρατεταμένα, μπαίνει σε ένα πίνακα ο οποίος τελικά θα περιλαμβάνει όλες τις φωτογραφίες που ο χρήστης επιθυμεί να στείλει μαζί.

- Toast

Το Toast είναι ένα μήνυμα που εμφανίζεται σε ένα μικρό αναδυόμενο παράθυρο. Γεμίζει μόνο τον χώρο που απαιτείται για το μήνυμα και το τρέχον Activity παραμένει ορατό και διαδραστικό. Το Toast έχει μικρή διάρκεια και εξαφανίζεται αυτόματα μετά από ένα σύντομο χρονικό περιθώριο. Χρησιμοποιείται με τη μέθοδο makeText() η οποία παίρνει σαν παραμέτρους το κείμενο που θα εμφανίζεται αλλά και το χρόνο που θα φαίνεται στην οθόνη.

Η εμφάνιση στην οθόνη επιτυγχάνεται μέσω της μεθόδου `show()`. [28] Είναι ένα πολύ βοηθητικό εργαλείο τόσο για τους προγραμματιστές όσο και για τους χρήστες.

Για παράδειγμα, στην εφαρμογή όταν ο χρήστης μεταβαίνει στο PDF Converter Fragment εμφανίζεται στην οθόνη ένα Toast που του εξηγεί την χρήση αυτής της καρτέλας. (Εικόνα 6.3.31)



Εικόνα 6.3.31 (α, β) Παράδειγμα χρήσης Toast στον κώδικα και στην οθόνη της εφαρμογής

- `findViewById`

Όπως έχει αναφερθεί προηγουμένως, τα Views μπορεί να χαρακτηρίζονται από μια ταυτότητα ή ID. Αυτό το όρισμα είναι απαιτούμενο όταν ο προγραμματιστής επιθυμεί να χρησιμοποιήσει το View σε ένα Activity ή Fragment για να του αναθέσει μια δουλειά.

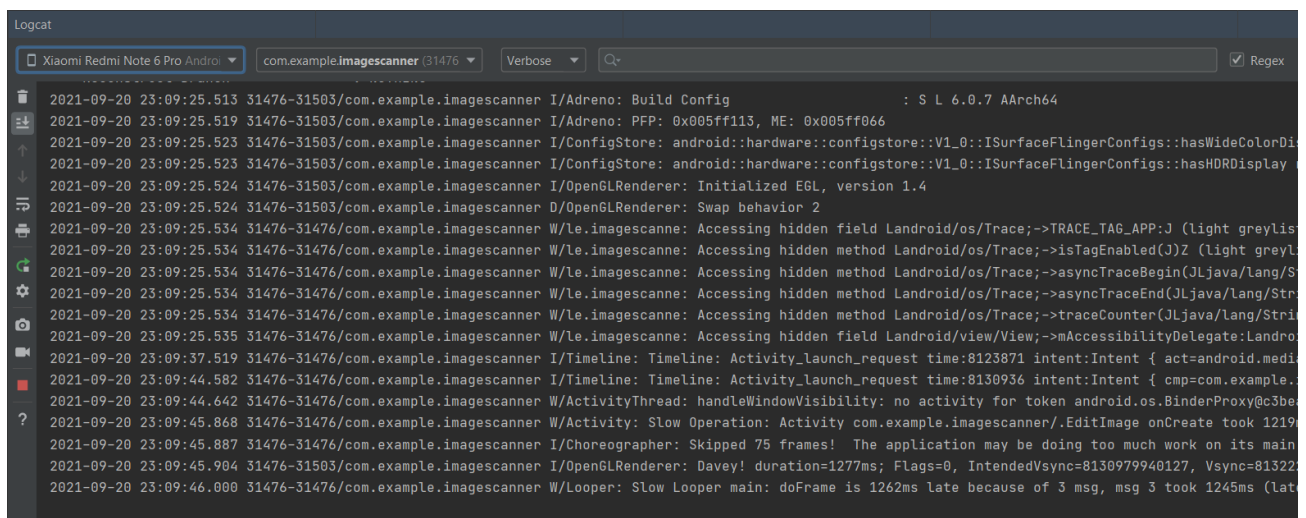
Η μέθοδος `findViewById()` επιστρέφει το View άμα ταιριάζει το ID του με το όνομα που βρίσκει η `getId()` ή null αν το ID είναι άκυρο/ δεν υπάρχει.

```
val rightButton = findViewById<ImageButton>(R.id.right_button)
val leftButton = findViewById<ImageButton>(R.id.left_button)
val mirrorButton = findViewById<ImageButton>(R.id.mirror_button)
val optimizeButton = findViewById<ImageButton>(R.id.optimize_button)
val retakeButton = findViewById<ImageButton>(R.id.retake_button)
```

Εικόνα 6.3.32 Κάποια κουμπιά που χρησιμοποιούνται στο Edit Image Activity και βρίσκονται μέσω της μεθόδου `findViewById`

- Log

Είναι ένα API (Application Programming Interface) του Android για εμφάνιση μηνυμάτων στο περιβάλλον του Android Studio που σχετίζονται με την κατάσταση του κώδικα. Όπως για παράδειγμα για το αν θα προκύψει κάποιο πρόβλημα, άμα μια διαδικασία πέτυχε ή απέτυχε, για την εμφάνιση της τιμής κάποιας μεταβλητής που χρειάζεται να ξέρει ο προγραμματιστής κ.ο.κ. Τα μηνύματα αυτά εμφανίζονται στην καρτέλα που έχει το Android Studio που ονομάζεται Logcat.



Εικόνα 6.3.33 Το Logcat στο Android

- URI

Το URI (Uniform Resource Identifier) είναι μια συμβολοσειρά χαρακτήρων που χρησιμοποιείται για τον προσδιορισμό ενός μέσου. Αυτός ο προσδιορισμός επιτρέπει την αλληλεπίδραση αυτού του μέσου με ένα δίκτυο.

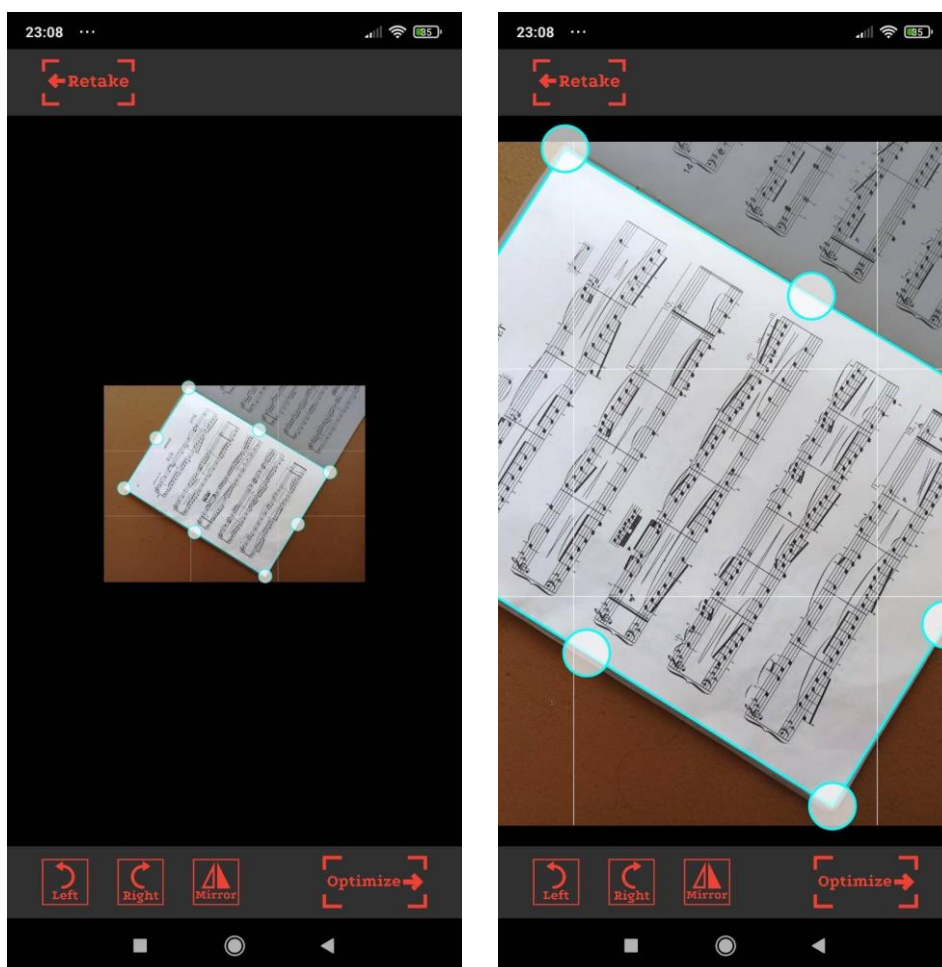
Στην εφαρμογή το URI είναι μια διεύθυνση που δείχνει σημαντικά αρχεία όπως φωτογραφίες και έγγραφα. Όταν ανοίγει η κάμερα και τραβιέται μια φωτογραφία ή όταν επιλέγεται από την συλλογή, αυτή χρησιμοποιείται στον κώδικα αφού μετατραπεί σε URI. Οι εικόνες μεταφέρονται από τη μια δραστηριότητα σε μια άλλη σε μορφή URI και επίσης όταν δημιουργούνται τα PDF αρχεία πάλι μετατρέπονται σε αυτή τη μορφή για να σταλούν ή να αποθηκευτούν. Είναι μια συμβολοσειρά χαρακτήρων που αντιπροσωπεύει το μονοπάτι των αρχείων που χρειάζονται συχνά στον κώδικα.

- ScaleGestureDetector

Το ScaleGestureDetector ανιχνεύει χειρονομίες που αλλάζουν την κλιμάκωση ενός αντικειμένου. Ο Listener OnScaleGestureListener ειδοποιεί όταν έχει συμβεί ένα συγκεκριμένο συμβάν χειρονομίας μέσω αφής. [29]

Στην εφαρμογή χρησιμοποιείται ο ScaleGestureDetector όταν προβάλλεται μια φωτογραφία και ο χρήστης επιθυμεί να την μεγεθύνει στην οθόνη, δηλαδή να κάνει “ζούμ”.

Για παράδειγμα, στην Εικόνα 6.3.34 η φωτογραφία που βρίσκεται στο Optimized Image Activity μεγεθύνεται όταν ανιχνεύεται η κατάλληλη κίνηση αφής από τον χρήστη. Έτσι, ο χρήστης έχει την δυνατότητα να βλέπει πιο καθαρά και πιο κοντά την εικόνα μπροστά του. Ο χρήστης πρέπει να κάνει “pinch” την οθόνη, δηλαδή να μετακινήσει τα δύο δάχτυλα του μεταξύ τους.



Εικόνα 6.3.34 Μεγέθυνση της εικόνας μέσω ScaleGestureDetector

- FileOutputStream

Το FileOutputStream δημιουργεί μια ροή εξόδου αρχείου για την εγγραφή δεδομένων σε ένα αρχείο.

Στην εφαρμογή συμμετέχει στην διαδικασία αποθήκευσης φωτογραφιών στην συσκευή του χρήστη. Αρχικά, δημιουργείται ένα File, ένα “αρχείο” μέσω της συνάρτησης File(). Αυτή παίρνει σαν όρισμα το μονοπάτι στο οποίο θα αποθηκευτεί η φωτογραφία, δηλαδή σε αυτή την περίπτωση εκεί που υπάρχουν όλες οι εικόνες στη συσκευή (Environment.DIRECTORY_PICTURES), καθώς και το μονοπάτι του “παιδιού” δηλαδή το όνομα του αρχείου που θα προστεθεί στο τελικό μονοπάτι. Μέσα σε αυτό το File γράφεται η εικόνα που θα αποθηκευτεί μέσω του FileOutputStream. Αυτό αποθηκεύεται σε μια μεταβλητή η οποία περνάει σαν όρισμα σε μια συνάρτηση που συμπιέζει ένα bitmap και έτσι και το αποθηκεύει στην συσκευή. (Εικόνα 6.3.35)

Το FileOutputStream συναντιέται και στο σημείο του κώδικα που αναλαμβάνει την αποστολή των εγγράφων. Όταν δημιουργούνται τα PDF αρχεία, χρησιμοποιώντας την κλάση PdfDocument(), σε κάθε σελίδα “γράφεται” μια εικόνα με τη μέθοδο FileOutputStream. Αυτή τη φορά το μονοπάτι που χρειάζεται είναι το μέρος που βρίσκονται όλα τα ληφθέντα αρχεία της συσκευής (Environment.DIRECTORY_DOWNLOADS).

- IOException

Το IOException σηματοδοτεί ότι έχει συμβεί κάποια εξαίρεση εισόδου/εξόδου. Αυτή η κλάση είναι η γενική κατηγορία κλάσεων που δημιουργούνται από αποτυχημένες ή διακοπτόμενες λειτουργίες εισόδου/εξόδου. [30]

Στην εφαρμογή χρησιμοποιείται σε περιπτώσεις που κάτι πάει στραβά και πρέπει ο προγραμματιστής να το γνωρίζει.

Για παράδειγμα, στην Εικόνα 6.3.35 απεικονίζεται μια διαδικασία αποθήκευσης εικόνας στην συσκευή. Στην περίπτωση που γίνει κάτι λάθος και η διαδικασία δεν πραγματοποιηθεί, εκτυπώνεται το σφάλμα και ο προγραμματιστής έχει μια καλύτερη εικόνα για το πρόβλημα που δημιουργήθηκε

```
private fun saveImage(bitmap: Bitmap, saveFile: File) {
    try {
        val fos = FileOutputStream(saveFile)
        bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, fos)
        fos.flush()
        fos.close()
    } catch (e: IOException) {
        e.printStackTrace()
    }
}
```

Εικόνα 6.3.35 Παράδειγμα χρήσης FileOutputStream() και IOException

- Canvas

Για να σχεδιαστεί ένα αρχείο, χρειάζονται τέσσερα βασικά στοιχεία: Ένα bitmap που περιλαμβάνει τα εικονοστοιχεία (πίξελς), έναν “καμβά” για να γράψει στο bitmap, ένα αρχικό σχέδιο και ένα χρώμα για το σχέδιο. Αυτά όλα περιλαμβάνονται στην κλάση Canvas. [31]

Η κλάση Canvas βρίσκεται στην εφαρμογή στο σημείο που δημιουργείται το PDF αρχείο. Σε κάθε σελίδα η συνάρτηση `getCanvas()` παίρνει τον “καμβά” της σελίδας και της τοποθετεί το σχέδιο που πρέπει να έχει, δηλαδή την εικόνα που θα μπει στο τελικό έγγραφο. Αυτό επιτυγχάνεται μέσω της συνάρτησης `drawBitmap()`. (Εικόνα 6.3.36)

```
//startPage
val page: PdfDocument.Page = pdfDocument!!.startPage(pageInfo)

val canvas = page.canvas
val paint = Paint()
paint.color = Color.parseColor( colorString: "#FFFFFF")
canvas.drawBitmap(scaledBitmap, 0.toFloat(), 0.toFloat(), paint: null)

//finishPage
pdfDocument!!.finishPage(page)
```

Εικόνα 5.3.36 Χρήση κλάσης Canvas

6.4 Μελλοντικές επεκτάσεις

Ενώ η εφαρμογή Image Scanner είναι πλήρως λειτουργική και επιφέρει άμεσα και ικανοποιητικά αποτελέσματα, υπάρχουν κάποια στοιχεία που θα μπορούσαν να προστεθούν στο μέλλον και να αυξήσουν τις δυνατότητες της. Κάποια από αυτά είναι:

- Χρήση κωδικού ασφάλειας για ευαίσθητα αρχεία
- Δυνατότητα αποθήκευσης προσωπικής υπογραφής και ενσωμάτωση στις εικόνες
- Αναγνώριση κειμένου με OCR (Optical Character Recognition)
- Προσθήκη παραπάνω εργαλείων για έγγραφα PDF όπως περικοπή, αφαίρεση/πρόσθεση σελίδων, αλλαγή σελίδων κ.ο.κ
- Παραπάνω επιλογές στους τύπους εξαγωγής αρχείων
- Δημιουργία προσωπικών λογαριασμών για πρόσβαση αρχείων και σε άλλες συσκευές

Παράλληλα, κάποια υπάρχοντα χαρακτηριστικά θα μπορούσαν να βελτιωθούν σε ένα βαθμό για ακόμα μεγαλύτερη απόδοση. Οι βελτιώσεις που θα μπορούσαν να εφαρμοστούν είναι οι εξής:

- Μείωση του χρόνου επεξεργασίας εικόνας
- Καλύτερη απόδοση σε έγχρωμες φωτογραφίες

ΚΕΦΑΛΑΙΟ 7

ΕΠΙΛΟΓΟΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

Αυτή η διπλωματική εργασία αφορά την ανάπτυξη μιας Android εφαρμογής για κινητές συσκευές που έχει σαν σκοπό να απλουστεύσει την διαδικασία σάρωσης φωτογραφιών και κειμένων. Ο στόχος της υλοποίησης της εφαρμογής ήταν η παροχή χρήσιμων εργαλείων στους χρήστες, συνδυάζοντας νέες τεχνολογίες και σημαντικές μεθόδους.

Τους τελευταίους μήνες, παρατηρήθηκε πως το Android Studio αποτελεί ένα πολύ εύχρηστο και λειτουργικό περιβάλλον. Η Kotlin είναι μια γλώσσα που προσφέρει σημαντικά βοηθήματα και ανεβάζει το επίπεδο του προγραμματισμού. Η αμεσότητα που προσφέρεται από αυτά τα δύο καθιστά την διαδικασία εκτέλεσης εξαιρετικά φιλική προς τον προγραμματιστή.

Παράλληλα, επισημάνθηκε η σημαντικότητα της επεξεργασίας και της βελτιστοποίησης εικόνας στην διαδικασία ανάγνωσης κειμένων. Μια φωτογραφία μπορεί να απαλειφθεί από δυσαναλογίες και περιττά σημεία και με τα κατάλληλα φίλτρα να γίνει πιο ευκρινής και καθαρή.

Γενικά, η χρήση της εφαρμογής Image Scanner που συνδύασε την επιστήμη του προγραμματισμού και της επεξεργασίας εικόνας παρατηρήθηκε πως επιφέρει πολλά χρήσιμα αποτελέσματα στους ανθρώπους που επιζητούν γρήγορη σάρωση εικόνων. Οι φωτογραφίες και τα κείμενα που επεξεργάστηκαν μέσω της εφαρμογής έχουν ικανοποιητικά αποτελέσματα και βελτιώνουν σημαντικά την ευκρίνεια τους. Παράλληλα, η γρήγορη αποστολή των αρχείων μέσω της εφαρμογής, φαίνεται να προσφέρει την ζητούμενη αμεσότητα. Τέλος, το χαρακτηριστικό που θα μπορούσε να έχει καλύτερη επίδοση είναι ο χρόνος αναμονής για την εφαρμογή των φίλτρων στην εικόνα, ο οποίος κρατάει μερικά δευτερόλεπτα λόγω της πολυπλοκότητας των αλγορίθμων.

ΠΑΡΑΡΤΗΜΑ

ΕΓΚΑΤΑΣΤΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΜΕΛΕΤΗ ΤΟΥ ΚΩΔΙΚΑ

Ο κώδικας της εφαρμογής βρίσκεται στο λογαριασμό μου στο Github στο παρακάτω link:

<https://github.com/sgkevrecki/ImageScanner>

Το περιβάλλον του Android Studio μπορεί να κατέβει και να εγκατασταθεί από το παρακάτω link:

<https://developer.android.com/studio>

Ο κώδικας μπορεί να επεξεργαστεί στο Android Studio με την παρακάτω διαδικασία:

- Ανοίγεται η καρτέλα “File”
- Πατιέται New > Project from Version Control...
- Στο πεδίο URL συμπληρώνεται το παρακάτω link:

`https://github.com/sgkevrecki/ImageScanner.git`

- Πατιέται το κουμπί “Clone”

Η εγκατάσταση της εφαρμογής Image Scanner μπορεί να επιτευχθεί σε Android συσκευές μέσω του αρχείου app-debug.apk.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] *Google Play: CamScanner [online]*. Ανάκτηση από:
<https://play.google.com/store/apps/details?id=com.intsig.camscanner&hl=en&gl=>
- [2] *Βικιπαίδεια: Σαρωτής [online]*. Ανάκτηση από:
<https://el.wikipedia.org/wiki/%CE%A3%CE%B1%CF%81%CF%89%CF%84%CE%AE%CF%82>
- [3] *Preservationtutorial.library.cornell.edu: HOW SCANNERS WORK [online]*. Ανάκτηση από:
<http://preservationtutorial.library.cornell.edu/technical/technicalB-02.html>
- [4] *Techwalla.com: What are the uses of scanners? [online]*. Ανάκτηση από:
<https://www.techwalla.com/articles/what-are-the-uses-of-scanners>
- [5] *Techopedia.com: Mobile Application (Mobile app) [online]*. Ανάκτηση από:
<https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [6] *Investopedia.com: Android Operating System [online]*. Ανάκτηση από:
<https://www.investopedia.com/terms/a/android-operating-system.asp>
- [7] *Android Developer: Meet Android Studio [online]*. Ανάκτηση από:
<https://developer.android.com/studio/intro>
- [8] *Wikipedia: Android Studio [online]*. Ανάκτηση από:
https://en.wikipedia.org/wiki/Android_Studio
- [9] *IndianAppDevelopers: The Major Advantages of Android Studio App Development [online]*. Ανάκτηση από:
<https://www.indianappdevelopers.com/blog/advantages-of-android-studio-app-development/>
- [10] *Android Developers: Studio [online]*. Ανάκτηση από:
<https://developer.android.com/studio>
- [11] *Wikipedia: Kotlin Language [online]*. Ανάκτηση από:
[https://en.wikipedia.org/wiki/Kotlin_\(programming_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language))
- [12] *Sagara Technology Idea Lab: What are The Benefits of Kotlin [online]*. Ανάκτηση από:
<https://sagaratechnology.medium.com/what-are-the-benefits-of-kotlin-d7fdcd1cfc0>
- [13] *DZone: 8 Reasons to Use Kotlin Over Java for Android Development [online]*. Ανάκτηση από:
<https://dzone.com/articles/what-are-the-biggest-advantages-of-kotlin-over-jav>

- [14] *Android Developers: Introduction to Activities [online]*. Ανάκτηση από:
<https://developer.android.com/guide/components/activities/intro-activities>
- [15] *Android Developers: Activity [online]*. Ανάκτηση από:
<https://developer.android.com/reference/android/app/Activity>
- [16] *Android Developers: Fragment [online]*. Ανάκτηση από:
<https://developer.android.com/reference/androidx/fragment/app/Fragment.html>
- [17] *Android Developers: Layout [online]*. Ανάκτηση από:
<https://developer.android.com/guide/topics/ui/declaring-layout>
- [18] *Android Developers: Manifest [online]*. Ανάκτηση από:
<https://developer.android.com/guide/topics/manifest/manifest-intro>
- [19] *Android Developers: Configure your build [online]*. Ανάκτηση από:
<https://developer.android.com/studio/build#kts>
- [20] *Wikipedia: JPEG [online]*. Ανάκτηση από:
<https://en.wikipedia.org/wiki/JPEG>
- [21] *Wikipedia: PDF [online]*. Ανάκτηση από:
<https://en.wikipedia.org/wiki/PDF>
- [22] *Wikipedia: Edge Detection [online]*. Ανάκτηση από:
https://en.wikipedia.org/wiki/Edge_detection
- [23] *Wikipedia: Digital Image Processing [online]*. Ανάκτηση από:
https://en.wikipedia.org/wiki/Digital_image_processing
- [24] *Wikipedia: Noise Reduction [online]*. Ανάκτηση από:
https://en.wikipedia.org/wiki/Noise_reduction
- [25] *Android Developers: RecyclerView [online]*. Ανάκτηση από:
<https://developer.android.com/guide/topics/ui/layout/recyclerview>
- [26] *Android Developers: Dialog [online]*. Ανάκτηση από:
<https://developer.android.com/guide/topics/ui/dialogs>
- [27] *Android Developers: Intent [online]*. Ανάκτηση από:
<https://developer.android.com/reference/android/content/Intent>
- [28] *Android Developers: Toasts [online]*. Ανάκτηση από:

- <https://developer.android.com/guide/topics/ui/notifiers/toasts>
- [29] *Android Developers: ScaleGestureDetector [online]*. Ανάκτηση από:
<https://developer.android.com/reference/android/view/ScaleGestureDetector>
- [30] *Android Developers: IOException [online]*. Ανάκτηση από:
<https://developer.android.com/reference/java/io/IOException>
- [31] *Android Developers: Canvas [online]*. Ανάκτηση από:
<https://developer.android.com/reference/android/graphics/Canvas>
- [32] *noteboox.de: HP ScanJet N9120 Document Flatbed Scanner - flatbed scanner [online]*.
Ανάκτηση από:
<https://noteboox.de/HP-ScanJet-N9120-Document-Flatbed-Scanner-flatbed-scanner-EN>
- [33] *HelpDeskGeek: How to Restore Old or Damaged Photos Using Digital Tools [online]*.
Ανάκτηση από:
<https://helpdeskgeek.com/how-to/how-to-restore-old-damaged-photos-using-digital-tools/>
- [34] *Hongkiat: "Hello World" in 20 Programming Languages [online]*. Ανάκτηση από:
<https://www.hongkiat.com/blog/hello-world-different-programming-languages/>
- [35] *KatatanPeneliti [online]*. Ανάκτηση από:
<https://catatanpeneliti.wordpress.com/2013/06/04/empat-tipe-dasar-citra-digital/>