Mobilna aplikacija za pretvorbu teksta sa slike u tekst

Andrišek, Kristina

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:883091

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-06**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

MOBILNA APLIKACIJA ZA PRETVORBU TEKSTA SA SLIKE U TEKST

Diplomski rad

Kristina Andrišek

Osijek, 2017.

Sadržaj

1.UVOD 2

1.1 Zadatak diplomskog rada 2

2. RAZVOJNE TEHNOLOGIJE I OKRUŽENJE 2

2.1. Android 2

2.1.1. Arhitektura 2

2.1.2. Verzije Androida 4

2.2. Android Studio 6

2.3. Java 6

2.4.Datoteka XML 7

3.OCR 9

3.1. Tesseract OCR 9

3.1.1. Povijest Tesseract-a 9

3.1.2. Arhitektura Tesseract-a 10

4. IZRADA APLIKACIJE 11

4.1. Struktura projekta 11

4.2. Baza podataka 12

4.3.Kod za detekciju teksta 14

4.4. Korištenje aplikacije 14

5. TESTIRANJE APLIKACIJE 16

5.1. Uspješno prepoznat tekst sa slike s malom greškom 16

5.2. Potpuno uspješno prepoznavanje teksta 17

5.3. Usporedba prepoznavanja iste slike različite kvalitete 18

5.4. Prepoznavanje slike bez teksta 19

5.5. Prepoznavanje teksta sa slike koja nema podlogu u bijeloj boji 20

5.6. Prepoznavanje teksta sa plastificirane slike koja nema podlogu u bijeloj boji 21

5.7.Editiranje teksta 22

5.8. Spremanje teksta 23

6. ZAKLJUČAK 25

LITERATURA 26

SAŽETAK 28

ABSTRACT 29

ŽIVOTOPIS 30

PRILOZI 31

1.UVOD

Pametni telefoni bilježe porast prodaje na tržištu, te bivaju sve više zastupljeni u svakodnevici čovjeka. Od svojih osnovnih funkcija uspostave poziva, primanja i slanja SMS poruka pametni telefon prerastao je u pokretno računalo koje osim vlastitog operacijskog sustava posjeduje i mnoštvo različitih funkcija i aplikacija. Upravo ta praktičnost i nebrojene mogućnosti dovele su do nezaustavljivog rasta razvoja aplikacija za pametne telefone.

Bez obzira na široku ponudu aplikacija na tržištu, i dalje postoji potražnja za novim idejama. Jedna takva je upravo tema ovog diplomskog rada.

1.1 Zadatak diplomskog rada

U okviru ovog diplomskog rada potrebno je izraditi aplikaciju za Android operacijski sustav koja će pretvarati fotografiju cijene proizvoda s nazivom proizvoda od interesa u tekstualnu datoteku. Oznaka cijene proizvoda i naziva proizvoda nalazi se ispred, ispod ili pokraj proizvoda. Često se nazivi proizvoda zapisuju kraticama koje je teško zapamtiti i lako pomiješati sa sličnim kraticama, a točan naziv nam je važan zbog provjere stvarne vrijednosti proizvoda na Internetu ili provjere kvalitete istog.

U Android aplikaciji potrebno je omogućiti učitavanje slike, postupak prepoznavanja teksta, mogućnost korekcije prepoznatog teksta, te snimanje prepoznatog teksta kao tekstualne datoteke.

2. RAZVOJNE TEHNOLOGIJE I OKRUŽENJE

2.1. Android

Android je operacijski sustav razvijen od strane Google kompanije i baziran je na Linux kernelu. Operacijski sustav Android prvenstveno je dizajniran za mobilne uređaje osjetljive na dodir.

Android je cjelokupna platforma otvorenog koda i samim time je popularan kod tehnoloških tvrtki koje zahtijevaju gotov, jeftin, a opet prilagodljiv sustav. Njegova otvorena priroda potaknula je veliku zajednicu razvojnih inženjera i entuzijasta da koriste otvoreni kod kao podlogu za projekte kojima isporučuju ažuriranja starijim uređajima te dodaju nove značajke naprednim korisnicima.[1]

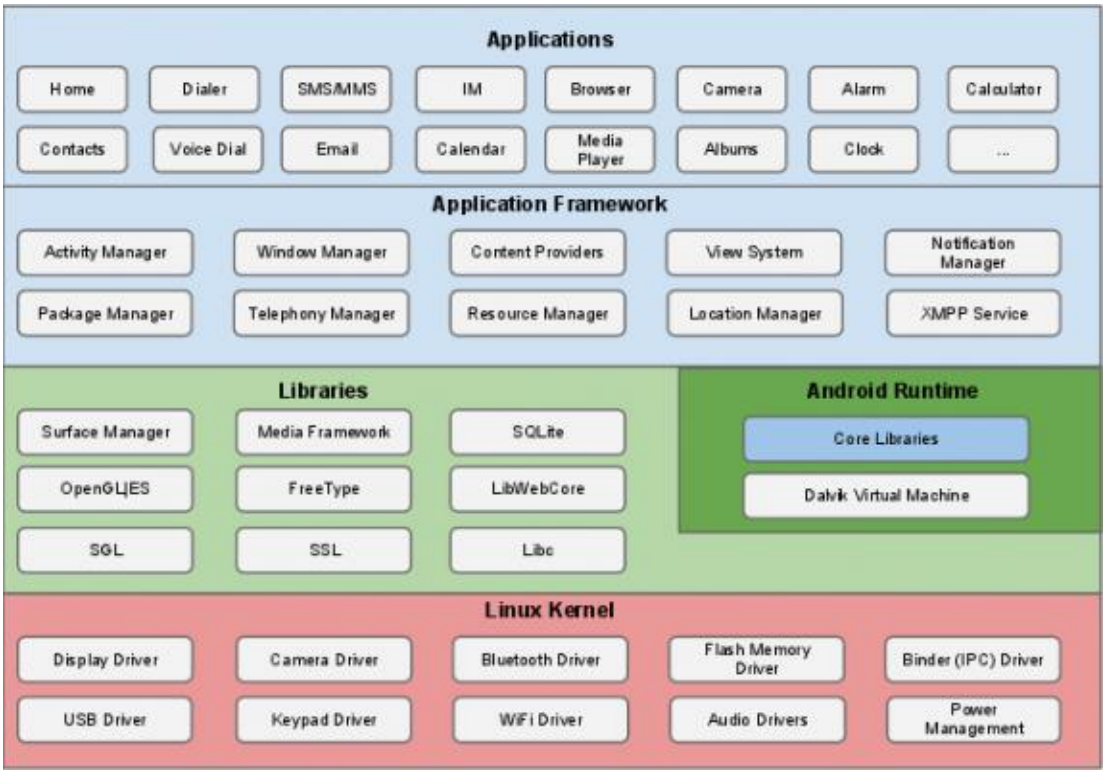
Android uređaji već su nekoliko godina najprodavaniji na tržištu mobilnih uređaja. Na slici 2.1. prikazana jedna od najzastupljenijih verzija Android 6.0 Marshmallow .

Slika 2.1. Prikaz sustava Android 6.0 Marshmallow

2.1.1. Arhitektura

Budući da je Android od samih svojih početaka osmišljen kao projekt otvorenog koda, aplikacije pomoću middlewara komuniciraju i pokreću neke druge aplikacije. Primarno su programski jezici C i C++ upotrebljavani za radno okruženje , no većina aplikacija je zapravo

nastala u Java programskom jeziku koristeći Android SDK. Arhitekturu operacijskog sustava Android možemo promatrati kao stog koji je podijeljen na pet razina.[2] Razine su prikazane na slici 2.1.1.



Slika 2.1.1. Arhitektura operacijskog sustava Android, [3]

Kao što se vidi na slici, na dnu stoga se nalazi Linux-ova jezgra (engl. Linux Kernel). Jezgra sadrži upravljačke programe koji brinu o upravljanju memorijom, komunikaciji između procesa, napajanju, mrežnim sučeljima i ostalim sustavima na sklopovskom nivou.

Iznad jezgre nalaze se biblioteke koje su pisane u C/C++ programskom jeziku.

Slijedeći sloj je za pokretanje aplikacija. Sastoji se od dvije komponente. Prva komponenta su biblioteke koje sadrže većinu jezgrenih biblioteka koje omogućuju pisanje Android aplikacija u programskom jeziku Java. Druga komponenta je virtualni stroj Dalvik (engl. Dalvik Virtual Machine, skraćeno DVM) koja je dizajnirana specijalno za Android i optimizirana je za uređaje s baterijama, limitiranom memorijom i procesorskom snagom.

Nakon knjižnica dolazi aplikacijski okvir koji se sastoji od mehanizama koji olakšavaju pisanje aplikacija. Aplikacijski okvir dozvoljava upotrebu svih API-ja koji su korišteni za bazne aplikacije. Na taj način je omogućeno upravljanje programskim paketima, aktivnostima aplikacije (odnosi se na životni ciklus aplikacije), pozivima, prozorima, resursima (pohrana komponenti aplikacija koje nisu sami kôd, primjerice slike), korištenje podataka od više različitih aplikacija, dohvaćanje i korištenje trenutne lokacije korisnika, prikaz obavijesti te baza pogleda i objekata koji mogu biti korišteni za dizajn aplikacije.

Na samom vrhu nalaze se aplikacije. Ovaj sloj je vidljiv krajnjem korisniku i sastoji se od skupa osnovnih aplikacija, uključujući e-mail klijent, SMS program, kalendar, karte, preglednik, kontakte i druge aplikacije koje se mogu pronaći u Android Trgovini .[4]

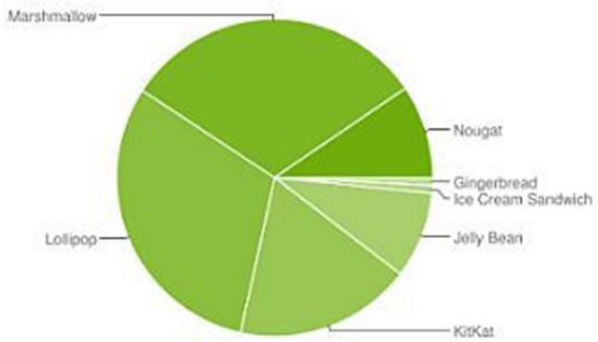
2.1.2. Verzije Androida

Android OS stalno se razvija i često se dodaju nove značajke. Počevši od prvog komercijalnog Android uređaja u rujnu 2008. , operacijski sustav prošao je kroz nekoliko verzija. Trenutna verzija je 7.0 "Nougat", objavljena u kolovozu prošle godine. Do svibnja 2017. Android ima dvije milijarde mjesečnih aktivnih korisnika. Verzije su navedene u tablici 2.1.2 [5] . Tablica prikazuje verziju Androida, prikazuje njegovo kodno ime i navodi njegov datum plasiranja na tržite.

Inačica	Kodno ime	Datum izlaska
Beta		5. studenoga 2007.
1.0	Apple Pie	23. rujna 2008.
1.1	Banana Bread	9. veljače 2009.
1.5	Cupcake	30. travnja 2009.
1.6	Donut	15. rujna 2009.
2.0/2.1	Eclair	26. listopada 2009.
2.2	Froyo	20. svibnja 2010.
2.3.x	Gingerbread	6. prosinca 2010.
3.x	Honeycomb	22. veljače 2011.
4.0.x	Ice Cream Sandwich	19. listopada 2011.
4.1/4.2/4.3	Jelly Bean	9. srpnja 2012., 4.2.1 - 27. studenoga 2012.
4.4	KitKat	3. rujan 2013.
5.0/5.1	Lollipop	25. lipnja 2014.
6.0.x	Marshmallow	5. listopad 2015.[1]
7.0	Nougat	22. kolovoza 2016.

Tablica 2.1.2. Android verzije

Bez obzira na zadnju verziju Google Android operativnog sustava i dalje su najzastupljenije verzije "Lollipop" (5.0) i "Marshmallow" (6.0). Zastupljenost ostalih verzija trenutno na tržištu prikazana je na slici 2.1.2



Slika 2.1.2. Zastupljenost Android verzija na tržištu, [6]

Imena verzija Android sustava su posebno zanimljiva jer im Google daje imena po slatkišima što se može vidjeti na slici 2.1.2. Neki od popularnih „slatkiša“ su Gingerbread (2.3), Ice Cream Sandwich (4.0), Nougat(7.0) .

Slika 2.1.2 Imena Android inačica, [7]

2.2. Android Studio

Android Studio je integrirano razvojno okruženje (IDE) za Android platformu. Najavljen je 16. svibnja 2013. na Google I / O konferenciji. Bio je u početnoj fazi pregleda, počevši od verzije 0.1 u svibnju 2013., a zatim je ušao u beta fazu počevši od verzije 0.8 koja je objavljena u lipnju 2014. Prva stabilna gradnja objavljena je u prosincu 2014., počevši od verzije 1.0. [8]

Android Studio dostupan je za preuzimanje na Windows, Mac OS i Linux. Dolaskom na tržište zamijenio je Eclipse ADT razvojni alat za aplikacije koji bio glavni alat za stvaranje aplikacija za Android. Android Studio nudi niz značajki koje poboljšavaju produktivnost prilikom razvoja Android aplikacije. Neke od niza značajki su:[9]

- Razvojni alati
- Bogat značajkama i brz emulator
- Jedinstveno okruženje pogodno za razvoj svih Android uređaja
- Opsežni alati za testiranje i okviri
- podrška za C + +

Budući da je Android Studio pisan u programskom jeziku Java na kojem se temelji i programiranje za Android, za razvoj aplikacija Android Studio zahtijeva instalaciju najnoviji Java Development Kit-a (JDK).

2.3. Java

Java je objektno orijentirani programski jezik razvijen od strane tima koji je predvodio James Goslingom u tvrtki Sun Microsystems. Razvoj je počeo 1991. kao dio projekta Green, a objavljen je u studenom 1995.

Prednost Jave u odnosu na neke druge programske jezike je u tome što se aplikacija nastala u programskom jeziku Java može pokrenuti na svim operacijskim sustavima za koje postoji JVM. Budući da posjeduje široki spektar klasa za rad s mrežnim komunikacijama ,velikoj većini programera Java je najbolji izbor za programiranje aplikacija.

U Java programskom jeziku su objektno-orijentirani principi obavezni. Sve je u Javi objekt, a sav izvorni kod je pisan unutar klasa.

Od 2016. Java je jedan od najpopularnijih programskih jezika u upotrebi posebno za web aplikacije klijent-poslužitelj, s prijavljenim 9 milijuna programera . U skladu sa svim navedenim Java se smatra osnovni jezik za programiranje sustava Android. [10]

2.4.Datoteka XML

XML je jezik za označavanje_podataka i dokumenata. Ideja je bila stvoriti jedan jezik koji će biti jednostavan za čitanje i razumijevanje ljudima koji se ne bave programiranjem i samim računalnim programima. XML je zamišljen tako da podatke čuva u tekstualnom obliku, ali da ih uokviruje oznakama koje govore o kakvom se podatku radi. Na taj način je XML jednostavno čitljiv čovjeku (tekst), a računalo jednostavno nalazi potrebne podatke koristeći oznake (tagove).

Ciljevi XML-a naglašavaju jednostavnost, općenitost i upotrebljivost preko Interneta. To je tekstualni format podataka s jakom podrškom preko Unicode-a za različite ljudske jezike. Iako je dizajn XML-a usredotočen na dokumente, koristi se za druge namjene: odvajanje podataka od prezentacije, razmjenu podataka, pohranu podataka i povećavanje dostupnosti podataka. XML je standardizirani jezik i za njegovu standardizaciju brine se World Wide Web Consortium. [11]

```
<?xml version="1.0" encoding="UTF-8" ?>
<memo>
    <contact type="person">
        <name>Ivo Ivić</name>
        <mobile>098 1234567</mobile>
        <email>ivo.ivic@fer.hr</email>
    </contact>
    <contact type="company">
        <name>Best Corporation</name>
        <mobile>091 123456</mobile>
        <email>best@best.com</email>
    </contact>
    <contact type="person">
        <name>Josip Horvat</name>
        <mobile>098 123456</mobile>
        <email>josip.horvat@fer.hr</email>
    </contact>
</memo>
```

Slika 2.5. Primjer XML koda, [12]

Na slici 2.5. vidljivo je da se XML dokument sastoji od 2 dijela. Prvi dio je označen plavom bojom i predstavlja prolog ili zaglavlje. U zaglavlju se navode podatci koji opisuju XML dokument. Drugi dio je označen crvenom bojom i predstavlja sadržaj dokumenta u kojem se nalazi korisni sadržaj. Sadržaj dokumente omeđen je XML oznakama. [12]

3.OCR

Optičko prepoznavanje znakova (eng. Optical Character Recognition) je računalni softver koji prevodi sliku otisnutog teksta s papira u tekst koji se može uređivati za daljnju obradu. OCR je vrlo korisna i popularna metoda u raznim primjenama.

Danas je na tržištu dostupno nekoliko verzija OCR-a poput Desktop OCR, Server OCR , Web OCR i drugi. Točnost bilo koje OCR verzije varira između 71 % do 98 %. Ponekad je teško dohvatiti tekst iz slika zbog različite veličine, stila, orijentacije, pozadine slike itd. Većina OCR alata je dostupna, no samo su neki od njih otvorenog koda i besplatni. Tesseract je upravo jedan od njih.

3.1. Tesseract OCR

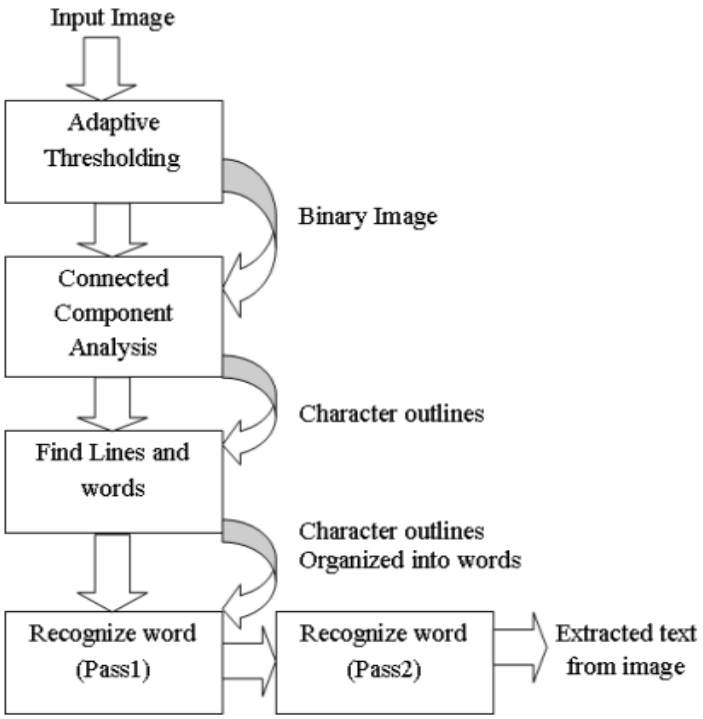
Tesseract OCR je besplatni softver otvorenog koda za prepoznavanje optičkih znakova za različite operacijske sustave. Napisan je u C++ programskom jeziku i može se koristiti u drugim aplikacijama u obliku DLL biblioteke (eng.Dynamic Link Library). Na taj način može se lako dodati kao referenca DLL-a u nekoj drugoj aplikaciji i koristiti sve svoje funkcionalnosti. [15]

3.1.1. Povijest Tesseract-a

Tesseract je razvijen u Hewelt Packard laboratoriju između 1984. i 1994. Već 1995. je izmijenjen i povećana mu je točnost. Krajem 2005. Hewelt Packard je objavio Tesseract kao alat otvorenog koda. Trenutačno je dostupna samo inačica naredbe baze. Verzija 3.01 je izdana i dostupana za upotrebu. Hewelt Packard nikada nije koristio Tesseract kao alat za vlastita istraživanja. Od 2006. Tesseract dalje razvija i održava Google. Najnovija stabilna verzija je 3.05.01, objavljena 1. lipnja 2017. Najnoviji izvorni kod 3.05 dostupan je od brancha 3.05 na GitHub-u. [14]

3.1.2. Arhitektura Tesseract-a

Tesseract OCR radi korak po korak kao po blok dijagramu prikazanom na slici 3.1.2.. Prvi korak je pretprocesiranje slike, koji pretvara sliku u binarnu sliku. Sljedeći korak je analiza koja se koristi za izlučivanje značajki. Ova metoda je vrlo korisna prilikom obrade OCR slike s bijelim tekstom i crnom pozadinom. Tesseract je vjerojatno bio prvi koji je osigurao ovu vrstu obrade. Zatim, obrisi se pretvaraju u blobove. Blobovi su organizirani u tekstualne linije, a linije i regije analiziraju se za neko fiksno područje ili ekvivalentnu veličinu teksta . Tekst je podijeljen na riječi pomoću određenih prostora i fuzzy prostora. Prepoznavanje teksta započinje dvosmjernim postupkom, odnosno fazom odlučivanja kao što je prikazano na slici 3.1.2. Zadatak faze odlučivanja unutar sustava za raspoznavanje tiskanog teksta jest izgraditi klasifikator koji će razvrstati nepoznate uzorke u odgovarajuće razrede. Evaluacija sustava je posljednji dio u sustavu za raspoznavanje tiskanog teksta. U ovoj fazi sustav je već odredio optimalni klasifikator za vektore značajki uzoraka za učenje. Cilj ove faze, evaluacije sustava jest odrediti koliko sustav griješi, odnosno koliko je točan pri raspoznavanju nepoznatih uzoraka . [15]



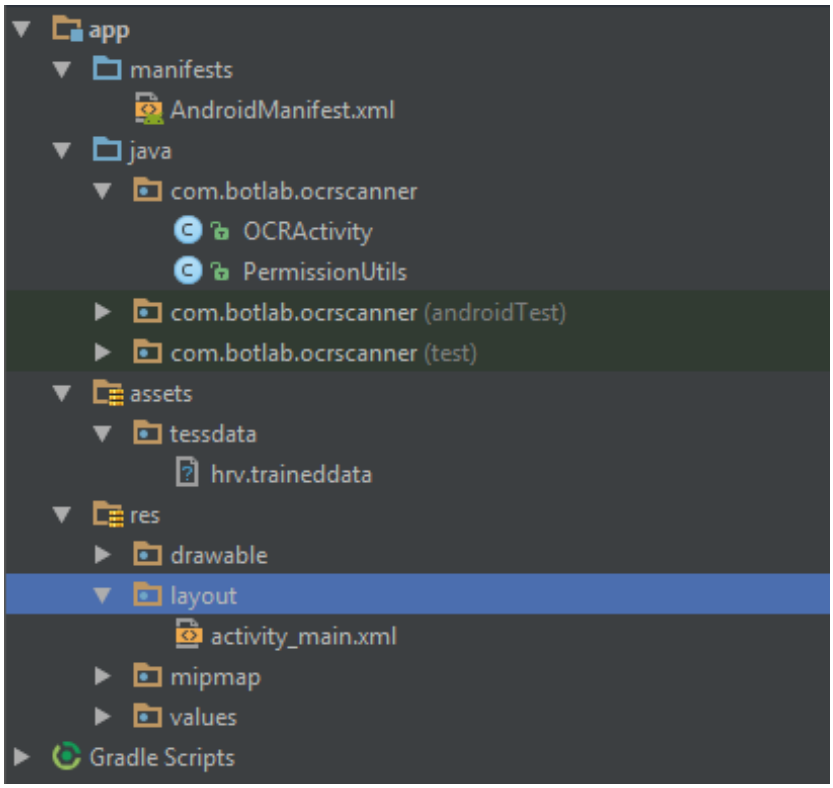
Slika 3.3.2. Tesseract OCR arhitektura, [14]

4. IZRADA APLIKACIJE

4.1. Struktura projekta

Rješenje problematike ovog završnog rada ostvareno je objektno orijentiranim programskim jezikom Java. Korišten je Android Studio sa svim alatima koji su potrebni za uspješno programiranje aplikacija s operativnim sustavom Android. Projekt se sastoji od:

- java/ : sadrži izvorne kodove u Javi;
- manifests/ : sadrži informacije o sustavu;
- assets/ : sadrži rječničke datoteke potrebne za detekciju teksta;
- res/ : sadrži ikone, definicije grafičkih korisničkih sučelja i datoteke XML;



Slika 4.1. Sadržaj projekta OCRScanner.

OCRScanner u AndroidManifest.xml datoteci sadržava nekoliko bitnih stvari o svojstvima aplikacije koji određuju njezinu funkcionalnost. Jedna od bitnih stvari je informacija

koje dozvole sadrži aplikacija. Bez navođenja ovih dozvola ne bi se moglo uspješno implementirati rješenje aplikacije .Osim navedenoga, datoteka sadrži još informacija i osnovnih elemenata. Dozvole koje aplikacija koristi prikazane su u sljedećem kodu.

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

Programski kod layout-a ili prikaza sučelja sastoji se od par osnovnih elemenata kao što su:

- ImageView – okvir za prikaz slike;
- Button – tipka koja na klik izvršava određeni dio koda;
- EditText - okvir za prikaz i uređivanje teksta

Primjeri elemenata prikazani su u kodu ispod.

```
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView"
     android:src="@drawable/icon"/>
```

Primjeri Button-a prikazani su u kodu ispod.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Camera"
    android:id="@+id/Camerabutton"
    android:textSize="18dp"
    android:layout_alignParentLeft="true"
    android:layout_centerHorizontal="true" />
```

Primjeri *EditText-a* prikazani su u kodu ispod.

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="OCR Text will appear here"
    android:id="@+id/OCRTextView"
    android:textSize="18dp"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true"
    android:background="#dedede"
    android:textColor="#a3a3a3" /
```

4.2. Baza podataka

Kao bazu podataka za detekciju teksta korišten je tesseract data datoteka datotečnog nastavka *.traineddata. Postoji posebna datoteka za svaki jezik, tako da je u ovom radu

korištena datoteka hrv.traineddata za hrvatski jezik. Datoteka *.traineddata za svaki jezik je arhivska datoteka u Tesseract specifičnom formatu. Sadrži nekoliko ne komprimiranih komponenti datoteka koje su potrebne Tesseract OCR procesu. Program combine_tessdata koristi se za stvaranje tessdata datoteke iz komponentskih datoteka. U slijedećem primjeru prikazano je izdvajanje tessdata datoteke na temelju hrvatskog jezika iz komponentske datoteke. Nazivi datoteka moraju imati odgovarajući sufiks koji ukazuje na njihov tip podataka. Kao rezultat toga, tessdata sadržavat će novi konfiguracijski jezik, odnosno hrvatski jezik.

```
combine_tessdata -u test/tessdata/ hrv.traineddata hrv.
Extracting tessdata components from test/tessdata/ hrv.traineddata
Wrote hrv.unicharset
Wrote hrv.unicharambigs
Wrote hrv.inttemp
Wrote hrv.pffmtable
Wrote hrv.normproto
Wrote hrv.punc-dawg
Wrote hrv.word-dawg
Wrote hrv.number-dawg
Wrote hrv.freq-dawg
Wrote hrv.cube-unicharset
Wrote hrv.cube-word-dawg
Wrote hrv.shapetable
Wrote hrv.bigram-dawg
Wrote hrv.lstm
Wrote hrv.lstm-punc-dawg
Wrote hrv.lstm-word-dawg
Wrote hrv.lstm-number-dawg
```

Postoje neki prijedlozi za zamjenu formata arhive Tesseract standardnim arhivskim formatom koji također može podržati kompresiju. Rasprava na talijanskom razvojnom forumu predložila je ZIP format već u 2014. godini. Godine 2017. provedena je eksperimentalna implementacija kao pull request.

4.3.Kod za detekciju teksta

Tesseract ima podršku za unicode (UTF-8) i može prepoznati više od 100 jezika bez problema. Tesseract podržava različite formate izvođenja: običan tekst, hocr (html), tsv, invisible-text-only ,pdf.

Kako bi se dobio bolji OCR rezultat, u nekim slučajevima je potrebno poboljšati kvalitetu slike koju dajete Tesseract-u. Ovaj kod za detekciju teksta ne uključuje GUI aplikaciju, te je aplikacija izrađena u okviru ovog diplomskog rada.

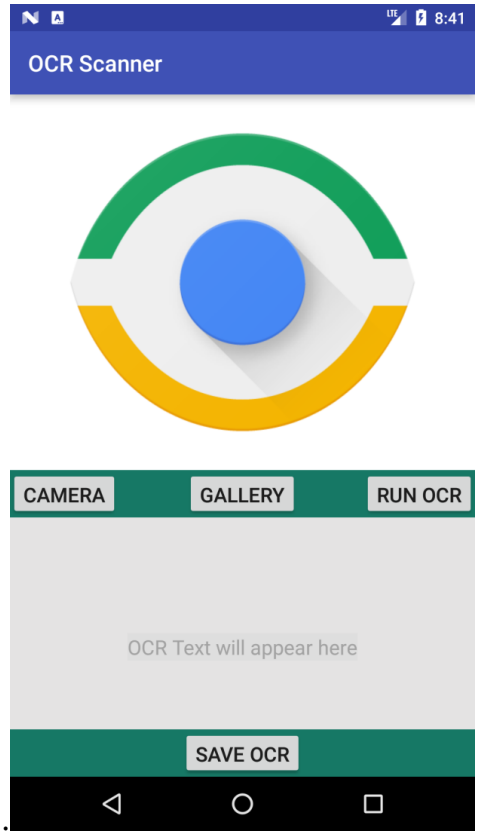
Tesseract može biti osposobljen za prepoznavanje drugih jezika.

4.4. Korištenje aplikacije

Aplikacija sama po sebi je jako lagana za korištenje i veoma je intuitivna. Aplikacija nudi mogućnost slikanja koristeći kameru mobilnog uređaja i mogućnost korištenja već postojeće slike na mobilnom uređaju. Odabirom slike na jedan od ta dva načina pokreće se proces prepoznavanja teksta.

Prilikom pokretanja aplikacije prikazat će se activity. U activity-u imamo naslovnu traku te ImageView prostor u kojem se prikazuje slika koju odaberemo za procesiranje. Ispod prostora za sliku nalazi se set od tri tipke. Prva tipka ili tipka pod nazviom “Camera“ pokreće kameru pametnog telefona i pokreće kod slikanja teksta za procesiranje. Druga tipka ili tipka pod nazivom “Gallery“ vodi u izbornik u kojem su prikazanje sve slike iz galerije od prije uslikane s uređajem, te dopušta birati samo jednu sliku. Treća tipka ili tipka pod nazivom “Run OCR“ pokreće kod za pretvaranje slike u tekst ,te stavlja pretvoreni tekst u EditText gdje se dobiveni tekst može editirati. Kada se tekst uredi po želji zadnja tipka na dnu ili tipka pod nazivom “Save OCR“ sprema tekst na memoriju uređaja u *.txt formatu.

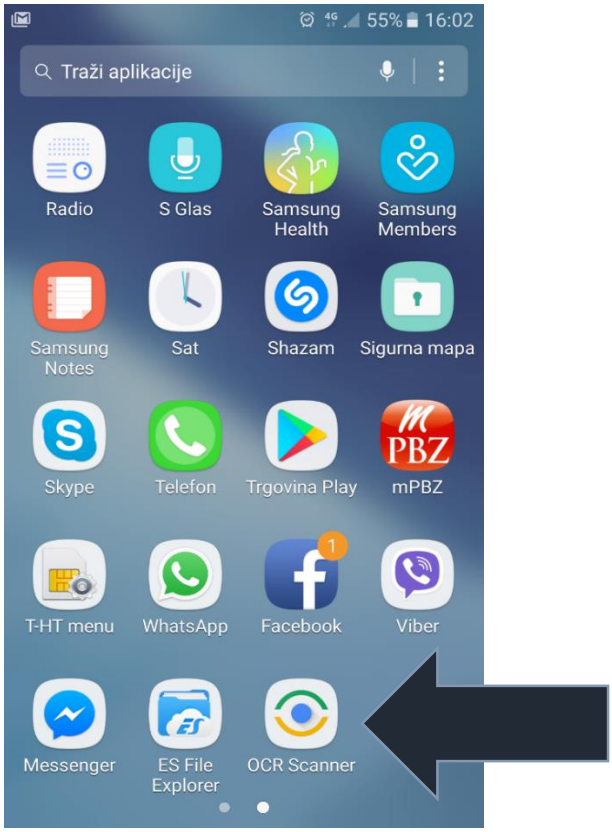
Na slici 4.2. prikazano je sučelje aplikacije OCR Scanner aplikacije za pretvorbu teksta sa slike u tekstualnu datoteku.



Slika 4.2. Izgled sučelja aplikacije

5. TESTIRANJE APLIKACIJE

U ovom poglavlju će se detaljno opisati testiranje aplikacije te će se slikama prikazati sve njezine mogućnosti kroz nekoliko testova. Nakon uspješne instalacije, aplikacija je spremna za korištenje. Kao i sve ostale Android aplikacije, pokreće se odabirom njene ikone u glavnom izborniku.



Slika 5. Prikaz aplikacije u glavnom izborniku

5.1. Uspješno prepoznat tekst sa slike s malom greškom

Na slici 5.1 prikazano je uspješno procesiranje teksta postojeće slike na mobilnom uređaju. Za uspješnost realizacije zaslužna je kvalitetna učitane slike visoke razlučivosti i čitljivosti samoga fonta teksta. Uspješnom procesiranju svakako doprinosi bijela podloga i crno otisnuti tekst. Jedina pogreška u pretvaranju teksta je u riječi DEKORACIJA gdje je OCR

Scanner spojio slova I i J u slovo U i ispisao riječ DEKORACUA . Pogreške ovog tipa su česte i zanemarive. Upravo zbog ovakvih pogrešaka u aplikaciji postoji mogućnost ispravka.

Slika 5.1 . Uspješno prepoznat tekst sa slike s malom greškom

5.2. Potpuno uspješno prepoznavanje teksta

Aplikacija uspješno procesuira tekst slike koja je uslikana pomoću kamere mobilnog uređaja kao što se može vidjeti na slici 5.2. Slika sadrži iste kvalitete kao i slika korištena u prvom testu, samim time je i procesuiranje rezultiralo uspjehom. Dakle, odabir lokacije slike, odnosno uporaba slike iz galerije mobilnog uređaja ili izravno uslikana slika upotrebom kamere mobilnog uređaja ne utječe na uspješnost pretvaranja teksta sa slike u tekstualnu datoteku.

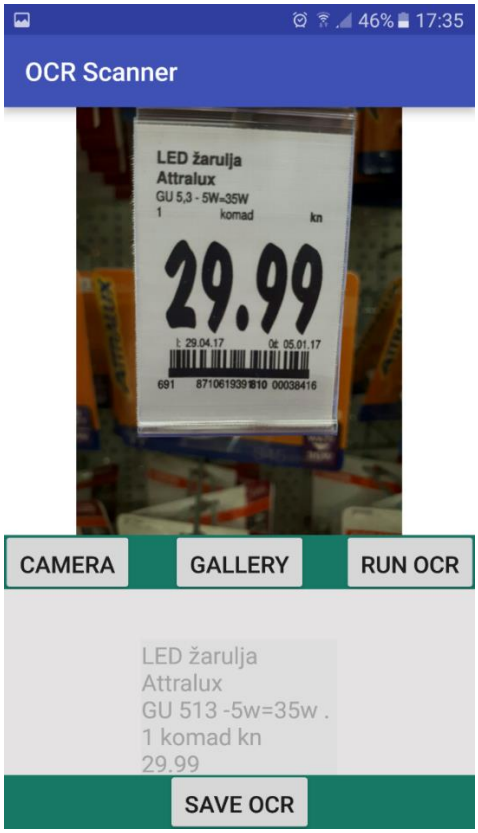
Slika 5.2. Potpuno uspješno prepoznavanje teksta

5.3. Usporedba prepoznavanja iste slike različite kvalitete

Testirane su dvije slike istog sadržaja ali različite kvalitete. Lijeva , odnosno slika 53. a) rezultat je procesuiranja slike loše kvalitete na kojoj tekst nije čitljiv, dok je na slici 5.3.b) prikaz rezultata čitanja teksta sa slike visoke kvalitete s jasno čitljivim znakovima. Iz ovoga se jasno vidi da je za uspješnost procesuiranja najvažnija kvaliteta slike. Dakle, ako se za prepoznavanje teksta koristi slika visoke kvalitete i sam rezultat prepoznavanja bit će dobar. No, ako u startu koristimo sliku koja je loše kvalitete za očekivati je da će i rezultat prepoznavanja biti lošs.

Za slike koje su loše kvalitete i koje se nalaze u galeriji mobilnog uređaja prije samog korištenja aplikacije preporuča se dodatna obrada slike da bi se postigla najbolja moguća kvaliteta kako bi i rezultat nakon korištenja aplikacije bio koliko toliko zadovoljavajući.

Slika 5.3. a) Zamućena slika

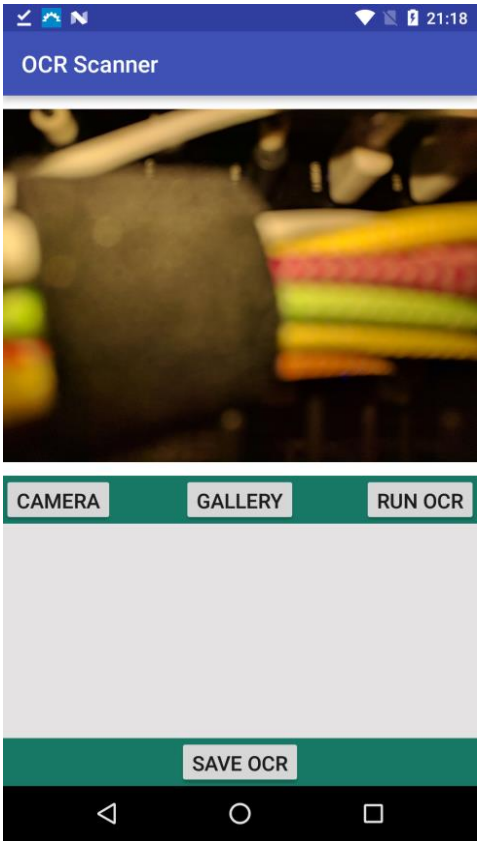


Slika 5.3. b) Slika bez zamućenja

5.4. Prepoznavanje slike bez teksta

Na slici 5.4 prikazano je testiranje slike kamere koja ne sadrži nikakav tekst. Budući da je ova aplikacija namijenjena za prepoznavanje teksta sa slike i pretvaranje tog prepoznatog teksta u tekstualnu datoteku, rezultat procesuiranja je neuspješan jer nema teksta za pretvorbu.

Da bi aplikacija funkcionirala tekst na slikama je nužan.

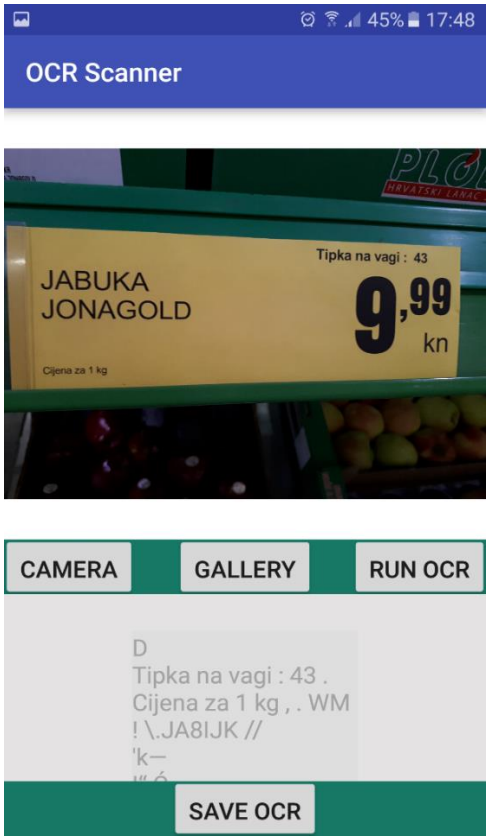


Slika 5.4. Prepoznavanje slike bez teksta

5.5. Prepoznavanje teksta sa slike koja nema podlogu u bijeloj boji

Iz slike 5.5. vidljivo je da boja podloge slike utječe na rezultat prepoznavanja teksta sa slike. Prilikom prepoznavanja došlo je do zamjene određenih slova, odnosno znakova. Tako je slovo B zamijenjeno brojem 8, a slovo U slovima I i J. Ove zamijene su nekako i logične jer su znakovi slični. Postoje još neke zamjene koje su česte prilikom prepoznavanja; neke od zamjena su : npr. veliko slovo O i broj 0,veliko slovo I i malo slovo l, malo slovo l i broj 1.

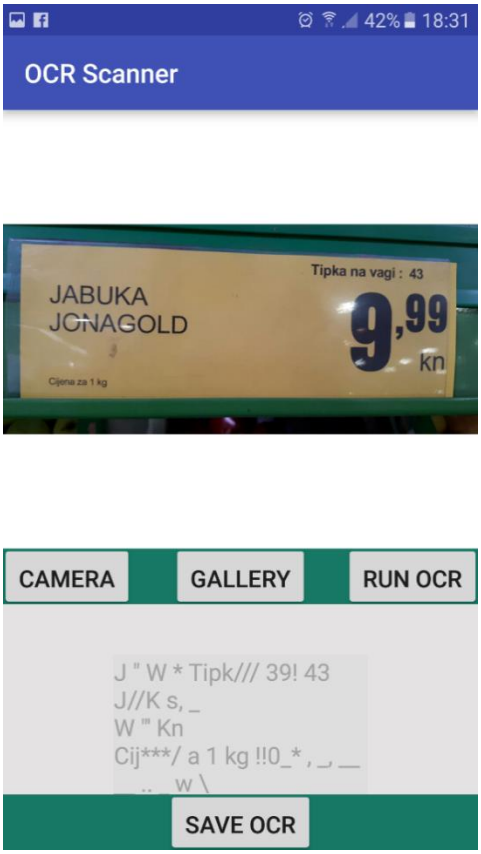
Iz slike 5.5. vidljivo je da neki znakovi uopće nisu procesuirani, odnosno prevedeni.



Slika 5.5. Prepoznavanje teksta sa slike koja nema podlogu u bijeloj boji

5.6. Prepoznavanje teksta sa plastificirane slike koja nema podlogu u bijeloj boji

Rezultat prepoznavanja teksta s plastificirane slike koja nema podlogu u bijeloj boji je dosta loš kao što je prikazano na slici 5.6. Uz probleme boje pozadine slike, prijevod teksta dodatno otežava i plastificirana površina koja prilikom slikanja kamerom mobilnog uređaja baca određeni odsjaj. Iz slike je vidljivo da većina znakova nije prepoznata, odnosno rezultat prepoznavanja ne odgovara tekstu učitane slike.

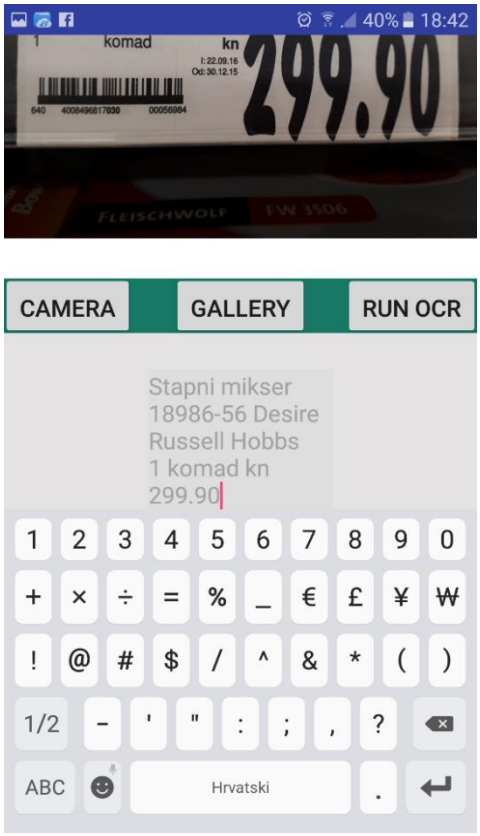


Slika 5.6. Prepoznavanje teksta sa plastificirane slike koja nema podlogu u bijeloj boji

5.7.Editiranje teksta

Kako je već ranije napisano aplikacija nudi mogućnost editiranja, tj. popravak prevedenog teksta. Primjer editiranja može se vidjeti na slici 5.7 .Rezultat procesuiranja bez editiranja prikazan je na slici 5.7. a), a rezultat nakon editiranja prikazan je na slici 5.7. b). Editiranje omogućuje pisanje i brisanje znakova prijevoda teksta.

Slika 5.7. a) Slika bez editiranja



Slika 5.7. b) Slika nakon editiranja

5.8. Spremanje teksta

Nakon što se završi procesuiranje i tekst po potrebi uredi, pritiskom zadnje tipke na dnu ili tipku pod nazivom “Save OCR“ tekst se sprema na memoriju uređaja u *.txt formatu. Na slici 5.8. prikazana je txt. datoteka u koju su spremljena sva ranija testiranja aplikacije. Iz slike se može uočiti da se prilikom spremanja rezultati pojedinog testa odvajaju crticama (----) kako bi se omogućilo lakše snalaženje u datoteci.

Slika 5.8. txt. datoteka aplikacije OCRScanner

6. ZAKLJUČAK

U ovom radu opisan je način dizajniranja, programiranje i korištenje mobilne aplikacije za pretvorbu teksta sa slike u tekst. Cilj je bio napraviti najjednostavniji sustav za pretvaranje teksta sa slike u tekst i demonstrirati jednostavnost korištenja istoga.

Sustav koristi sve besplatne i najmodernije tehnologije i alate kao što su: Android Studio, Java i Tesseract OCR.

Na kraju je predstavljen dizajn mobilne aplikacije i testiranje aplikacije.

Za izradu ovakvog sustava koji bi bio više komercijalan, preporučeno je da više ljudi radi na tome, od osobe koja će smisliti cijeli koncept mobilne aplikacije, preko dizajnera i programera pa sve do marketinških stručnjaka koji će predstaviti takav proizvod na tržištu.

LITERATURA

[1] Android (https://en.wikipedia.org/wiki/Android_(operating_system)), pristup ostvaren 27.4.2017.

[2] Arhitektura (http://blog.dnevnik.hr/print/id/1631938909/arhitektura-androida.html) , pristup ostvaren 29.4.2017.

[3] Arhitektura (https://www.google.hr/search?hl=hr&site=imghp&tbm=isch&source=hp&biw=1366&bih=613&q=arhitektura+operacijskog+sustava+android&oq=arhitektura+operacijskog+sustava+android&gs_l=img.3...3818.16208.0.17115.43.43.0.0.0.0.188.3192.38j2.40.0....0...1.1.64.img..3.14.1272.0..0j35i39k1j0i5i30k1j0i8i30k1j0i24k1._B9CSntAQdc#imgrc=rArkEwE5aFQgdM:) , pristup ostvaren 2.5.2017.

[4] Arhitektura (https://hr.wikipedia.org/wiki/Android_(operacijski_sustav)), pristup ostvaren 5.5.2017.

[5] Verzije (https://hr.wikipedia.org/wiki/Dosada%C5%A1nje_ina%C4%8Dice_sustava_Androida), pristup ostvaren 7.5.2017.

[6] Verzije (http://www.racunalo.com/marshmallow-je-najpopularnija-verzija-androida-nougat-mu-za-petama/), pristup ostvaren 10.6.2017.

[7] Verzije (https://www.viper.rs/servis-telefona/azuriranje-softvera) , pristup ostvaren 10.6.2017.

[8] Android Studio (https://en.wikipedia.org/wiki/Android_Studio) , pristup ostvaren 10.6.2017.

[9] Android Studio (https://developer.android.com/studio/features.html), pristup ostvaren 11.6.2017.

[10] Java (https://en.wikipedia.org/wiki/Java_(programming_language)), pristup ostvaren 12.6.2017.

[11] XML (https://en.wikipedia.org/wiki/XML), pristup ostvaren 14.6.2017.

[12] XML (http://www.fer.unizg.hr/_download/repository/mipro_xml_tekst.pdf), pristup ostvaren 16.6.2017.

[13] Tesseract OCR (https://en.wikipedia.org/wiki/Tesseract_(software)), pristup ostvaren 16.6.2017.

[14] Arhitektura Tesseract-a

(http://www.assistivetechnology.vcu.edu/wp-content/uploads/sites/1864/2013/09/pxc3882784.pdf) , pristup ostvaren 17.6.2017.

[15] Tesseract OCR : Cheriet, M., Kharma, N., Liu C., Suen, C.: Character Recognition Systems: A Guide for Students and Practitioners, Wiley, 2007

SAŽETAK

Mobilna aplikacija za pretvorbu teksta sa slike u tekst

U radu je opisano razvojno okruženje za razvoj Android aplikacija, Android Studio, te osnovne smjernice za izradu aplikacije. Predstavljene su osnovne informacije o Android operacijskom sustavu i o programskom jeziku Java. Tesseract OCR je besplatni softver za prepoznavanje znakova na slikama, čiji je razvoj i arhitektura predstavljen u ovom radu. . Također je izrađena aplikacija za pretvaranje teksta sa slike u tekst u razvojnom okruženju Android Studio.

Glavni zadatak diplomskog rada je pretvoriti tekst sa slike u tekstualnu datoteku uporabom mobilne aplikacije. Prvi korak je učitati sliku s tekstom iz već postojeće galerije uređaja ili snimiti sliku pomoću kamere mobilnog telefona i odmah ju učitati. Pretvaranje teksta sa slike je ostvareno preko Tesseract OCR alata. Nakon procesuiranja teksta sa slike, aplikacija nudi mogućnost editiranja i spremanja rezultata obrade u obliku txt datoteke na memoriju mobilnog uređaja.

Predstavljeni su i detaljno opisani rezultati testiranja aplikacije.

Ključne riječi: mobilna aplikacija, Android, Android Studio,Java, Tesseract

ABSTRACT

Mobile application for transformation of text in an image to text

This project describes the development environment for Android application development, Android Studio, and basic application development guidelines. Basic information about the Android operating system and the Java programming language has been introduced. Tesseract OCR is a free character recognition software on images, the development and architecture of which is presented in this project. . There is also an application to convert text from image to text in the Android Studio development environment.

The main task of the thesis is to convert text from image to text file using mobile application. The first step is to load an image with text from an existing gallery of your device or capture an image using your mobile phone's camera and load it right away. Converting text from the image is achieved through the Tesseract OCR tool. After processing the text from the image, the application offers the ability to edit and save processing results in .txt format. files on your mobile device memory.

The test results of the application are described in detail.

Keywords: mobile app, Android, Android Studio, Java, Tesseract

ŽIVOTOPIS

Kristina Andrišek rođena je 20. srpnja 1991. godine u Požegi. Osnovno obrazovanje stječe u Osnovnoj školi Zdenka Turkovića u Kutjevu. Po završetku osnovne škole upisuje Prirodoslovno - matematičku Gimnaziju u Požegi. Srednjoškolsko obrazovanje završava 2011. godine nakon čega upisuje preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku. Na istom fakultetu od 2014. pohađa Sveučilišni diplomski studij, smjer Procesno Računarstvo.

PRILOZI

Prilog 1. Direktorij KOD/

Sadrži kod aplikacije.