Unaprjeđenje algoritma za prepoznavanje tekstualnih znakova

Brisinello, Matteo

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

Sveučilišni studij

UNAPRJEĐENJE ALGORITMA ZA PREPOZNAVANJE TEKSTUALNIH ZNAKOVA

Diplomski rad

Matteo Brisinello

Osijek, 2017.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 22.09.2017.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Matteo Brisinello
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 796 R, 09.10.2015.
OIB studenta:	38025588610
Mentor:	Doc.dr.sc. Ratko Grbić
Sumentor:	
Sumentor iz tvrtke:	Matija Pul
Predsjednik Povjerenstva:	Doc.dr.sc. Mario Vranješ
Član Povjerenstva:	Izv. prof. dr. sc. Marijan Herceg
Naslov diplomskog rada:	Unaprjeđenje algoritma za prepoznavanje tekstualnih znakova
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	Jedan od čestih zadataka u testiranju BBT (engl. Black Box Testing) metodologijom je prepoznavanje tekstualnih sadržaja u statičkim slikama. Izazovi koji se javljaju u ovom području vezani su za promjenljivu pozadinu iza tekstualnih sadržaja (gotovo uvijek se tekst nalazi u sloju iznad "živog" video sadržaja), prisustvo šuma u slici (kod analognih signala), nedovoljna veličina teksta i mnogi drugi. Zadatak ovog rada je unaprjeđenje pouzdanosti rada OCR (engl. Optical Character Recognition) algoritma u situacijama kada postoji referentna slika s traženim tekstom, pri čemu bi se do konačnog rezultata došlo kombiniranjem OCR algoritma s algoritmom za usporedbu slika. (sumentor Matija Pul, Institut RT-RK Osijek, Osijek)
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	22.09.2017.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 28.09.2017.

Ime i prezime studenta:	Matteo Brisinello
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D 796 R, 09.10.2015.
Ephorus podudaranje [%]:	1

Ovom izjavom izjavljujem da je rad pod nazivom**: Unaprjeđenje algoritma za prepoznavanje tekstualnih znakova**

izrađen pod vodstvom mentora Doc.dr.sc. Ratko Grbić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

Sadržaj

1. UVOD

Prepoznavanje tekstualnih znakova (engl. *Optical Character Recognition*) može se definirati kao postupak čitanja teksta sa digitalne slike pomoću računalnog programa. Postoje razni računalni programi za prepoznavanje tekstualnih znakova. Neki od najpoznatijih su *Tesseract*, *CuneiForm* i *ABBYY*. Najčešća primjena programa za prepoznavanje tekstualnih znakova je čitanje teksta sa skeniranih slika. Oni se također primjenjuju za čitanje teksta s osobnih dokumenata kao što su osobna iskaznica i putovnica. Relativno nedavno, njihova primjena se proširila na čitanje teksta sa registracijskih oznaka te u automobilskoj tehnologiji, za čitanje teksta s prometnih znakova. Međutim, prepoznavanje tekstualnih znakova može se koristiti i u druge razne svrhe. Jedna od njih je čitanje teksta sa slika koje su dohvaćene sa *set-top box* (STB) uređaja u svrhu testiranja istih. U navedenom scenariju, postoje određeni problemi koje je potrebno uzeti u obzir prije samog pokretanja programa za prepoznavanje tekstualnih znakova kako bi se postigao visok stupanj točnosti samog prepoznavanja. Slike dobivene sa STB uređaja su često niže rezolucije od one propisane od strane korištenog programa za prepoznavanje teksta. Loša kvaliteta dobivenih slika također može utjecati na točnost rezultata prepoznavanja teksta. Pod pojmom loše kvalitete u ovom kontekstu, podrazumijevaju se situacije u kojima su dva ili više tekstualna znaka spojena ili situacije u kojima postoje rupe u znakovima gdje ne bi trebale postojati. Moderniji STB uređaji pružaju mogućnost korištenja sučelja koji su transparentni, pa je samim time pozadina iza teksta šarena i promjenjiva što također utječe na točnost prepoznavanja tekstualnih znakova.

U ovom radu *Tesseract* je korišten za prepoznavanje tekstualnih znakova na slikama dohvaćenih s različitih STB uređaja [1]. Pokretanje *Tesseract*-a bez ikakve predobrade slike u navedenim okolnostima često rezultira lošim rezultatima prepoznavanja teksta upravo zbog neprikladne rezolucije slike, loše kvalitete i šarene pozadine. Što se tiče predobrade slike, *Tesseract* koristi metodu prilagodljivog praga u svrhu binarizacije slike. Nekoliko tehnika za poboljšanje točnosti prepoznavanja teksta je preporučeno u tehničkoj dokumentaciji *Tesseract*-a pod poglavljem *ImproveQuality* [2]. Međutim, ni jedna tehnika se ne odnosi na spomenuti problem šarene pozadine.

U ovom radu su predložene četiri različite metode za poboljšanje točnosti *Tesseract*-a na slikama dohvaćenih sa STB uređaja. Predložene metode primijenjene su prije samog prepoznavanja tekstualnih znakova pomoću *Tesseract*-a. Prva metoda povećava rezoluciju analiziranih slika. Druga i treća metoda primjenjuju filter izoštravanja, odnosno filter zamućivanja. Posljednja metoda je metoda grupiranja (engl. *clustering*) čiji je cilj razdvajanje teksta od šarene pozadine. U radu je također ispitana mogućnost automatizacije procesa prepoznavanja teksta kada su u pitanju

slike sa šarenom pozadinom. Točnost prepoznavanja tekstualnih znakova provjerena je na slikama dohvaćenih s različitih STB uređaja. Dobivene slike podijeljene su u šest grupa prema njihovim karakteristikama kao što su rezolucija, tip pozadine itd. Rezultati prepoznavanja tekstualnih znakova dobiveni su koristeći *Tesseract 3.5* i relativnog novog *Tesseract 4.0* na originalnim slikama kao i na slikama koje su najprije obrađene spomenutim metodama.

Rad je podijeljen u pet poglavlja. U drugom poglavlju opisane su postojeće metode koje se bave sličnim problemom poboljšanja algoritma za prepoznavanje tekstualnih znakova. U trećem poglavlju detaljnije su opisane četiri različite metode za poboljšanje točnosti prepoznavanja tekstualnih znakova. Četvrto poglavlje prezentira dostupne testne slike i prikazuje dobivene rezultate koristeći *Tesseract 3.5* i *Tesseract 4.0* na originalnim slikama kao i rezultate dobivene nakon primjene četiri različitih metoda predobrade slike. Na kraju rada dan je zaključak.

2. PREGLED STANJA

Prepoznavanje tekstualnih znakova je grana prepoznavanja uzoraka, umjetne inteligencije i računalnog vida koja se konstantno poboljšava. Ne postoji program za prepoznavanje tekstualnih znakova čija je točnost 100% u svim mogućim primjenama. Međutim, postoje razni pristupi i rješenja koja mogu unaprijediti točnost algoritma u određenim primjenama. U nastavku su istaknuti najčešći problemi pri korištenju algoritma za prepoznavanje tekstualnih znakova te su predstavljena postojeća rješenja za slične primjene.

2.1. Problemi pri korištenju algoritma za prepoznavanje tekstualnih znakova

Postoje mnoge primjene algoritma za prepoznavanje tekstualnih znakova, pa samo poboljšanje algoritma ovisi o samoj primjeni. Neki programi za prepoznavanje tekstualnih znakova su otvorenog kôda, pa jedna od mogućnosti unaprjeđenja točnosti samog algoritma je modifikacija samog kôda. Navedene programe obično godinama razvijaju iskusni inženjeri, pa je navedena mogućnost unaprjeđenja gotovo nemoguća za izvesti. Najčešći način poboljšanja točnosti je predobrada slike prije samog pokretanja algoritma za prepoznavanje tekstualnih znakova. Ovim pristupom važno je poznavati karakteristike slika na kojima će prepoznavanje biti izvršeno. Ako je poznato da se algoritam primjenjuje na crno-bijelim slikama, fokus poboljšanja algoritma bit će isključivo na samoj kvaliteti slike. Ako se algoritam primjenjuje na slikama sa šarenom pozadinom, fokus će biti na izdvajanju teksta, detekciji teksta ili nekom sličnom pristupu. Bitne stavke su i pozicija teksta na slikama te veličina teksta. Ako nije poznato gdje se tekst točno nalazi, potrebno je primijeniti neki algoritam za detekciju teksta. S obzirom da postoje razne primjene algoritma za prepoznavanje tekstualnih znakova, samo unaprjeđenje točnosti donosi probleme specifične za pojedinu primjenu. U nekim primjenama, korištene slike su premale ili prevelike rezolucije u odnosu na one propisane od korištenog programa. Postoje slučajevi gdje slike sadrže previše detalja, odnosno relativno puno informacija visokih frekvencija koje mogu bitno utjecati na rezultate prepoznavanja teksta. Na takvim slikama vrlo često budu prepoznati dodatni dijakritički znakovi koji nisu dio teksta kao točke, zarezi, kvačice, kružići, apostrofi itd. Postoje i slučajevi gdje je boja teksta vrlo slična boji pozadine. U takvim slučajevima postoji šansa da tekst bude samo djelomično prepoznat, odnosno da neka slova budu smatrana kao dio pozadine te izostavljena u konačnom rezultatu. Možda najveći problem i izazov pri korištenju te unaprjeđenju algoritma za prepoznavanje tekstualnih znakova predstavljaju slike koje sadrže šarenu pozadinu. Navede su slike slučaj za sebe te postoje razni problemi koji mogu utjecati na točnost

prepoznavanja teksta. U nekim primjenama potrebno je pročitati tekst sa slike gdje pozicija teksta nije poznata, kao ni njegova veličina ili boja.

2.2. Postojeća rješenja

Većina postojećih rješenja poboljšanja algoritma za prepoznavanje tekstualnih znakova bavi se eliminiranjem šarene pozadine i ostalim metodama predobrade koje se primjenjuju prije pokretanja algoritma za prepoznavanje tekstualnih znakova. U [3] je predložena metoda koja uklanja pozadinu od teksta. Metoda koristi dvije pretpostavke. Slike u pozadini su bogatije što se tiče teksture u usporedbi s tekstom te šarene slike u pozadini imaju veću međusobnu razliku *RGB* vrijednosti svakog elementa slike (engl. *pixel*). Na temelju ovih dviju pretpostavki, prvi korak metode je poboljšanje kontrasta slike. Nakon toga slika se pretvara u sliku u sivim tonovima te se pretvara u binarnu sliku. Ova metoda se bavi slikama skeniranih dokumenata u kojima iza teksta postoji neka slika. Algoritam identifikacije teksta u slikama izvučenih iz video zapisa je predstavljen u [4]. Algoritam koristi metodu potpornih vektora (engl. *Support vector machine*). Sastoji se od dva koraka: pronalaženje linija teksta te identifikacija teksta korištenjem metode potpornih vektora. Algoritam je izvršen na slikama izvučenih iz video zapisa koji prikazuju reklame, sport, intervjue, vijesti, filmove te na slikama iz novina, geografskih karata te letaka. U [5] je predstavljena metoda koja je zasnovana na pretpostavki da se element slike koji je dio teksta uvijek nalazi između jednog „para rubova“. U [6] je predstavljen hibridni pristup koji kombinira analizu povezanih komponenata te pretvaranje slike u binarnu sliku korištenjem praga u svrhu izdvajanja teksta od kompleksne pozadine. Loša strana ove metoda je ta da je primjenjiva samo na slikama s relativno jednostavnom pozadinom. Kompliciranije pozadine stvarale bi problem pri detekciji rubova.

Kao što je prethodno navedeno postoje primjene algoritma za prepoznavanje tekstualnih znakova na slikama sa šarenom pozadinom gdje pozicije teksta nisu poznate. U takvim slučajevima najbolje rješenje je koristiti neki od algoritama za detekciju teksta koji bi kao rezultat dali poziciju teksta na šarenoj slici. Nakon detekcije pozicije teksta moguće je izrezati dijelove s tekstom te dalje ih obrađivati ili odmah pokrenuti algoritam za prepoznavanje tekstualnih znakova. Jedan od načina detekcije teksta je korištenje algoritma maksimalno stabilnih ekstremnih regija (engl. *Maximally stable extremal regions - MSER*). Prema [7] navedeni algoritam je općenito korišten za prepoznavanje objekata koji je baziran na principu detekcije grumenčića (engl. *blob*). U kombinaciji s algoritmom za detekciju rubova, spomenuti algoritam može detektirati regije teksta na slikama sa šarenom pozadinom. Nedostatak kod korištenja ove metode je taj da je ona vremenski zahtjevna te nije prigodna u primjenama kada je vremenska učinkovitost bitna.

3. POBOLJŠANJE TOČNOSTI

Kao što je spomenuto u uvodu, slike korištene u ovom radu dobivene su sa raznih STB uređaja. Tekst na tim slikama može se nalaziti bilo gdje i može biti bilo koje veličine i bilo kojeg fonta. Pri testiranju STB uređaja lokacije teksta na dobivenim slikama su unaprijed poznate, pa je moguće izrezati regije slike na kojima se nalazi tekst te proslijediti ih *Tesseract*-u. Takve slike obično sadrže jednu liniju teksta s određenom bojom i fontom. Nakon izvršavanja algoritma prepoznavanja tekstualnih znakova na navedenim slikama, uspoređuje se prepoznati tekst s očekivanim tekstom koji je unaprijed poznat. Ukoliko je prepoznati tekst jednak očekivanom, smatra se da je uređaj prošao test. Testiranje STB uređaja uključuje veliki broj testova koji se izvode redom, jedan za drugim, pa je prilikom testiranja istih bitna i brzina izvođenja samih testova. Stoga, bitno je da metode predobrade slike koje se koriste tijekom testiranja ne budu računalno zahtjevne, te moraju biti vremenski učinkovite.

3.1. Poboljšanje točnosti na slikama niske rezolucije

Tesseract preporučuje minimalnu veličinu potrebnu za točno prepoznavanje teksta. Točnost opada ako je *X*-visina manja od 20 elemenata slike. *X*-visina definirana je kao visina malog slova x. Stoga, jednostavna metoda predobrade slike koja uvelike može poboljšati točnost *Tesseract*-a je povećanje slike tako da je *X*-visina veća od minimalne preporučene. Prva metoda koja je primijenjena u ovom radu za poboljšanje točnosti na slikama niske rezolucije je povećanje slike tako da njena visina bude 100 elemenata slike. Povećanje slike primijenjeno je samo na slikama čija je visina manja od 100 elemenata slike. Iako *Tesseract* preporučuje minimalnu visinu od 20 elemenata slike, odabrana je minimalna visina od 100 elemenata slike zato što preporuka *Tesseract*-a se uglavnom odnosi na skenirane crno-bijele slike. Uzevši u obzir slike loše kvalitete kojima se ovaj rad bavi, odabrana je veća veličina minimalne visine slike. Metoda povećanja slike korištena u ovom radu je bikubična interpolacija. Za razliku od ostalih metoda povećanja slike, bikubična interpolacija osigurava očuvanje detalja [8]. Pošto se druga i treća metoda u ovom radu bave detaljima na slikama, od velike je važnosti očuvanje detalja pri povećanju slike.

3.2. Poboljšanje točnosti na slikama loše kvalitete

Druga metoda predobrade slike je izoštravanje slike. Glavni razlog primjenjivanja ove metode je povećanje kontrasta između rubova, odnosno povećanje kontrasta između teksta i pozadine. U ovom radu, korištena metoda izoštravanja slike je metoda zamućenog maskiranja (engl. *unsharp*

masking). Prema izrazu (3-1) zamućeno maskiranje je postupak u kojem se oduzima zamućena slika $f_{zamućena}$ od originalne slike *f*, čiji je rezultat slika *g* koja sadrži samo rubove sa slike.

$$g(i,j) = f(i,j) - f_{zamućena}(i,j) \tag{3-1}$$

Postoji više načina zamućivanja slike. Način koji je korišten u ovom radu je Gaussov niskopropusni filtar. Prema izrazu (3-2) slika *g* koja sadrži samo rubove slike zbraja se sa originalnom slikom *f* te se u konačnici dobiva izoštrena slika $f_{izoštrena}$ koja se dalje obrađuje *Tesseract*-om:

$$f_{izoštrena}(i,j) = f(i,j) + g(i,j) \tag{3-2}$$

Općenito, filtri zamućivanja implementirani su korištenjem težinskog zbroja vrijednosti elemenata slike u prostoru veličine *M x M* elemenata slike. Pri korištenju Gaussovog filtra, težinski koeficijenti odabiru se prema obliku Gaussove funkcije. Moguće je kontrolirati tri različita parametra kako bi se dobio željeni učinak izoštravanja. Prvi parametar je standardna devijacija Gaussovog niskopropusnog filtra koji određuje oblik same funkcije. Ovim parametrom moguće je kontrolirati veličinu regije na koju će samo izoštravanje djelovati. Drugi parametar određuje jačinu efekta izoštravanja. Veća vrijednost ovog parametra rezultira većim kontrastom između izoštrenih elemenata slike. Obično postoji i treći parametar koji predstavlja prag koji određuje koji je minimalni kontrast između susjednih elemenata slike koji su rubovi.

Treća metoda je metoda zamućivanja. Ova metoda smanjuje informacije visokih frekvencija koje mogu pogoršati točnost prepoznavanja tekstualnih znakova. Zamućivanje je ostvareno primjenom niskopropusnog filtra na originalnu sliku *f* tako da je svaki element slike zamijenjena prosječnom vrijednošću svih vrijednosti elemenata u susjedstvu veličine 9 x 9 elemenata:

$$h(i,j) = \frac{1}{81}\sum_{k=i-4}^{i+4}\sum_{l=j-4}^{j+4} f(k,l) \tag{3-3}$$

Primjenjivanjem ovog filtra dobije se zamućenija slika *h* s manje detalja u odnosu na originalnu sliku *f*. Na slikama kojim se bavi ovaj rad, primjena ovog filtra otklanja šum iz pozadine koji često može biti prepoznat kao dijakritički znak.

3.3. Poboljšanje točnosti na slikama sa šarenom pozadinom

Četvrta metoda predobrade slike namijenjena je obradi slika koje sadrže šarenu pozadinu. Ideja je da ovom metodom predobrade najprije odvoji tekst od pozadine, a zatim da se primjeni algoritam za prepoznavanje teksta. S obzirom da je slika izrezana i sadrži jednu liniju teksta, može se zaključiti da elementi slike koji su dio teksta zauzimaju značajan dio slike i one su otprilike iste boje. Uzimajući ovo u obzir, moguće je izdvojiti elemente slike koji su dio teksta metodom grupiranja (engl. *clustering*). U ovom radu za grupiranje je korišten algoritam *k* srednjih vrijednosti (engl. *k-means*). Nakon primjene ovog algoritma za pretpostaviti je da će jedna grupa sadržavati samo elemente slike koji su dio teksta. Upravo tu grupu, odnosno sliku je potrebno proslijediti *Tesseract*-u na prepoznavanje tekstualnih znakova.

3.3.1. Algoritam *k* srednjih vrijednosti

Prema [9], algoritam *k* srednjih vrijednosti jedan je od najjednostavnijih algoritama nenadziranog učenja koji rješava vrlo čest problem grupiranja. Algoritam se primjenjuje na skup podataka koje je potrebno podijeliti u grupe. Podaci mogu biti višedimenzionalni. Preduvjet pri korištenju ovog algoritma je poznavanje konačnog broja grupa *k* u koje će podaci biti podijeljeni. Postoji nekoliko algoritama za računanje idealnog broja grupa, no ti algoritmi su često vremenski zahtjevni. Kako sve metode preporučene u ovom radu teže tomu da budu jednostavne i vremenski učinkovite, niti jedan algoritam za računanje grupa nije korišten. Algoritam *k* srednjih vrijednosti radi tako da se prvo odaberu predstavnici grupa pri čemu je svaki predstavnik vektor čija dimenzija ovisi o dimenziji podataka koji se grupiraju. Naime, postoji odgovarajuća kriterijska funkcija koja se minimizira, a temelji se na euklidskoj udaljenosti. Algoritam je iterativan, najprije se zadaju početne vrijednosti predstavnika grupa čije vrijednosti se onda preračunavaju s ciljem minimizacije kriterijske funkcije. Svaka iteracija se odvija u dva koraka: prvo se svaki podatak dodijeli najbližem predstavniku, a zatim se preračunavaju vrijednosti predstavnika itd.

U ovom radu, algoritam *k* srednjih vrijednosti korišten je tako da su slike sa šarenom pozadinom prvo pretvorene iz *RGB* u *L*a*b* prostor boja. S obzirom da informacije o boji postoje samo u a*b* prostoru boja, algoritam *k* srednjih vrijednosti korišten je za razdvajanje elemenata slike u grupe koristeći njihove a* i b* vrijednosti. Na slici 3.1. prikazan je primjer grupiranja korištenjem algoritma *k* srednjih vrijednosti.

(a) (b)

(c) (d)

Sl. 3.1. Primjer grupiranja gdje je *k* = 3. (a) Originalna slika, (b) grupa koja sadrži tekst, (c) prva grupa koja sadrži pozadinu, (d) druga grupa koja sadrži pozadinu.

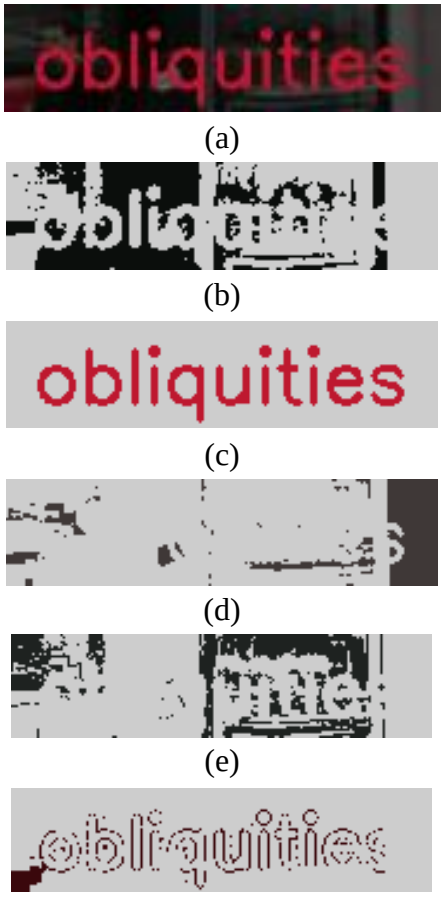
3.3.2. Prepoznavanje teksta nakon grupiranja

Kao rezultat izvršavanja algoritma grupiranja dobije se *k* slika od kojih jedna sadrži uglavnom tekst, a ostale slike sadrže dijelove šarene pozadine. Potrebno je pokrenuti algoritam za prepoznavanje tekstualnih znakova upravo na slici koja sadrži tekst. Izazov koji se javlja u ovoj situaciji je identifikacija upravo te slike u skupu *k* slika. Jedan od najjednostavnijih načina koji će osigurati prepoznavanje teksta na slici koji sadrži tekst je da korisnik prije pokretanja algoritma grupiranja zada boju teksta. Na ovaj način će se i zaobići potreba za identifikacijom slike koja sadrži tekst. Zadavanje boje teksta od strane korisnika u nekim situacijama nije pogodno i zahtjeva puno korisničkog vremena npr. Ako se algoritam pokreće na više stotina ili tisuća slika na kojima tekst nije uvijek iste boje. U ovom slučaju potrebno je uvesti jedan dodatan korak automatizacije koji će osigurati prepoznavanje teksta na slici koja sadrži tekst nakon grupiranja bez obzira na samu boju teksta. U ovom radu ispitana su dva pristupa koja rješavaju ovaj problem. Kao što je ranije spomenuto, pri testiranju STB uređaja, tekst kojeg je potrebno pročitati unaprijed je poznat. Na taj način moguće je pročitati tekst na svakoj slici nakon grupiranja te usporediti ga s onim koji se traži, no ovaj pristup u slučaju većeg broja grupa zna biti prilično neučinkovit. Drugi pristup koristi strojno učenje za klasifikaciju svih grupa nakon izvršavanja algoritma *k* srednjih vrijednosti. Kao rezultat, algoritam od *k* slika izdvaja onu na kojoj se nalazi tekst.

3.3.3. Uzorkovanje boja

Prvi način automatizacije procesa prepoznavanja teksta na slikama sa šarenom pozadinom koji je ispitan u ovom radu je uzorkovanje boja. Navedeni postupak smanjuje broj ukupnih boja na slici. Za ostvarivanje navedenog korišten je algoritam *k* srednjih vrijednosti na način opisan u poglavlju 3.3.1. Nakon primjene navedenog algoritma na originalnu sliku, dobije se *k* grupa gdje svaka grupa sadrži jednu od *k* dominantnih boja originalne slike. Iz navedenih grupa stvara se *k* slika na kojima će biti prikazane elementi slike pojedinih grupa. Vrijednosti elemenata slike odnosno njihove boje

su vrijednosti centara pojedinih grupa. Te vrijednosti su ujedno dominante boje sa originalne slike. Na taj način će dobivene slike biti binarne, odnosno sadržavat će samo dvije boje: boju pozadine i boju centra određene grupe. Primjer uzorkovanja boje prikazan je na slici 3.3. Nakon uzorkovanja boja na slici potrebno je pokrenuti prepoznavanje teksta, no nije poznato na kojoj slici se nalazi tekst. Samo prepoznavanje teksta je relativno brza operacija, pa je ono pokrenuto na svim grupama. U ovom radu, tekst kojeg je potrebno pročitati unaprijed je poznat, pa je moguće usporediti očekivani tekst i tekst koji je pročitan.



(a)

(b)

(c)

(d)

(e)

(f)

Sl. 3.2. Primjer uzorkovanja boja sa $k = 5$. (a) Originalna slika, (b-f) grupe.

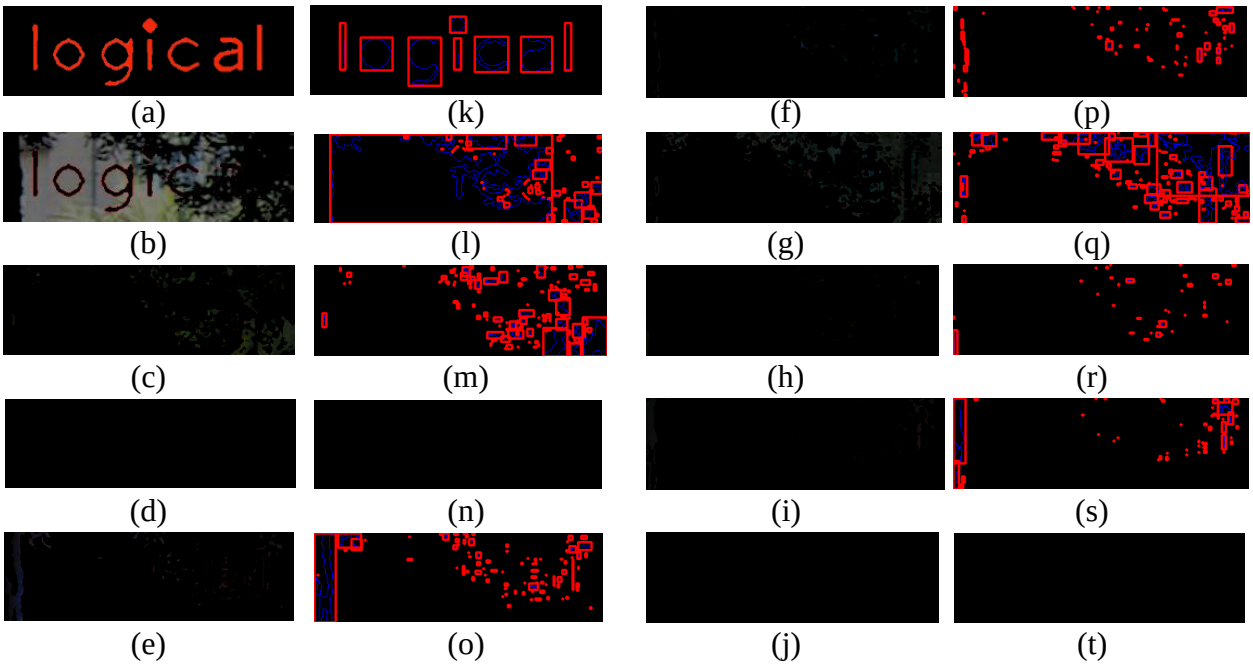
3.3.4. Identifikacija slike s tekstom

Jedan od načina automatizacije procesa prepoznavanja teksta na slikama sa šarenom pozadinom je identifikacija slike koja sadrži tekst u skupu k slika nakon izvršavanja algoritma grupiranja. U ovom radu navedeni proces je obavljen korištenjem strojnog učenja. Strojno učenje omogućava izradu matematičkog modela koji se može negdje implementirati. Postoje razne potrebe za korištenjem strojnog učenja, a jedan od čestih problema je problem klasifikacije. Kod problema

klasifikacije potrebno je ulaznim podacima dodijeliti jednu od mogućih klasa. Ako izlazni podaci mogu biti samo dvije vrijednosti, riječ je o binarnoj klasifikaciji. Prije samog korištenja modela potrebno ga je istrenirati, što podrazumijeva određivanje strukture matematičkog modela i određivanje parametara te strukture. Za učinkovit postupak učenja najčešće je potreban relativno veliki skup podataka koji je označen odnosno na raspolaganju su podatkovni primjeri (ulazno-izlazni parovi). Također, potrebno je odabrati i algoritam koji će se koristiti u procesu učenja, kao i kasnije u fazi eksploatacije odnosno u procesu dodjeljivanja klase novim ulaznim podacima. Spomenuti algoritam strojnog učenja često se naziva i klasifikator. Postoje razni klasifikatori, a neki od njih su neuronska mreža, šume odlučivanja, metoda najbližih susjeda, strojevi s potpornim vektorima, logistička regresija, Bayesov klasifikator itd. Izbor klasifikatora ovisi o puno parametara, a neki od njih su brzina treniranja, brzina izvođenja, točnost, broj značajki, složenost podataka itd. Nakon ispitivanja točnosti različitih algoritama za podatke korištene u ovom radu, došlo se do zaključka da je Bayesov klasifikator najpogodniji. Navedeni algoritam zasnovan je na Bayesovom teoremu iz područja teorije vjerojatnosti. Klasifikatoru je kao ulaz potrebno predati skup podataka koji se naziva vektor značajki, a on kao odgovor daje kojoj klasi pripada taj skup podataka.

U ovom radu analizira se problem binarne klasifikacije, tj. klasifikator na temelju ulazne slike treba odrediti nalazi li se na slici tekst ili pozadina. Međutim, umjesto cijele slike na ulaz klasifikatora se dovode odgovarajuće značajke (engl. *features*) slike. Odabir značajki ključan je dio izrade sustava za klasifikaciju. Značajke moraju predstavljati svoju klasu što je moguće bolje kako bi sustav radio sa zadovoljavajućom točnošću. Postoje određene pretpostavke koje su uzete u obzir pri odabiru značajki u ovom radu. Prva pretpostavka je da je tekst jednobojan. Navedena pretpostavka osigurava ispravan rad algoritma za grupiranje. Kao što je ranije spomenuto, algoritam grupiranja grupira elemente slike po boji, stoga elementi slike koji su dio teksta moraju biti iste ili približno iste boje kako bi ih algoritam *k* srednjih vrijednosti svrstao u istu grupu. Druga pretpostavka je ta da se na slici nalazi jedna linija teksta. Treća i zadnja pretpostavka je ta da se na slici nalazi jedna riječ, odnosno da nema praznih razmaka između slova. Prva pretpostavka osigurava da prije same klasifikacije postoji *k* slika od kojih jedna sadrži elemente slike koji su dio teksta, a ostale sadrže elemente slike koji su dio pozadine. Iz tog skupa potrebno je iz svake slike izvaditi značajke. Na temelju druge i treće pretpostavke odabrane su značajke koje će se koristiti za treniranje sustava kao i pri samom korištenju sustava. Prvi korak pri određivanju značajki na pojedinoj slici je pronalaženje kontura. Konture su spojeni elementi slike koji imaju istu boju ili intenzitet [10]. Kako su elementi slike na pojedinim slikama rezultat grupiranja po boji, svi spojeni elementi slike smatraju se konturama. Drugi korak je određivanje pravokutnika oko svake konture,

što omogućava bolji pregled pozicija pojedinih kontura, mogućnost računanja visine, širine, površine, koordinate krajnjih elemenata slike konture itd. Primjer određivanja kontura i pravokutnika oko kontura na slikama nakon grupiranja prikazan je na slici 3.2. Vidljivo je kako su pravokutnici na slikama s pozadinom relativno nepravilno pozicionirani, različitih veličina i oblika. Na temelju pretpostavke da se na slici nalazi jedna linija teksta, može se reći kako su pravokutnici približno isto pozicionirani po *y* osi slike. Kako bi se izračunala navedena sličnost pozicija pravokutnika, izračunata je standardna devijacija donjih *y* vrijednosti svih pravokutnika prema (3-4), gdje je x_i pojedina donja *y* koordinata pravokutnika, $\bar{x}$ srednja vrijednost svih donjih *y* koordinata te *N* ukupan broj pravokutnika.



Sl. 3.3. (a-j) Slike nakon grupiranja, (k-t) konture i pravokutnici oko kontura.

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})}{N-1}} \tag{3-4}$$

Navedenom formulom dobije se standardna devijacija izražena u broju elemenata slike po *y* osi. Slike mogu biti različitih veličina, pa standardna devijacija izražena na navedeni način nije najpogodnija. Kako bi se postigla neovisnost o veličini slike izračunata je relativna standardna devijacija prema (3-5) gdje je *s* standardna devijacija izračunata prema (3-4), a $\bar{x}$ srednja vrijednost svih donjih *y* koordinata.

$$RSD\ [\%] = \frac{s}{\bar{x}} * 100 \tag{3-5}$$

Relativnom standardnom devijacijom u ovom slučaju dobije se standardna devijacija svih donjih *y* pozicija, ali izražena u postocima što osigurava neovisnost o veličini slike. Što je ova vrijednost bliža nuli to znači da su donje *y* koordinate svih pravokutnika ujednačenije. Upravo je ova vrijednost prva značajka korištena u strojnom učenju za potrebu klasifikacije. Na slikama na kojoj se nalazi tekst, ova vrijednost bi trebala biti bliža nuli nego na slikama gdje se nalazi pozadina. Drugim riječima, donje *y* koordinate pravokutnika oko slova su otprilike iste. Postoje neke iznimke kao npr. pravokutnik oko točke koja je dio malog slova „i" ili slova „j" te pravokutnik oko malih pisanih slova kao što su „q", „g", „j", „y" i „p" čije su donje *y* koordinate niže od ostalih slova. Bez obzira na ove iznimke, relativna standardna devijacija donjih *y* koordinata bi trebala biti bliža nuli nego ona na slikama koje sadrže konture pozadine čije su donje *y* koordinate puno nepravilnije što je vidljivo na slici 3.2. Druga i treća značajka temeljene su na pretpostavci da su pravokutnici oko slova približno jednakih površina te približno jednakih visina. Približna jednakost površina i visina je također izražena pomoću relativne standardne devijacije izračunate prema (3-4) i (3-5) gdje su x_i pojedine površine, odnosno visine pravokutnika, $\bar{x}$ srednje vrijednosti svih površina, odnosno visina pravokutnika te N ukupan broj pravokutnika. Pri računanju ovih dviju značajki, također vrijedi da što je ona bliža nuli, to znači da su vrijednosti ujednačenije, odnosno da su površine i visine ujednačenije. Ove dvije značajke će biti bliže nuli kad su u pitanju pravokutnici oko slova, no također postoje neke iznimke. Površina malog slova „i" te točke na malim slovima „i" te „j" bit će puno manje od ostalih, no iz slike 3.2. je vidljivo da se površine pravokutnika na slikama pozadine puno više razlikuju te će relativna standardna devijacija biti puno veća u odnosu na onu sa slike gdje se nalazi tekst. Također vrijedi i za visinu pravokutnika. Postoje iznimke kao točka na malim slovima „i" te „j". Visina pravokutnika je u suštini povezana s prvom značajkom, odnosno donjom *y* koordinatom pravokutnika. Postoje slučajevi gdje sličnost pravokutnika oko slova nije najbolje opisana donjom *y* koordinatom, a puno je bolje opisana visinom. Jedan primjer je sličnost pravokutnika oko malog slova „t" i malog slova „g". Donja *y* koordinata pravokutnika oko ovih slova se dosta razlikuje dok su njihove visine jednake ili vrlo slične. Četvrta značajka temelji se na pretpostavci da se na slici nalazi jedna riječ bez razmaka između slova. Na temelju ove pretpostavke moguće je zaključiti da se pravokutnici oko slova ponavljaju u nekakvim pravilnim razmacima po *x* osi. Prvi korak pri računanju te pravilnosti je računanje centralne *x* koordinate svih pravokutnika. Nakon toga potrebno je sortirati te koordinate te izračunati udaljenosti svih susjednih centralnih *x* koordinata. Zadnji korak je isti kao i kod izračuna prethodnih značajki, a to je računanje relativne standardne devijacije. U ovom slučaju računa se relativna standardna devijaciju udaljenosti pravokutnika po *x* osi, koja će biti bliža nuli što su razmaci pravilniji. Kao i kod prethodnih značajki i u ovom slučaju postoje iznimke kao što su

udaljenost točke koja je dio malog slova „i“ te ostalog dijela malog slova „i“ koja bi trebala biti nula i samim time bi odskakala od ostalih udaljenosti, no kao što je vidljivo na slici 3.2, udaljenosti pravokutnika oko slova su pravilnije nego udaljenosti pravokutnika na slikama s pozadinom. Na slici 3.2. je vidljivo da postoje slučajevi gdje nije pronađena ni jedna kontura i pravokutnik oko nje te je slika potpuno crna. To se događa kad je broj grupa *k* prevelik, odnosno elementi slike su već podijeljeni u maksimalan broj grupa. U tom slučaju ne računaju se relativne standardne devijacije, nego je potrebno vrijednosti značajki postaviti na neku određenu vrijednost. Ta vrijednost je u ovom radu postavljena na temelju analize značajki iz skupa slika za treniranje gdje je broj pravokutnika veći ili jednak jedan. U takvoj situaciji cilj je da navedena slika bude klasificirana kao pozadina, a ne tekst, pa je samim time potrebno tu vrijednost postaviti na neku koja sigurno spada u klasu pozadine. Vrijednost nula je idealna vrijednost značajke koja predstavlja sliku s tekstom, tako da je vrijednost sa slike koja ne sadrži ni jednu konturu potrebno postaviti na neku koja sigurno nije nula. Analizom su utvrđene vrijednosti značajki sa slika koje sadrže dijelove pozadine te uočene su vrijednosti svih značajki koje su najudaljenije od onih koje predstavljaju slike s tekstom. Upravo su te najudaljenije značajke odabrane kao vrijednosti za slike gdje nije pronađena ni jedna kontura i pravokutnik oko nje.

4. REZULTATI

U ovom radu korištena su dva skupa slika. Prvi skup slika korišten je isključivo za potrebe ispitivanja metode povećanja slike predstavljene u potpoglavlju 3.1, izoštravanja slike i zamućivanja slike koje su predstavljene u potpoglavlju 3.2. Navedeni skup ne sadrži slike sa šarenom pozadinom. Drugi skup slika sadrži isključivo slike sa šarenom pozadinom na kojem su ispitane metode povećanja slike, izoštravanja slike te metoda grupiranja predstavljena u potpoglavlju 3.3.

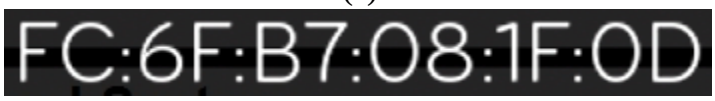
4.1. Rezultati na slikama s jednobojnom pozadinom

Skup slika s jednobojnom pozadinom sastoji se od 83 slike koje je moguće podijeliti u pet grupa o obzirom na njihove karakteristike. U tablici 1. prikazane su karakteristike pojedinih grupa, dok slika 4.1. prikazuje po jednu reprezentativnu sliku za pojedinu grupu.

Tab. 4.1. Karakteristike pojedinih grupa.

Grupa	**Prosječna visina [elementi slike]**	**Tekst / pozadina**
1	22,44	Bijel / crna
2	46,56	Bijel / crna
3	18	Bijel / jednobojna
4	54,42	Bijel / crna
5	54,28	Bijel / jednobojna

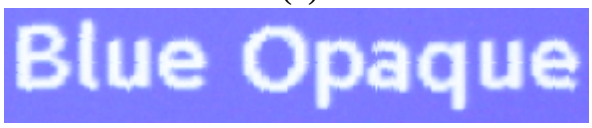
(a)



(b)

(c)

(d)



(e)

Sl. 4.1. (a-e) Reprezentativne slike pojedinih grupa.

Prva se grupa sastoji od slika niske rezolucije i loše kvalitete. Druga i četvrta grupa se razlikuju od prve u rezoluciji. Prosječna rezolucija slika u prvoj grupi je približno dva puta manja od rezolucije slika u drugoj i četvrtoj grupi. Također, slike u drugoj i četvrtoj grupi sadrže tekst s relativno puno znakova koji ne spadaju u alfanumeričke znakove, tj. interpunkcijske znakove poput točke, dvotočke i sl. Treća i peta grupa sadrže slike s približno jednobojnom pozadinom. Razlika ovih dviju grupa je ta da slike u trećoj grupi imaju značajno manju rezoluciju. Prepoznavanje teksta je pokrenuto sa *Tesseract* verzijom 3.5 i verzijom 4.0 i to na originalnim slikama, kao i na slikama nakon korištenja metode povećanja slike, izoštravanja i zamućivanja. Navedene metode primijenjene su koristeći programski paket *Matlab*. Povećanje slike izvršeno je funkcijom *imresize* koja koristi bikubičnu interpolaciju. Izoštravanje slike izvršeno je prema izrazu (3-2) koristeći funkciju *imsharpen*. Zamućivanje je izvršeno koristeći funkciju *imfilter* sa filtrom prema izrazu (3-4). Na slikama 4.2. i 4.3. prikazani su primjeri slika nakon primjene metoda izoštravanja i zamućivanja što rezultira s točnim prepoznavanjem teksta.

(a)

(b)

Sl. 4.2. (a) Originalna slika, (b) izoštrena slika.

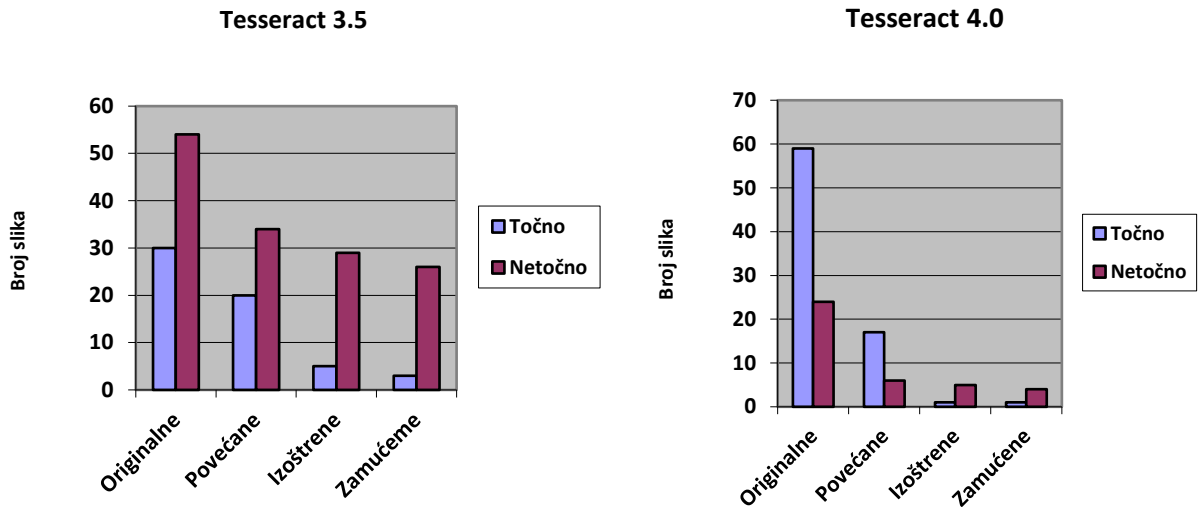
(a)

(b)

Sl. 4.3 (a) Originalna slika, (b) zamućena slika.

Pri korištenju *Tesseract*-a moguće je odabrati različite opcije. Jedna od njih je način segmentacije stranice koju je moguće postaviti na vrijednost koja označava *Tesseract*-u da se na slici nalazi jedna linija teksta. Ova opcija ja korištena u ovom radu. Navedena opcija dostupna je u obje verzije *Tesseract*-a. Verzija 4.0 također omogućava opciju odabira načina rada. Dvije opcije načina rada su korištene u ovom radu: korištenje originalnog *Tesseract*-a i korištenje *Tesseract*-a i *LSTM*-a. Za postizanje boljih rezultata moguće je odabrati i jezik teksta. Jezik teksta na analiziranim slikama je engleski i portugalski, pa su navedeni jezici odabrani pri pokretanju prepoznavanja teksta.

Smatra se da je tekst uspješno prepoznat samo ako su svi znakovi uspješno pročitani. Prepoznavanje tekstualnih znakova prvo je pokrenuto na originalnim slikama korištenjem *Tesseract*-a 3.5 i 4.0. Točnost prepoznavanja *Tesseract*-a 3.5 na originalnim slikama je 35.7%, dok je točnost prepoznavanja *Tesseract*-a 4.0 na istim slikama 70.2%. Rezultati su prikazani na slici 4.3.



(a) (b)

Sl. 4.3. (a) Rezultati Tesseract-a 3.5 i (b) Tesseract-a 4.0.

Metode predobrade slike primijenjene su na slikama na kojima je prepoznavanje teksta netočno te je unaprjeđenje predstavljeno kao broj slika na kojima je prepoznavanje točno podijeljeno s ukupnim brojem slika. Unaprjeđenja su zbrojena nakon primjene pojedine metode. Nakon zbrajanja dobivena su ukupna unaprjeđenja za obje verzije *Tesseract*-a. Metoda povećanja slike primijenjena je na slikama na kojima je prepoznavanje teksta netočno Ako je visina slike manja od 100 elemenata slike te je prepoznavanje teksta pokrenuto ponovno. Povećanjem slike ostvarilo se unaprjeđenje od 23.8% za *Tesseract* 3.5 i 20.2% za *Tesseract* 4.0. Metoda izoštravanja je primijenjena na povećanim slikama na kojima je prepoznavanje teksta i dalje netočno. Metode povećanja slike i izoštravanja ostvarile su unaprjeđenje od 6% za *Tesseract* 3.5 na grupama tri i četiri te 1.2% za *Tesseract* 4.0 na drugoj grupi slika. Metoda zamućivanja je primijenjena na povećanim slikama na kojima je prepoznavanje teksta nakon izoštravanja i dalje netočno. Metoda povećanja i zamućivanja ostvarile su unaprjeđenje od 3.5% za *Tesseract* 3.5 na grupama jedan i tri te 1.2% za *Tesseract* 4.0 na grupi dva. Prethodne tri metode ostvarile su unaprjeđenje od 33.3% za *Tesseract* 3.5 te 22.6% za *Tesseract* 4.0 što sveukupno dovodi to sveukupne točnosti od 69% za *Tesseract* 3.5 te 92.9% za *Tesseract* 4.0. Iz dobivenih rezultata na originalnim slikama može se zaključiti da je *Tesseract* 4.0 značajna nadogradnja na prethodnu verziju 3.5. Korištenje *Tesseract*-a 3.5 na slikama kojima se ovaj rad bavi rezultira lošom točnošću prepoznavanja teksta ukoliko predložene metode nisu primijenjene prije samog prepoznavanja.

Tesseract omogućuje generiranje binarne slike na kojoj se izvršava prepoznavanje teksta. Na slici 4.4. prikazan je primjer binarne slike generirane iz slike loše kvalitete bez primjene nekih od

metoda predobrade čiji je rezultat prepoznavanja teksta netočan te binarnu sliku iste slike nakon primjene metode izoštravanja čiji je rezultat prepoznavanja točan.

(a)

(b)

(c)

Sl. 4.4. (a) Originalna slika, (b) binarna originalna, (c) binarna izoštrena.

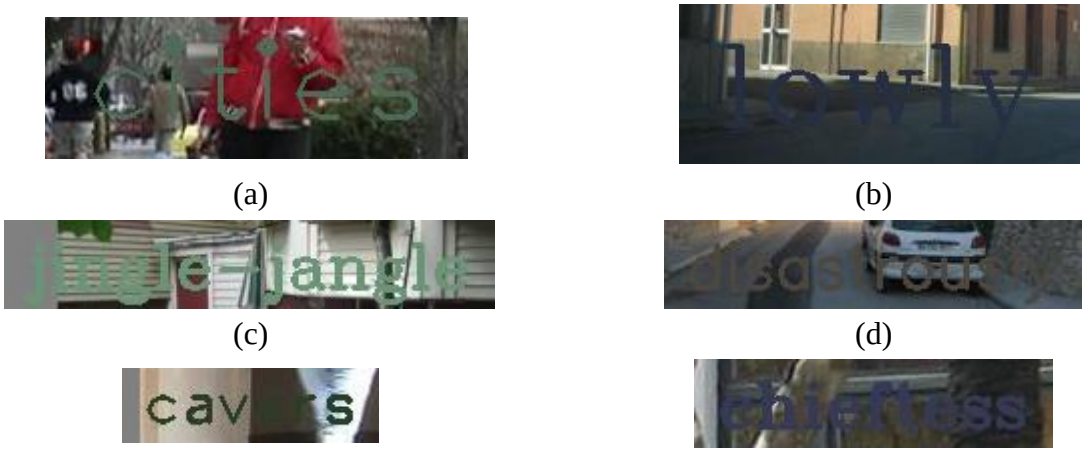
Razlike između binarne originalne slike i binarne izoštrene slike su ljudskom oku teško vidljive, no na temelju rezultata može se zaključiti da su te razlike itekako važne za učinkovitost *Tesseract*-a.

4.2. Rezultati na slikama sa šarenom pozadinom

U ovom radu su ispitane dvije metode za unaprjeđivanje algoritma za prepoznavanje tekstualnih znakova na slikama sa šarenom pozadinom. Prva metoda, koja je opisana u poglavlju 3.3.3., najprije koristi algoritam *k* srednjih vrijednosti za grupiranje elemenata slike sa slike po boji. Nakon toga koristi se algoritam koji je prethodno treniran strojnim učenjem za klasifikaciju slike gdje klase određuju nalazi li se na slici tekst ili pozadina. Nakon identifikacije slike s tekstom, na njoj se pokreće algoritam prepoznavanja teksta. Druga metoda, opisana u poglavlju 3.3.4, također koristi algoritam *k* srednjih vrijednosti za grupiranje elemenata slike sa slike po boji. Nakon grupiranja, generira se *k* slika. Svaka slika će sadržavati elemente slike jedne grupe. Vrijednost elemenata slike bit će vrijednosti centara pojedinih grupa. Na taj način, svaka od generiranih slika sadržavat će jednu od *k* dominantnih boja originalne slike.

4.2.1. Rezultati metodom strojnog učenja

Kao što je opisano u poglavlju 3.3.3., ova metoda realizirana je izradom modela pomoću strojnog učenja korištenjem Bayesovog klasifikatora. Za treniranje korišten je set od 10000 slika. Navedene slike generirane su tako da su iz seta od 10000 fotografija nasumično izrezani dijelovi istih koji će služiti kao šarena pozadina. Preko navedene pozadine potrebno je postaviti jednobojni tekst nasumične boje. Kako su i pozadina i boja teksta potpuno nasumično odabrani, postoje slučajevi gdje bi tekst bio vrlo slične ili čak iste boje kao i dijelovi pozadine. U takvim slučajevima, tekst je teško čitljiv i samom čovjeku. Primjeri takvih slika prikazani su na slici 4.5.



(a) (b)

(c) (d)

(e) (f)

Sl. 4.5. (a-f) Slike sa šarenom pozadinom s tekstom slične boje pozadine.

Kako su navedene slike nerealne za potrebe ovog rada, prije postavljanja teksta, postavlja se crna transparentna slika preko cijele pozadinske slike. Na taj način, pozadina je i dalje šarena, no tekst je uvijek čovjeku čitljiv. Primjer takvih slika prikazan je na slici 4.6.

Veličina teksta je također nasumično odabrana, no u granicama realnih scenarija. Skup slika za testiranje generiran je na isti način kao i skup slika za treniranje, no s različitim tekstom, bojom teksta te pozadinom. Značajke koje su korištene za izradu modela izračunate su iz pravokutnika određenih oko svih kontura na slikama nakon grupiranja algoritmom *k* srednjih vrijednosti kao što je opisano u poglavlju 3.3.3.

(a) (b) (c) (d) (e) (f)

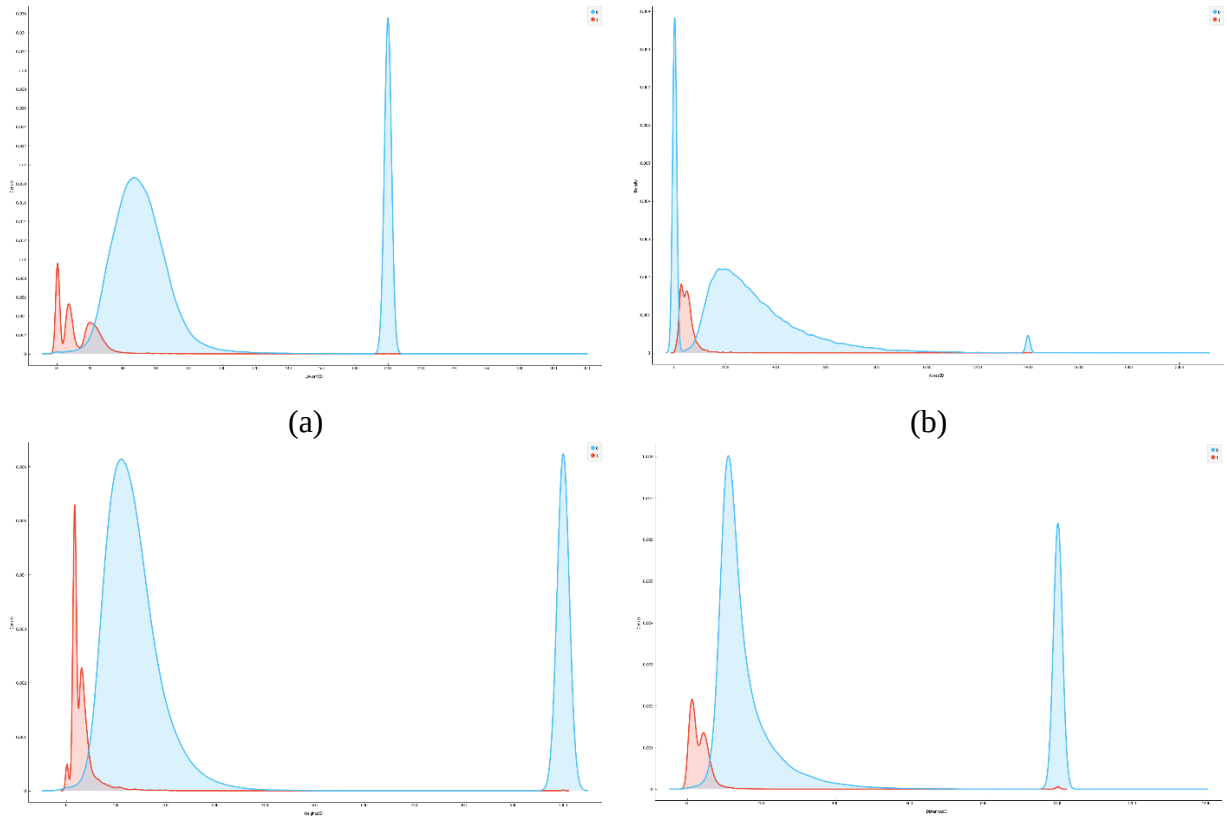
Sl. 4.6. (a-f) Generirane slike dodavanjem crne transparentne slike preko šarene pozadine.

Izračunate značajke su:

1. Relativna standardna devijacija donjih *y* koordinata
2. Relativna standardna devijacija površina
3. Relativna standardna devijacija visina
4. Relativna standardna devijacija udaljenosti

Primjer pravokutnika određenih oko svih kontura na slikama nakon grupiranja te vrijednosti pojedinih značajki prikazan je u prilogu P.4.1.

Pri izradi modela strojnim učenjem ključno je da značajke što je moguće bolje predstavljaju svoju klasu. Na slici 4.7. prikazane su distribucije svih značajki, gdje su crvenom bojom označene značajke izračunate iz slika na kojima je prikazan tekst, a plavom bojom značajke izračunate iz slika na kojima je prikazana pozadina, odnosno dio pozadine. Vidljivo je kako su crvene i plave značajke relativno dobro odvojene i postoji jako malo preklapanja istih, što je bitno za postizanje što boljih rezultata. Testiranje modela izrađenog strojnim učenjem obično se testira nekim od programskih paketa za analizu velikih skupova podataka te izradu i testiranje modela strojnim učenjem. Najčešća procedura je ta da se skup podataka podjeli na skup za treniranje i na skup za testiranje, gdje skup za treniranje sadrži barem 70% ukupnih podataka. Nakon podjele, postoje gotovi alati za testiranje, koji koriste prethodno istrenirani model za prikaz detaljnih rezultata te ukupne točnosti. U ovom radu nije korišten ovaj način testiranja modela. Kao što je ranije spomenuto, klasifikator koji se koristi u ovom radu je Bayesov klasifikator koji klasificira određeni skup značajki računajući vjerojatnosti da taj skup pripada pojedinoj klasi. Navedene vjerojatnosti moguće je dobiti pri korištenju samog modela.



(a) (b)

(c) (d)

Sl. 4.7. (a) Distribucije relativne standardne devijacije donjih y koordinata pravokutnika, (b) površina pravokutnika, (c) visina pravokutnika, (d) udaljenosti pravokutnika.

Kao što je ranije opisano, model se koristi nakon grupiranja algoritmom *k* srednjih vrijednosti kako bi se odredilo koja slika u skupu od *k* slika sadrži tekst. Samo jedna od *k* slika sadrži tekst, a sve ostale sadrže dijelove pozadine. Kada bi se odmah nakon grupiranja koristio model za klasifikaciju, postojala bi mogućnost da iz skupa od *k* slika više slika bude klasificirano kao tekst ili da ni jedna slika ne bude klasificirana kao tekst. Da bi se izbjegle navedene situacije, korištene su vjerojatnosti koje Bayesov algoritam može dati kao odgovor. Nakon grupiranja slika u *k* grupa, sve slike se šalju izrađenom modelu, spremaju se vjerojatnosti da se na pojedinoj slici nalazi tekst te se jednostavno uzima slika na kojoj je najveća vjerojatnost da se nalazi tekst. Skup slika za testiranje sastoji se od 1000 slika generiranih na način koji je opisan ranije u ovom poglavlju. Algoritam grupiranja je izvršen na navedenom skupu slika uz parametar *k* jednak deset te se u konačnici skup slika za testiranje sastoji od 10000 slika od kojih neke sadrže tekst a neke pozadinu. Točnost modela određena je brojem točnih identifikacija slike s tekstom nakon grupiranja u generiranom skupu slika za testiranje od 1000 slika. Nakon izvršenog testiranja dobivena je točnost identifikacije slike s tekstom od 95,5%.

4.2.2. Rezultati metodom uzorkovanja boja

Uzorkovanje boja je metoda smanjivanja broja boja na slici kako bi dominantne boje bile izraženije. Kao što je ranije spomenuto, tekst na slikama korištenih u ovom radu zauzima značajan dio slike, pa se može zaključiti da će jedna od dominantnih boja na slici biti boja teksta. Metoda je detaljnije opisana u poglavlju 3.3.4. Skup slika koji je korišten za testiranje ove metode sastoji se od 1000 slika generiranih na isti način kao i skup slika za testiranje metode identifikacije teksta kako je opisano u poglavlju 4.2.1 no s različitom pozadinom, tekstom i bojom teksta. Nakon primjene metode uzorkovanja boja algoritmom *k* srednjih vrijednosti, prepoznavanje teksta izvršeno je na svakoj slici, odnosno *k* puta. Tekst koji je potrebno prepoznati unaprijed je poznat, pa se pri svakom pokretanju prepoznavanja teksta uspoređuju prepoznati tekst i očekivani tekst. Ako je na jednoj od *k* slika prepoznati tekst jednak očekivanom tekstu, proces se prekida i prepoznavanje teksta smatra se točnim. Ako ni na jednoj od *k* slika nije prepoznat tekst koji se očekuje, prepoznavanje teksta smatra se netočnim. Pri korištenju algoritma *k* srednjih vrijednosti potrebno je unaprijed odrediti broj *k*. U ovom radu broj *k* nije fiksno određen. U prvom pokušaju uzorkovanja boja i prepoznavanja teksta broj *k* je dva. Ako je prepoznavanje netočno, cijeli proces se ponavlja, ali tako da je broj *k* povećan za jedan. Povećanje broja *k* izvršava se dok *k* ne dostigne vrijednost pet. Ako proces dostigne navedenu vrijednost te ni na jednoj slici nije prepoznat traženi tekst, prepoznavanje se smatra netočnim. Maksimalna vrijednost boja *k* odabrana je na temelju analize skupa slika kojima se ovaj rad bavi. Analiza je pokazala kako vrijednost *k* koja je veća od pet ne donosi nikakva poboljšanja. Testiranjem ove metode na skupu od 1000 slika dobila se točnost prepoznavanja od 94,4%.

5. ZAKLJUČAK

U ovom radu ispitana je točnost prepoznavanja teksta na slikama dohvaćenih sa STB uređaja korištenjem *Tesseract*-a. Navedene slike mogu biti niske rezolucije, mogu biti loše kvalitete te mogu sadržavati šarenu pozadinu. Ispitane su tri različite metode predobrade slike za poboljšanje točnosti prepoznavanja teksta na slikama niske rezolucije i loše kvalitete te dvije metode za rješavanje problema šarene pozadine. Korištenje *Tesseract* verzije 3.5 rezultiralo je relativnom niskom točnošću prepoznavanja teksta bez korištenja predloženih metoda predobrade slike. Pokazalo se kako je novija verzija *Tesseract*-a, verzija 4.0, značajna nadogradnja u odnosu na verziju 3.5 s obzirom da su rezultati prepoznavanja značajno veći bez primjene predloženih metoda predobrade. Kombinacija povećanja slike, izoštravanja te zamućivanja poboljšava točnost prepoznavanja teksta za 33.3% za *Tesseract* 3.5 i 22.6% za *Tessract* 4.0. na skupu slika korištenom u ovom radu. Prepoznavanje teksta na slikama sa šarenom pozadinom rijetko rezultira točnim rezultatom bez dodatne predobrade slike. U ovom radu ispitana je metoda koja izdvaja tekst od šarene pozadine. Nakon izvršavanja navedene metode dobivena je točnost prepoznavanja teksta od 94,4%. Ispitana je dodatna metoda automatizacije procesa prepoznavanja teksta na slikama sa šarenom pozadinom koja identificira sliku na kojoj je potrebno pokrenuti algoritam prepoznavanja teksta nakon izdvajanja teksta od šarene pozadine. Testiranje navedene metode rezultiralo je s točnošću identifikacije slike s tekstom od 95,5%. Može se zaključiti da je prepoznavanje tekstualnih znakova i dalje veliki izazov u području računarstva te ne postoji program čija je točnost prepoznavanja 100% u svim slučajevima. Unaprjeđenje točnosti je moguće, ali ovisi o primjeni. Ono se najčešće ostvaruje predobradom slike na kojoj je potrebno pokrenuti algoritam prepoznavanja teksta.

LITERATURA

[1] Z. Podobný, Tesseract OCR [online]. GitHub, 2017., dostupno na: https://github.com/tesseract-ocr/tesseract [10.9.2017.]

[2] Z. Podobný, Tesseract: ImproveQuality [online] GitHub, 2017., dostupno na: https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality. [10.9.2017.]

[3] M. Shen i H. Lei, Improving OCR performance with background image elimination, 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), str. 1566–1570.

[4] D. Chen, H. Bourlard i J. P. Thiran, Text identification in complex background using SVM, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 2, str. 306-309, 2001.

[5] Q. Ye, W. Gao i Q. Huang, Automatic text segmentation from complex background, 2004 International Conference on Image Processing, sv. 5, str. 2905–2908, 2004.

[6] N. Shivananda i P. Nagabhushan, Separation of Foreground Text from Complex Background in Color Document Images, 2009 Seventh International Conference on Advances in Pattern Recognition, str. 306–309, 2009.

[7] J. Matas, O. Chum, M. Urban i T. Pajdla, Robust Wide Baseline Stereo from Maximally Stable Extremal Regions, Electronic Proceedings of The 13th British Machine Vision Conference University of Cardiff, Cardiff, 2002.

[8] R. C. G. Richard E. Woods, Digital Image Processing, Pearson Education, New Jersey, 2007.

[9] A. Moore, Clustering: - K-means [online] A Tutorial on Clustering Algorithms, Italija, dostupno na: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html. [10.9.2017.]

[10] OpenCV team, OpenCV: Contours [online], OpenCV tutorials, dostupno na: http://docs.opencv.org/3.2.0/d4/d73/tutorial_py_contours_begin.html [10.9.2017.]

SAŽETAK

Točnost algoritma za prepoznavanje tekstualnih znakova (engl. *Optical character recognition - OCR*) na slikama dohvaćenih sa *set-top box* (STB) uređaja igra vrlo važnu ulogu pri testiranju istih. Međutim, pokretanje algoritma za prepoznavanje tekstualnih znakova na takvim slikama često rezultira niskim postotkom točnosti. Uzrok tomu su navedene slike koje mogu biti niske rezolucije, loše kvalitete ili mogu sadržavat šarenu pozadinu. U ovom radu preporučene su četiri metode predobrade slike kako bi se poboljšala točnost prepoznavanja tekstualnih znakova. Ispitane su točnosti na slikama dohvaćenih sa STB uređaja korištenjem programa *Tesseract* 3.5 te relativno nove verzije *Tesseract* 4.0. Na tim slikama točnost prepoznavanja pomoću *Tesseract*-a 3.5 je 35.7%, dok *Tesseract* 4.0 ostvaruje točnost od 70.2%. Metode ispitane u ovom radu unaprjeđuju točnost algoritma za prepoznavanje tekstualnih znakova za 33.3% za *Tesseract* 3.5 te za 22.6% za *Tesseract* 4.0. Također je ispitana metoda koja odvaja tekst od pozadine kada su u pitanju slike sa šarenom pozadinom. Točnost prepoznavanja tekstualnih znakova nakon izvršavanja navedene metode je 94.4%. Također, izgrađen je dodatni korak automatizacije procesa prepoznavanja tekstualnih znakova na slikama sa šarenom pozadinom pristupom strojnog učenja. Navedeni pristup detektira sliku na kojoj je prikazan tekst nakon odvajanja teksta od pozadine. Testiranje istog rezultiralo je s točnošću od 95,5%.

Ključne riječi: Optičko prepoznavanje znakova, slike loše kvalitete, predobrada slike, grupiranje po boji, STB

IMPROVING OPTICAL CHARACTER RECOGNITION PERFORMANCE

ABSTRACT

Efficient Optical Character Recognition (OCR) in images grabbed from Set-Top Boxes (STBs) plays an important role in STB testing. However, running OCR software on such images usually ends with low OCR performance since images can have low resolution, low image quality or colorful background. In order to improve OCR performance, four different image preprocessing methods are proposed. In this paper OCR is performed with Tesseract 3.5 and the relatively new Tesseract 4.0 on the images grabbed from different STBs. On the original images Tesseract 3.5 provides a 35.7% accuracy while Tesseract 4.0 attains a 70.2% accuracy. The proposed preprocessing methods improve OCR performance by 33.3% for Tesseract 3.5 and 22.6% for Tesseract 4.0 on the available images. On images with colorful background, a method which separates text from background is proposed. OCR performance after separating text from background results with an accuracy of 94,4%. Aditionally, a step of automation in a process of reading text from images with colorful background was build with an approach of machine learning. This approach detects the image containing text after the separation from the background with the accuracy of 95,5%.

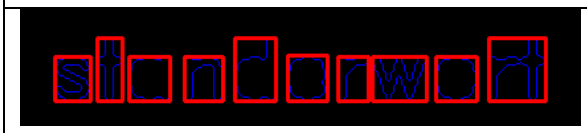
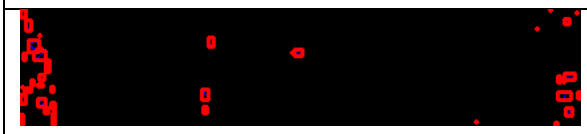
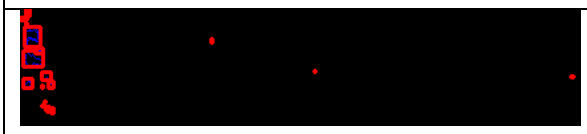
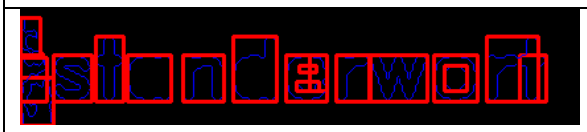
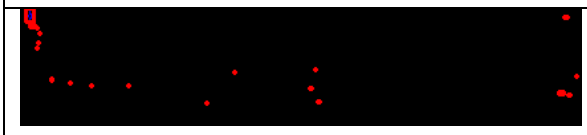
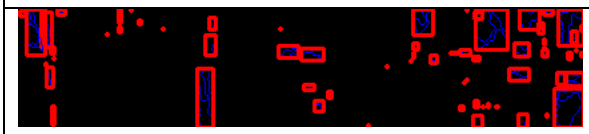
Keywords: OCR, Tesseract, low quality images, image preprocessing, color clustering, STB

ŽIVOTOPIS

Matteo Brisinello rođen je 13. prosinca 1993. godine u mjestu Belluno u Italiji. Pohađao je osnovu školu "Scuola elementare Farra d'Alpago" (1999. - 2003.), "OŠ Lipik" (2003. - 2008.), srednju školu "Tehnička škola Daruvar" (2008. - 2012.). Završio je sveučilišni preddiplomski studij računarstva na „Elektrotehničkom fakultetu Osijek“. Trenutno je stipendist tvrtke RT-RK Osijek. Natjecao se na Elektrijadi 2016. godine u području informatike. Aktivno govori i piše engleski i talijanski jezik.

PRILOZI

Prilog 4.1.

Konture i pravokutnici oko kontura	Relativna standardna devijacija donjih *y* koordinata	Relativna standardna devijacija površina	Relativna standardna devijacija visina	Relativna standardna devijacija udaljenosti
	0.000000	32.901199	17.964468	16.071430
	27.416695	462.929688	126.327774	120.712349
	31.645191	258.290863	123.035881	90.994446
	51.055305	99.981018	62.143921	267.119202
	50.254486	193.950348	100.878189	240.090057
	200.000000	1000.000000	1000.000000	1000.000000
	19.511318	63.569252	44.646168	79.555305
	50.546223	260.856659	126.288033	191.723541
	62.723045	194.659821	116.173111	141.803711
	64.928101	198.926880	156.577240	161.713058