



**УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**

**УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације**

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Сандра Кукољ

Број индекса: 12570

Тема рада: Детекција региона текста и његова сегментација у ТВ слици

Ментор рада: др Јелена Ковачевић



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

Редни број, РБР :			
Идентификациони број, ИБР :			
Тип документације, ТД :	Монографска документација		
Тип записа, ТЗ :	Текстуални штампани материјал		
Врста рада, ВР :	Завршни (Bachelor) рад		
Аутор, АУ :	Сандра Кукољ		
Ментор, МН :	др Јелена Ковачевић		
Наслов рада, НР :	Детекција региона текста и његова сегментација у ТВ слици		
Језик публикације, ЈП :	Српски / латиница		
Језик извода, ЈИ :	Српски		
Земља публиковања, ЗП :	Република Србија		
Уже географско подручје, УГП :	Војводина		
Година, ГО :	2012.		
Издавач, ИЗ :	Ауторски репринт		
Место и адреса, МА :	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО : <small>(поглавља/страна/ цитата/табела/слика/графика/прилога)</small>	5/30/10/2/13/0/0		
Научна област, НО :	Електротехника и рачунарство		
Научна дисциплина, НД :	Рачунарска техника		
Предметна одредница/Кључне речи, ПО :	Препознавање карактера, сегментација текста, обрада слике		
УДК			
Чува се, ЧУ :	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН :			
Извод, ИЗ :	Овај рад описује алгоритам за сегментацију текста и представља припремну фазу за препознавање текста у ТВ слици, која је имплементирана у систему за ВВТ тестирање.		
Датум прихватања теме, ДП :			
Датум одбране, ДО :			
Чланови комисије, КО :	Председник:		
	Члан:	Иван Каштелан	Потпис ментора
	Члан, ментор:	др Јелена Ковачевић	



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

Accession number, ANO :	
Identification number, INO :	
Document type, DT :	Monographic publication
Type of record, TR :	Textual printed material
Contents code, CC :	Bachelor Thesis
Author, AU :	Sandra Kukulj
Mentor, MN :	Jelena Kovačević, Ph.D.
Title, TI :	Text region detection and its segmentation in TV images
Language of text, LT :	Serbian
Language of abstract, LA :	Serbian
Country of publication, CP :	Republic of Serbia
Locality of publication, LP :	Vojvodina
Publication year, PY :	2012.
Publisher, PB :	Author's reprint
Publication place, PP :	Novi Sad, Dositeja Obradovica sq. 6
Physical description, PD : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	5/30/10/2/13/0/0
Scientific field, SF :	Electrical Engineering
Scientific discipline, SD :	Computer Engineering, Engineering of Computer Based Systems
Subject/Key words, S/KW :	Character recognition, text segmentation, image processing, OCR
UC	
Holding data, HD :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia
Note, N :	
Abstract, AB :	This paper presents an algorithm for text segmentation and presents a preparation phase for text detection in images, implemented in a system for BBT testing.
Accepted by the Scientific Board on, ASB :	
Defended on, DE :	
Defended Board, DB :	President:
	Member: Ivan Kaštelan
	Member, Mentor: Doc. Jelena Kovačević
	Menthor's sign

Zahvalnost

Želela bih da zahvalim Ivanu Kaštelanu na stručnoj pomoći prilikom izrade ovog rada, kao i svim kolegama koji su doprineli mom radu. Takođe bih htela da se zahvalim roditeljima na njihovoj podršci tokom studiranja.

SADRŽAJ

1. Uvod	6
2. Teorijske osnove	7
3. Koncept rešenja	9
3.1 Pred-obrađena slika	14
3.2 Detekcija ivica	15
3.3 Lokalizacija i detekcija tekstualnih regiona	16
3.4 Računanje linijskih histograma	17
4. Programsko rešenje	19
4.1 Funkcija RGB2YUV	19
4.2 Funkcija Lpfiltering	19
4.3 Funkcija EdgeImage	20
4.4 Funkcija Labeling	21
4.5 Funkcija Rectangle	21
4.6 Dopunjavanje blokova teksta	22
4.7 Funkcija AdaptiveThreshold	23
4.8 Funkcija CalculateLineHistogram	24
5. Rezultati	25
Zaključak	28
Literatura	30

SPISAK SLIKA

Slika 1. Predloženi algoritam obrade slike

Slika 2. Originalna slika

Slika 3. Slika dobijena algoritmom pronalaska ivica

Slika 4. Detektovani regioni slike sa tekstom

Slika 5. Histogram horizontalnog projekcionog profila

Slika 6. Histogram vertikalnog projekcionog profila

Slika 7. Ukupan projekcioni profil

Slika 8. Gaussian-ova maska

Slika 9. Izgled orijentacije susednih piksela u odnosu na posmatrani piksel X_{ij}

Slika 10. Dobijena slika nakon primene adaptivnog praga

Slika 11. Poređenje dobijenih rezultata

Slika 12. Aplikacija za ispitivanje

Slika 13. Slike DUT_0_6, DUT_0_25, DUT_0_26, DUT_0_36, redom

SPISAK TABELA

Tabela 1. Koordinate blokova za predstavljenu sliku

Tabela 2. Poređenje tačnosti rezultata

SKRAĆENICE

OCR – Optical Character Recognition (mehanički ili elektronski prevodilac)

DLL – Dynamic Link Library (biblioteka za dinamičko povezivanje)

CCA – Connected Component Analysis (analiza povezanih komponenti)

DTV – Digital Television (digitalna televizija)

BBT – Black Box Testing (ispitivanje metodom crne kutije)

STB – Set Top Box (uređaj za prijem digitalnog TV signala)

DUT – Device Under Test (ispitivani uređaj)

HD – High-Definition (visoka rezolucija)

LCS – Longest Common Subsequence (najduža zajednička podsekvenc)

1. Uvod

U ovom zadatku je bilo potrebno proučiti postojeće algoritme za detekciju i segmentaciju teksta u odnosu na pozadinu slike preuzete sa TV ekrana. Cilj zadatka je bio implementirati algoritme pred-obrade u C programskom jeziku, i time pripremiti TV slike za postojeći programski modul prepoznavanja teksta (eng. *Optical Character Recognition (OCR) Dynamic Link Library (DLL)*), realizovan u okviru sistema za automatsko ispitivanje i proveru ispravnosti rezultata (eng. *Black Box Testing (BBT)*). Na kraju je bilo neophodno ispitati performanse datog OCR modula sa implementiranim algoritmima za pred-obradu slike.

Tekst sadržan unutar slike je važan za razumevanje i izdvajanje značajnog multimedijalnog sadržaja, što je detaljnije opisano u radu [1]. Takav tekst je od značaja prilikom pretrage digitalne arhive slika, baza podataka videa ili internet stranica. Sadržaj slike može biti perceptualan, kao npr. boja, teksture, oblici, itd., ili semantički, kao npr. tekst ili objekti. Jedan od problema je varijacija u fontu, kao i u formatu teksta (postojanje različitih veličina, stilova, orijentacija, ...), a postoje i efekti alfa kanala (prozračnosti) u razmatranim slikama sa TV menijima. Često se tekst nalazi na pozadini koja je različitih nijansi, ili postoje neke konture koje mogu predstavljati veliki problem pri detekciji teksta. Uz to je moguća i slaba rezolucija slike (kvalitet).

U ovom radu predstavljena je pred-obrada slike radi poboljšanja detekcije teksta u slici i njegove segmentacije. Organizacija rada je sledeća: u drugom poglavlju su date teorijske osnove, dok je u trećem poglavlju opisan koncept rešenja.

2. Teorijske osnove

Tekst u slici sadrži važne i korisne informacije. U TV slikama se mogu naći dve vrste teksta-onaj koji se slučajno javlja u slici, kao scena, i on ne predstavlja bitne informacije sadržaja slike, dok je druga vrsta upravo nama važan sadržaj, tekst izdvojen veštački. Ekstrakcija teksta je sposobnost izdvajanja njegovog sadržaja van slike ili dokumenta. Tekstualni podaci su posebno interesantni, jer tekst služi tome da bi jasno opisao sadržaj slike. Slike, kao i njihov sadržaj, mogu biti sive ili u boji. Slike su bitske dubine 24, što znači da je svakom pikselu dostupno 16.7 miliona boja, sačinjenih od 256 nijansi crvene, 256 nijansi zelene i 256 nijansi plave. Sve boje su načinjene od kombinacija ovih osnovnih boja, pa je bela boja predstavljena sa 255,255,255, crna sa 0,0,0, crvena sa 255,0,0, zelena sa 0,255,0, a plava sa 0,0,255, dok se kombinacijom npr. crvene i zelene dobija žuta(255,255,0). Svaka vrednost od 0-255 sadrži 8 bita, tako da je ukupan zbir prostora boja koji je potreban da se definiše boja svakog piksela 24 bita.

Sadržaj može biti u formi objekata, boje, teksture, oblika, kao i odnosa između njih. Tekstualni podaci unutar slike mogu biti različitih stilova fonta, veličine, orijentacije, boje, i nasuprot kompleksne pozadine.

Lakša lokalizacija i pojednostavljenje pozadine je ključno za pristup automatskoj detekciji teksta. Da bi se tekst lakše prepoznao OCR sistemom, slika sa tekstom mora ispunjavati određene zahteve, kao što su uniformna pozadina i tekst, sa visokim kontrastom između teksta i pozadine.

Tekst poseduje uobičajene karakteristike po kojima se razlikuje, u pogledu informacija o orijentaciji i učestanosti, kao i prostorne kohezije. Prostorna kohezija se odnosi na činjenicu da su tekstualni karakteri određenih reči bliže jedno drugome i sličnih su visina, orijentacija i razmaka.

Različiti pristupi ekstrakcije tekstualnih regiona predloženi su na osnovu osnovnih osobina teksta. Dva osnovna pristupa ekstrakcije teksta su metod povezanih komponenti (ili sličnost intenziteta) i metod zasnovan na ivicama. Pored ovih, postoje i metodi zasnovani na uglovima i metodi zasnovani na teksturi. Osnovni i neophodni uslovi su upravo pomenute karakteristike teksta, date grafičkim prikazom na slici.

Metod zasnovan na ivicama je fokusiran na pretragu onih područja koja imaju visok kontrast između teksta i pozadine. Na ovaj način, ivice slova su identifikovane i spojene. Kada su svi regioni prepoznati, karakteristike (osobine) prostorne kohezije se primenjuju kako bi odbacili lažne objekte. Detekcija teksta vrši se pronalaskom vertikalnih ivica.

Metod zasnovan na povezanim komponentama koristi pristup koji iterativno spaja setove povezanih piksela pomoću kriterijuma homogenosti koji vodi ka kriterijumu ravnih zona ili povezanih komponenti (CCA). Na kraju iterativne procedure, sve ravne zone su identifikovane. Takođe, i u ovom slučaju je primenjena prostorna kohezija, odnosno njene karakteristike.

Obe metode (metod zasnovan na ivicama i metod povezanih komponenti) mogu se klasifikovati u istu grupu, odnosno, metode zasnovane na regionima. Uopšteno, kod ovih metoda se karakteri detektuju kao homogeni regioni koji zadovoljavaju određena heuristična ograničenja. Pretpostavlja se da pikseli svakog karaktera imaju sličnu boju i da se mogu izdvojiti iz pozadine, segmentacijom slike ili grupisanjem boja. Rezultujući monohromatski regioni su selektovani kao karakteri sa nekim jednostavnim heurističnim ograničenjima, kao što su veličina i odnos visina-širina regiona ili osnovnih linija. Ovi metodi ne samo da pronalaze tekstualne regione, već i vrše segmentaciju karaktera iz pozadine (izdvajaju ih). Ipak, monohromatsko ograničenje ne može uvek biti zadovoljeno, i time ovi metodi nisu robusni nad složenim pozadinama.

Detekcija teksta odnosi se na određivanje postojanja teksta u određenom okviru (obično se detekcija teksta koristi za sekvence slika). Lokalizacija teksta je proces određivanja pozicije teksta u slici i generisanja okvira oko takvog teksta. Ekstrakcija teksta je faza kada se tekstualne komponente izdvajaju iz pozadine. Potrebno je poboljšanje izdvojenih tekstualnih komponenti zbog mogućnosti da je slika male rezolucije i zašumljena. Nakon ovoga, izdvojene tekstualne slike se mogu transformisati u čist tekst pomoću OCR tehnologije.

Nakon prilagođavanja slike za kasniju ekstrakciju samog teksta, dolazi se do alata pod nazivom Tesseract [2] koji služi za prepoznavanje izdvojenih delova slike u kojima se tekst nalazi. Pomoću ovog alata je takođe moguće vršiti pred obradu slike, kako bi se dobila što jasnija slika (uprošćena), koja je samim tim pogodnija za dalju obradu.

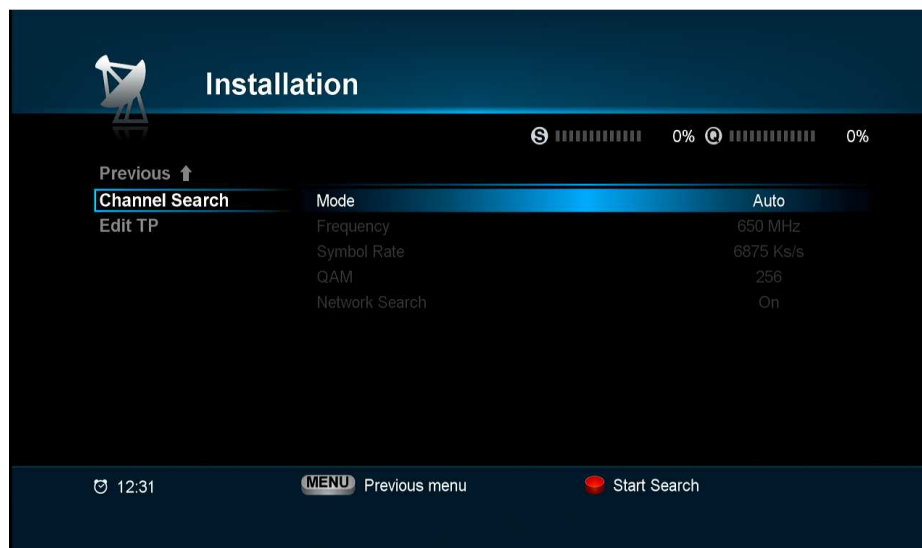
3. Koncept rešenja

Pristup koji je predstavljen za lokalizaciju teksta spada u metod zasnovan na ivicama. Dizajn sistema je baziran na sledećim pretpostavkama: (a) ulaz sistema može biti slika u boji ili siva slika, (b) moguće je detektovati samo tekst koji ima horizontalno poravnanje, i (c) tekst koji je manji od određenog (malog) fonta, neće biti detektovan.

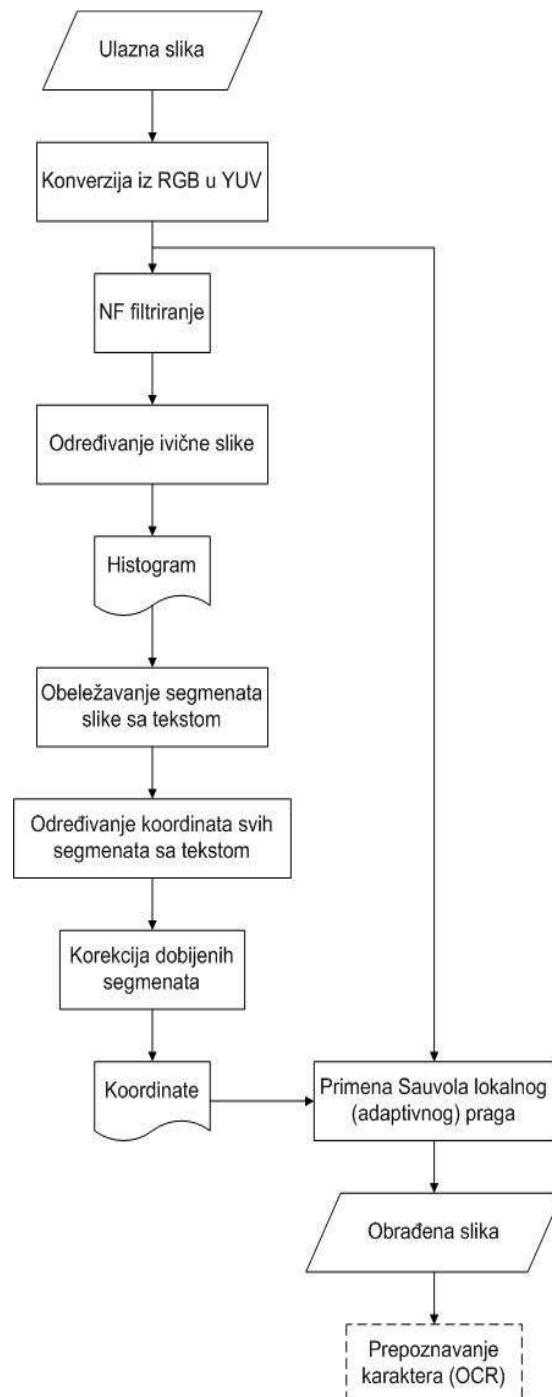
Na slici 2. dat je prikaz realizovanog algoritma.

Koraci ovog pristupa su:

1. Pred-obrađivanje slike – Na slici 1. data je originalna slika nad kojom je vršena obrada. Ukoliko podaci o slici nisu već predstavljeni u YUV prostoru boja (luma i hroma, gde U i V kanali daju informaciju o boji), oni se konvertuju u ovaj prostor boja odgovarajućom transformacijom. Ovaj sistem koristi samo osvetljaj u DTV, (eng. *luminance*, odnosno Y kanal od YUV) tokom daljnje obrade. Siva slika se dobija samo na osnovu informacija o osvetljaju i kontrastu. Potom se vrši niskofrekventno filtriranje Gauss-ovim filtrom, čiji su koeficijenti dobijeni iz Matlab simulatora na osnovu prethodno proverenih performansi nad slikom i iskorišćeni u svrhu uklanjanja šuma u datoj slici. Maska dobijena na ovaj način se pokazala kao najpogodnija iz razloga što se redukuje šum u slici u obliku izolovanih piksela koji imaju drugačiju boju od svojih suseda, kao i suvišna segmentacija duž ivica.



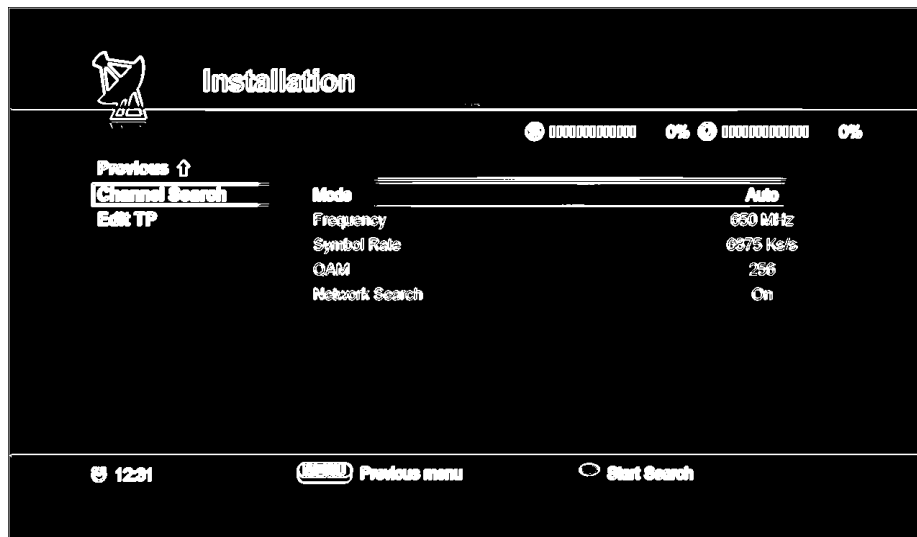
Slika 1. Originalna slika



Slika 2. Predloženi algoritam obrade slike

2. Detekcija ivica – Kao jedan od algoritama detekcije teksta zasnovan na detekciji ivica, posmatraju se pozicije na kojima postoji visok nivo kontrasta između susednih piksela, kako bi se ustanovila pozicija teksta. Generalno, tekst je sastavljen od velikog broja ivica.

U ovom koraku se pažnja obraća na područja u kojima se tekst može pojaviti. Siva slika se konvertuje u sliku koju čine ivice pomoću algoritma predloženog u ovom koraku. Algoritam je zasnovan na činjenici da je kontrast između karaktera i susednih kontura velik. Kao rezultat, svi pikseli karaktera, kao i neki pikseli koji nisu karakteri (sa takođe velikim kontrastom), sačinjavaju ivičnu sliku. U ovakvoj slici vrednosti svakog piksela se zamenjuju najvećom razlikom između posmatranog piksela i njegovih suseda (u vertikalnom, horizontalnom i dijagonalnom pravcu). Nad slikom je primenjen globalni prag (eng. *thresholding*), čime je dobijena crno-bela slika (binarizovana), a taj prag određen je analizom histograma pomenute ivične slike. Tako binarizovana slika (slika 3.) je pogodnija za dalju obradu i segmentaciju zbog izraženog kontrasta između teksta i pozadine.



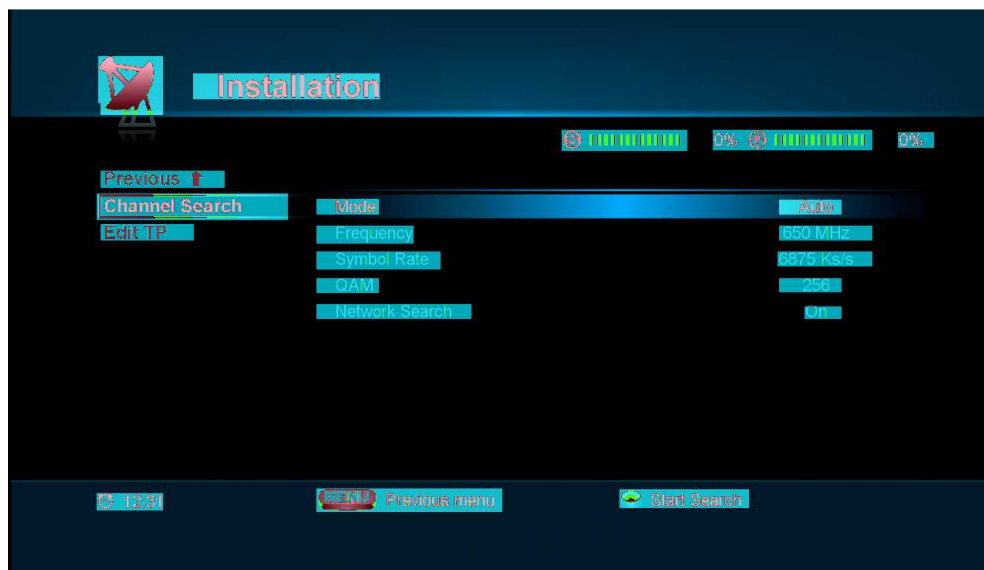
Slika 3. Slika dobijena algoritmom pronalaska ivica

3. Lokalizacija, segmentacija i detekcija tekstualnih regiona – Kod slika koje su snimljene u prirodnom okruženju, pozadina je složenija i onemogućava izdvajanje teksta iz pozadine samo na osnovu podataka dobijenih nakon primenjenog praga. Kako bi se odredile tačne koordinate tekstualnih delova slike, prvo se zapisuju pikseli koji imaju vrednost 1, i na osnovu toga se dobija tačna lokacija teksta.

Raščlanjenu detekciju čine bazna i stvarna detekcija. Primarni cilj pronalaska tekstualnih regiona je brzo izdvajanje očigledne pozadine iz slika i video frejmova. Lokalizacione metode su zbog toga očekivano brze i idealno ne svrstavaju tekstualne regione u pozadine. Sekundarni cilj je da se izdvoje pojedinačne linije teksta kako bi se veličina ovih linija normalizovala. Time, lokalizacija tekstualnih regiona je ostvarena pomoću dva pod-zadatka: izdvajanje kandidata za blokove sa tekстом i izdvajanje pojedinačnih linija sa tekстом.

Predložena lokalizacija teksta se vrši u nekoliko koraka. U prvom koraku se, u okviru posmatranih linija koje čine predefinisani broj piksela, beleže oni pikseli koji su međusobno povezani i ima ih više od predefinisano broja piksela. U drugom koraku kandidati tekstualnih blokova se lociraju pronalaskom prve koordinate bloka, a potom i poslednje koordinate, nakon proverenih ograničenja. Treći korak je korekcija dobijenih segmenata, radi dobijanja donekle pravilnijih blokova. Na osnovu svega se primenjuje adaptivni prag i dobijena slika predstavlja ulaz u OCR.

Tačnost detekcije zavisi od toga u kojoj meri je tekst uspešno detektovan, odnosno da li je klasifikacija tekstualnih karaktera i onih koji to nisu, uspešno izvršena. Potrebno je na ulaz OCR sistemu proslediti sliku čiji je sadržaj, po mogućnosti, samo tekst, kako bi te podatke sistem mogao izdvojiti sa što većom preciznošću.



Slika 4. Detektovani regioni slike sa tekстом

4. Segmentacija i prepoznavanje karaktera – Optical Character Recognition (OCR) je algoritam koji omogućava izdvajanje tekstualnog sadržaja iz dokumenata ili slika u mašinski prepoznat tekst. Korišćen je Google-ov Tesseract OCR iz razloga što je to trenutno najpreciznija besplatna programska podrška za prepoznavanje karaktera. Proces prepoznavanja uglavnom sačinjavaju pretraga najverovatnije sekvence karaktera, na osnovu zadate sekvence ulaznih osobina, leksikona i modela jezika.

Segmentacija je važan zadatak svakog OCR-a. OCR izdvaja delove slike sa tekstom u linije, reči i karaktere. Preciznost OCR-a pretežno zavisi od algoritma za segmentaciju koji se koristi.

Osnovne faze ovakvog sistema su pred-obrađivanje, segmentacija i prepoznavanje. Pred-obrađivanje obuhvata konverziju iz sive slike u binarnu, uklanjanje šuma, treniranje, kao i detekcija izobličene karakteristike i njihovo smanjenje. Faza segmentacije podrazumeva segmentaciju slika sa tekstom u linije, reči i karaktere. Poslednja faza, tj. prepoznavanje, obuhvata izdvajanje odlika, selekciju i klasifikaciju.

Obrada u pravom smislu vrši se nad binarnim slikama, jer su kod njih pikseli pozadine izdvojeni od onih u prednjem planu. Često se za binarizaciju koristi Otsu metod [3], koji bira prag na osnovu koga se vrednosti inteziteta koje prelaze ovaj prag konvertuju u jednu vrednost inteziteta (vrednost bele boje), a one vrednosti inteziteta koje su ispod ovog praga konvertuju se u drugu vrednost inteziteta (vrednost crne boje). S ciljem uklanjanja šuma u slici, koji se javlja iz raznih razloga, često se koristi niskofrekventno filtriranje i poboljšana slika se koristi za dalju obradu. Nakon uklanjanja šuma se eventualno vrši erozija (eng. *thinning*) kako bi se dobile konture širine jednog piksela, odnosno redukovale komponente slike na samo osnovne informacije za dalju analizu i prepoznavanje. Kod skeniranih dokumenata se takođe može javiti iskrivljenost, koja se koriguje definisanjem ugla iskrivljenosti kako bi se tekst poravnao.

Prilikom segmentacije slike zasnovanoj na histogramima, opseg vrednosti piksela (0-255) se raščlani na diskretne intervale. Ovakva segmentacija se vrši nad svakim pikselom. Objekti pri tome moraju biti sastavljeni od piksela sličnih vrednosti. Ovakva nezavisna obrada ignoriše drugu važnu osobinu: pikseli unutar istog objekta bi trebali da budu prostorno povezani.

3.1 Pred-obrađa slike

Binarizacija se obično vrši nad sivim slikama i tada je potrebno odabrati odgovarajući prag kako bi dobili crno-belu sliku. Dobijena slika sadrži bele piksele koji predstavljaju tekst nasuprot pozadini koja je predstavljena crnim pikselima (ili obrnuto). Ovaj korak omogućava lakšu segmentaciju i detekciju teksta, što je objašnjeno u narednim koracima. Nakon što je slika „očišćena“ i binarizovana, dobija se crno-bela slika koja sadrži samo tekst. Tekstualni sadržaj se najpre obeležava bojom kako bi dobili vizuelni prikaz željenih delova slike, ali cilj je da se dobiju koordinate datih pravougaonika koji obuhvataju delove slike sa tekstom. Poželjno je što tačnije uokviriti tekst, odnosno kao rezultat dobiti što manju uokvirenu površinu, što može biti postignuto odsecanjem početnog i krajnjeg dela koji je ranije obeležen. Reč koja se detektuje može sadržati jedan ili više karaktera, a takođe u istoj reči se mogu naći i velika i mala slova. Svaka reč se deli na karaktere i čuva u nizu.

Primena praga predstavlja jedan od najčešćih načina pristupa binarizacije slike. Rezultujuća preciznost i brzina obrade su sasvim dovoljni. Odabir praga nije jednostavan. Postoje dve vrste metoda binarizacije slike: metod sa globalnim pragom (jedna vrednost praga), koji se izvršava nad celom slikom i odlučuje se nad jednom particijom slike, i metod sa lokalnim pragom (adaptivnim), koji uzima u obzir i odlike manjih regiona, kako bi se dobile po mogućnosti različite particije za svaki segment. Svaki piksel se klasifikuje u pozadinsku ili prednju sliku. Lokalne metode problem loše binarizacije rešavaju računanjem praga za svaki piksel posebno, koristeći informacije dobijenih od susednih piksela. Eksperimentalno je dokazano da binarizacija selektovanog dela teksta iz slike korišćenjem Otsu metoda (tehnika primene globalnog praga, nad sivim slikama) daje bolje rezultate u pogledu čitljivosti teksta. Jedan od popularnijih algoritama je Canny ivični filter, koji procenjuje horizontalnu, vertikalnu i dijagonalnu ivicu pronalaskom lokalnog maksimuma pomoću gradijenta inteziteta u ovim pravcima. Drugi algoritmi koriste i morfološke operatore, poput morfološkog gradijenta, koji predstavlja razliku između dilacije i erozije. Ovi metodi proučavaju boje, veličine i prostornu heuristiku u kombinaciji sa izdvojenim ivicama. Još neke od metoda su detekcija zasnovana na teksturi i detekcija zasnovana na boji i heurističnom filtriranju povezanih komponenti.

3.2 Detekcija ivica

Ivice karakterišu ograničenja i stoga se na to obraća velika pažnja jer mogu predstavljati problem pri obradi slike. Detekcija ivica obezbeđuje informacije o poziciji ivica objekata i prisustvu diskontinuiteta, a takođe se značajno redukuje količina podataka i izdvajaju se nebitne informacije, dok se očuvavaju važne strukturne karakteristike slike. S obzirom da je detekcija ivica bitan korak koji prethodi prepoznavanju teksta, neophodno je dobro razumevanje algoritama za detekciju ivica. Radi poređenja predloženog algoritma sa ostalim tehnikama detekcije ivica, sledi kratak opis nekih od popularnijih algoritama.

Canny algoritam detekcije ivica (eng. *Canny edge detection*) daje bolje rezultate nego ostali operatori u većini slučajeva. Nad zašumljenom slikom performanse Canny algoritma su najbolje, a potom dolaze Laplacian of Gaussian (LoG), Sobel-ov, Prewitt-ov i Robert's-ov operator (pogledati [3]). Detekcija ivica se u suštini odnosi na lokalizaciju i pronalazak oštih diskontinuiteta u slici. Diskontinuiteti su nagle promene u intezitetu piksela koje karakterizuju granice objekata u slici.

Klasične metode detekcije ivica podrazumevaju konvoluciju slike sa operatorom (2-D filter), koja je konstruisana da bude osetljiva na velike gradijente u slici, pri čemu se vraćaju vrednosti nula za uniformne delove slike. Promenljive koje su uključene u izbor operatora za detekciju ivica uključuju i orijentaciju ivica, šum okoline i strukturu ivica. Geometrija operatora određuje karakterističan pravac u kome je najviše osetljiv na ivice. Operatori mogu da se optimizuju tako da se traže horizontalne, vertikalne ili dijagonalne ivice. U zašumljenim slikama je selekcija ivica otežana s obzirom na činjenicu da je sadržaj šuma i ivica visokofrekventan. Pokušajima da se šum u slici izbací, dobijaju se izobličene ivice i van fokusa (nejasne).

Međutim, nemaju sve ivice promenu koraka u intezitetu. Efekti, kao što su prelamanje (refrakcija) i slab fokus, mogu rezultovati objektima sa ograničenjima definisanim putem postepene promene inteziteta. U ovim slučajevima operator se mora izabrati tako da bude odgovarajući za postepene promene. Postoje problemi pogrešne detekcije ivica, nedostatak pravih ivica, lokalizacija ivica, dug period obrade, problemi vezani za šum, itd. Iz tog razloga je neophodno utvrditi razlike između različitih tehnika detekcije ivica i analizirati performanse ovih tehnika pod različitim uslovima. Postoje različiti algoritmi detekcije ivica, ali većina ovih metoda se može svrstati u dve kategorije:

- Detekcija ivica zasnovana na gradijentima
- Detekcija ivica zasnovana na Laplasijan-u.

3.3 Lokalizacija i detekcija tekstualnih regiona

Ovo poglavlje daje uvid u neke od mogućih sistema za detekciju i prepoznavanje teksta sadržanog u slikama. Razlikuju se tri tipične kategorije pristupa pri razvoju sistema detekcije i prepoznavanja teksta:

1) Metode od dole ka gore, koje razdvajaju slike na regione i grupišu regione karaktera u reči. Ove metode, do nekog stadijuma, mogu izbeći detekciju teksta. Usled poteškoća pri razvoju efikasnog algoritma za segmentaciju slike sa složenijom pozadinom, ove metode nisu robusne za detekciju teksta u slikama načinjenih kamerom/fotoaparatom.

2) Heuristične metode od gore ka dole prvo detektuju regione teksta pomoću heurističnih filtera, a potom primenjuju tehnike od dole ka gore, unutar regiona gde se tekst nalazi. Ove metode su u mogućnosti da obrade složenije slike nego metodi od dole ka gore. Ipak, ručno dizajnirani filtri su empirijski i zbog toga se dobije veći broj lažnih detekcija.

3) Metode od gore ka dole zasnovane na obuci (treniranju) obuhvataju detekciju teksta zasnovanu na filtrima koji su obučeni tzv. „Machine learning“ alatima.

Detekcija teksta definiše se kao zadatak koji lokalizuje nizove karaktera u kompleksnoj pozadini. Postoje mnoge prednosti detekcije teksta pre segmentacije, kao i samog prepoznavanja. Tekst u slikama obično ne pokriva većinu piksela. Lokalizacija teksta nudi informacije o opsegu teksta, što je veoma korisno za segmentaciju tekstualnih regiona iz pozadine. Pozadina unutar lokalizovanog tekstualnog regiona je obično manje složena nego cela slika ukoliko su tekstualni karakteri jasno vidljivi. Time dobra lokalizacija teksta može dovesti do bolje segmentacije i prepoznavanja teksta. Takođe, karakteristike nizova karaktera mogu se iskoristiti pri pronalaženju teksta. Imajući u vidu činjenicu da nizovi karaktera imaju tipične oblike i da su linijski poravnati, može se zaključiti da je lokalizacija niza karaktera lakša i robusnija nego lokalizacija individualnih karaktera.

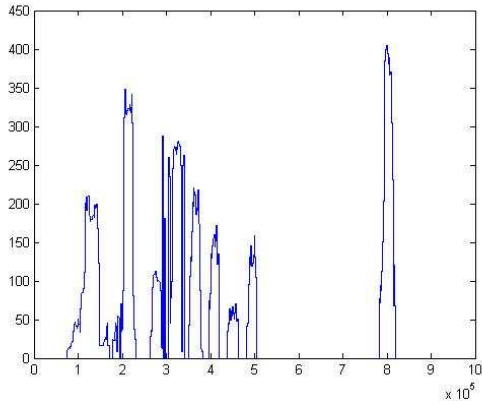
Pozicija ivica se može iskoristiti za računanje širine posmatranog dela teksta. Višestruke lokacije ivica mogu se iskoristiti kako bi se izračunale projekcije, tačke presecanja ili preklapanja i spajanje. Radi ubrzanja algoritma, iskorišćena je činjenica da blokovi sa potencijalnim tekstom sadrže kratke ivice u vertikalnim i horizontalnim orijentacijama, kao i da su pojedine ivice međusobno povezane.

3.4 Računanje linijskih histograma

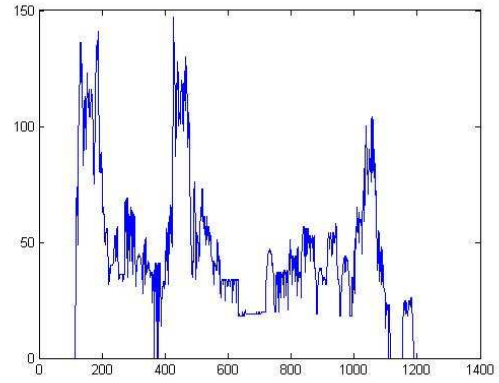
Ovaj pristup se pokazao kao neefikasan u pogledu pronalaska potencijalnih segmenata sa tekstem u slici, stoga je izostavljen iz konačnog algoritma. Linijski histogrami prikazuju horizontalnu i vertikalnu prijekciju. Računanje ovih histograma omogućava eliminaciju delova slike, tačnije linija segmenata, posmatranjem dobijenih projekcija. Pojedine linije sa nebitnim delovima za detekciju teksta uklanjaju se primenom odgovarajućeg praga za svaku projekciju.

Pre pronalaska linija sa tekstem u slici, računa se horizontalna projekcija, a potom se u svakoj liniji teksta pronalaze gornja i donja tačka linije. Analiziraju se i horizontalni i vertikalni projekcioni profili ivične slike kako bi se uklonili nevažni delovi primećeni u određenoj projekciji. To se postiže računanjem praga za svaku projekciju, prvobitnim određivanjem srednje vrednosti za svaki profil. S obzirom da ovi delovi sadrže vrednosti sa velikim kontrastom, očekivano je bilo da se dobiju visoki pikovi u projekcijama.

Vertikalni projekcioni histogram je zbir piksela pozadine u svakom vertikalnom pregledu kolona posmatrane slike. Vertikalni projekcioni histogrami su posebno korisni kada je osnovna linija teksta poznata, i tekst je podeljen pravom linijom koja je ortogonalna na histogram. Analiza projekcionih profila je popularna tehnika za segmentaciju linija. Horizontalni projekcioni profili se koriste iz razloga što je tekst u većini slika pozicioniran horizontalno. Prilikom pravljenja horizontalnog projekcionog profila slike $M \times N$, dobija se niz veličine $M \times 1$ (kolona). Elementi ove kolone su suma vrednosti piksela za svaku vrstu slike. Kao poslednji korak, ekstrahovane linije se izdele na karaktere. Za pronalazak granica između karaktera primenjuje se vrednost praga na praznine između karaktera. Nakon pronalaska lokacija praznina između karaktera, eliminišu se delovi segmenta. Vertikalni projekcioni profil rezultuje sa pozicijama linija teksta, a da bi se detektovale ekstremne pozicije u horizontalnom pravcu, primenjuje se horizontalna projekcija. Pomenuti histogrami se dobijaju učitavanjem novokreiranih datoteka u Matlab simulatoru i posmatraju se elementi koji predstavljaju broj piksela u svakoj liniji ivične slike koji su nebitni, do određene vrednosti praga. Takvi pikseli se odbacuju jer se karakteristike koje su očigledne za očekivano povezane komponente, ne slažu, kao i oni koji prelaze određenu granicu praga, u zavisnosti od slike koja se posmatra, odnosno od oblika histograma posmatrane slike. Na slikama 5. i 6. horizontalna osa predstavlja piksele, dok vertikalna osa predstavlja broj piksela u svakoj liniji. Na slici 7. prikazan je ukupan projekcioni profil dobijen zbirom oba profila.



Slika 5. Histogram horizontalnog projekcionog profila



Slika 6. Histogram vertikalnog projekcionog profila



Slika 7. Ukupan projekcioni profil

4. Programsko rešenje

Implementacija programskog rešenja ostvarena je upotrebom EasyBMP biblioteke, koja služi za učitavanje, zapis i izmenu slika u BMP formatu i realizovana je programskim jezikom C, s tim da je korišćen OCR Tesseract napisan u C++.

Postoje dva toka algoritma, gde je jedan razvijan u cilju pronalaska tačnih koordinata pravougaonih segmenata slike koji uokviruju tekst, dok je drugi tok razvijan s ciljem da se slika što bolje pripremi, odnosno optimizuje za OCR, koji zahteva tačne koordinate blokova sa tekstem (ili cele slike), kako bi na izlazu dobili prepoznate karaktere.

Na početku algoritma je urađena konverzija iz RGB u YUV prostor boja pomoću transformacije obezbeđene pomenutom bibliotekom. Nakon ovoga je odrađeno niskofrekventno filtriranje radi uklanjanja šuma u obrađivanoj slici, a potom slede važni koraci u okviru ovog algoritma, u koje spadaju detekcija ivica i adaptivna binarizacija.

Implementirana metoda detekcije ivica je po ugledu na algoritam iz rada [5]. Takođe je metod adaptivne binarizacije osmišljen na osnovu rada [6] autora Sauvola.

4.1 Funkcija RGB2YUV

```
void RGB2YUV(double *R, double *G, double *B, double *Y, double *U, double *V, int Width, int Height)
```

Funkcija RGB2YUV vrši konverziju u YUV prostor boja pomoću odgovarajuće transformacije, a pri daljoj obradi posmatra se samo luminantna komponenta Y.

4.2 Funkcija Lpfiltering

```
void LPfiltering(double *Yout, double *Y, int Width, int Height)
```

Primenjena je Gauss-ova maska 3x3 radi uklanjanja šuma iz slike, a da se pri tome ne zamute ivice objekata u slici. Ovakvo filtriranje je linearna konvolucija sa Gauss-ovom maskom predstavljenom na slici ispod.

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

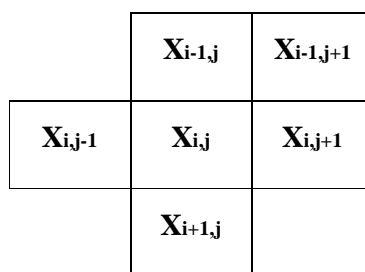
Slika 8. Gaussian maska

Ovim filtriranjem omogućeno je podešavanje količine prostornog usrednjavanja koje se javlja u slici. Filtriranje je neophodno iz razloga što manji, pa čak i izolovani delovi kontura mogu dovesti do nepreciznog uokvirivanja teksta, odnosno njegove netačne detekcije.

4.3 Funkcija EdgeImage

`void EdgeImage(double *Yout, double *Y, int Width, int Height, int Threshold)`

U ovoj funkciji se vrši preobražaj sive slike u ivičnu sliku. Dati algoritam je zasnovan na činjenici da konture karaktera imaju visok kontrast u odnosu na susedne piksele. Time se ovi karakteri registruju ivičnom slikom (vizuelno izdvajaju), kao i pikseli nekih delova slike koji nisu karkteri, a takođe poseduju veći kontrast u odnosu na piksele iz svoje okoline.



Slika 9. Izgled orijentacije susednih piksela u odnosu na posmatrani piksel $X_{i,j}$

U ivičnoj slici su vrednosti svih piksela originalne slike zamenjene najvećom razlikom između njih i susednih piksela (slika 9., kao u radu [7]), izračunatih apsolutnom vrednošću razlike dva piksela u horizontalnom, vertikalnom i dijagonalnom pravcu, prema sledećim uslovima:

$$X_{i,j-1} > X_{i-1,j} \wedge X_{i,j-1} > X_{i-1,j+1} \Rightarrow X_{\max} = X_{i,j-1}$$

$$X_{i,j-1} < X_{i-1,j} \wedge X_{i-1,j} > X_{i-1,j+1} \Rightarrow X_{\max} = X_{i-1,j}$$

, gde je tekući piksel $X_{i,j}$, dok je $X_{i,j-1}$ piksel sa leve strane, $X_{i-1,j}$ piksel sa gornje strane, a $X_{i-1,j+1}$ piksel dijagonalno sa desne gornje strane.

Ovakva slika je dobijena u binarizovanom obliku, postavljanjem vrednosti registrovanih ivica koje su veće od određenog praga na 255 (potencijalni kandidati za tekst), a ostalih delova (pozadine i neželjenih piksela) na 0. Pomenuti prag je određen analizom vrednosti histograma, nakon učitavanja dobijenih vrednosti u Matlab-u. Međutim, globalni prag ograničava performanse detekcije iz razloga što broj očuvanih linija (ivica koje

predstavljaju tekst) prvenstveno zavisi od složenosti pozadine i zbog toga ovaj algoritam služi samo za određivanje ivica.

4.4 Funkcija Labeling

```
void Labeling(double *R, double *G, double *B, double *Y, double *Yin, int Width, int Height)
```

Labeling funkcija prvenstveno je namenjena kao metod za obeležavanje delova u slici gde su pronađene promene u vrednostima piksela, obeležavanjem i bojenjem tih segmenata. Pronalaze se blokovi i njihovi pikseli obeležavaju poređenjem svake vrednosti slike sa pragom, za svaki od piksela. Dalje je razmatrana segmentacija posmatranih blokova. U predloženom algoritmu se, nakon pronalaska osnovnih linija koje sadrže piksele sa vrednostima koje odgovaraju tekstu, dopunjavaju (obeležavaju) susedne linije kako bi se dobili pravilniji blokovi koji obuhvataju delove slike sa tekstem. Naime, za trenutno posmatranu liniju proveravaju se susedni pikseli. Za liniju u istom segmentu postoji četiri slučaja. Linije se dodaju, odnosno praznine se dopunjavaju u slučajevima ako posmatrana linija poseduje piksele koji su već verifikovani da pripadaju tekstualnom sadržaju sa svoje:

- desne i donje strane
- leve i donje strane
- desne i gornje strane
- leve i gornje strane.

4.5 Funkcija Rectangle

```
void Rectangle(int totalLinesW, int Height, int *line)
```

U ovoj funkciji se pronalazi broj blokova sa tekstem u slici, tj. pod-slika od značaja. Kroz sliku se prolazi po dužinama predefinisanih blokova. Za linije koje sa sigurnošću sadrže piksele koji bi mogli biti tekst, prolazi se kroz svaki pronađeni segment sa tekstem.

Poziva se unutar funkcije Labeling, neposredno pre obeležavanja pronađenih blokova, s ciljem da se koordinate svih blokova sa tekstem zapišu.

Zapisane koordinate čuvaju se u datoteci (primer koordinata svih blokova za posmatranu sliku dat je tabelom 1.), ali pre toga se vrši korekcija ovih blokova opisana u sledećem odeljku. Koordinate su predstavljene sledećim promenljivama:

$rect[1][m]$, $rect[0][n]$, predstavljajući početnu koordinatu segmenta i $rect[3][m]$, $rect[2][n]$, predstavljajući poslednju koordinatu segmenta.

rect[0][0]: 60	rect[1][0]: 117	rect[2][0]: 119	rect[3][0]: 200
rect[0][1]: 82	rect[1][1]: 240	rect[2][1]: 114	rect[3][1]: 481
rect[0][2]: 114	rect[1][2]: 120	rect[2][2]: 134	rect[3][2]: 200
rect[0][3]: 154	rect[1][3]: 718	rect[2][3]: 180	rect[3][3]: 880
rect[0][4]: 154	rect[1][4]: 914	rect[2][4]: 179	rect[3][4]: 1120
rect[0][5]: 157	rect[1][5]: 1154	rect[2][5]: 176	rect[3][5]: 1200
rect[0][6]: 206	rect[1][6]: 120	rect[2][6]: 227	rect[3][6]: 280
rect[0][7]: 235	rect[1][7]: 115	rect[2][7]: 268	rect[3][7]: 362
rect[0][8]: 242	rect[1][8]: 400	rect[2][8]: 261	rect[3][8]: 480
rect[0][9]: 243	rect[1][9]: 1000	rect[2][9]: 261	rect[3][9]: 1080
rect[0][10]: 274	rect[1][10]: 120	rect[2][10]: 294	rect[3][10]: 240
rect[0][11]: 276	rect[1][11]: 400	rect[2][11]: 298	rect[3][11]: 525
rect[0][12]: 276	rect[1][12]: 1000	rect[2][12]: 294	rect[3][12]: 1120
rect[0][13]: 309	rect[1][13]: 400	rect[2][13]: 331	rect[3][13]: 560
rect[0][14]: 309	rect[1][14]: 998	rect[2][14]: 327	rect[3][14]: 1120
rect[0][15]: 343	rect[1][15]: 400	rect[2][15]: 361	rect[3][15]: 480
rect[0][16]: 343	rect[1][16]: 1000	rect[2][16]: 360	rect[3][16]: 1080
rect[0][17]: 376	rect[1][17]: 400	rect[2][17]: 394	rect[3][17]: 600
rect[0][18]: 379	rect[1][18]: 1033	rect[2][18]: 394	rect[3][18]: 1080
rect[0][19]: 611	rect[1][19]: 400	rect[2][19]: 640	rect[3][19]: 640
rect[0][20]: 612	rect[1][20]: 793	rect[2][20]: 635	rect[3][20]: 960
rect[0][21]: 617	rect[1][21]: 115	rect[2][21]: 638	rect[3][21]: 200

Tabela 1. Koordinate blokova za predstavljenu sliku

4.6 Dopunjavanje blokova teksta

`void FillHorizontal(int numRect, double *Y, int Width, int *line)`

`void FillVertical(int numRect, double *Y, int Width, int *line)`

Algoritam po kome se dopunjavaju praznine „uokvirujućih“ blokova (oni koji obuhvataju tekst) predstavlja prolazak kroz sve blokove (njihove vrste i kolone), pri čemu se preskaču vrste/kolone za horizontalno/vertikalno dopunjavanje. Kada se nađe na piksel crne boje, povećava se broj linija u datoj vrsti/koloni i time se željeni blokovi proširuju.

Ovakvo dopunjavanje se vrši u horizontalnim i vertikalnim pravcima, stoga je razdvojeno u dve zasebne funkcije. Podešavanje broja linija po vrsti i koloni, koji je neophodan kako bi se obuhvatio ceo blok, je podešen za praktično sve tipične slučajeve reči - tekst sa slovima koja su nešto duža ili šira, odnosno proširuju se na dole ili gore, tj. levo ili desno. To je postignuto odgovarajućim konstantnim horizontalnim ili vertikalnim korakom

kojim se dodaju linije na ivicama blokova.

Binarizovana slika se u sledećem koraku priprema za prepoznavanje teksta od strane OCR-a, što je postignuto efikasnom binarizacionom metodom pomoću odgovarajućeg praga.

4.7 Funkcija AdaptiveThreshold

```
void AdaptiveThresholding(int numRect, double *Y, double *Yout, double Width)
```

Adaptivna binarizacija (odabir praga) je važan korak u pripremi slike za OCR. Prag se u većini slučajeva kod adaptivne binarizacije određuje lokalno, piksel po piksel ili region po region. Prema tome, svaki region može imati odgovarajući prag dobijen na osnovu izbora kandidata praga za datu oblast. Prag varira, prolazeći kroz sliku, računanjem male okoline svakog piksela.

Primenjen je Sauvola adaptivni prag $t(x, y)$, koji koristi srednje vrednosti $m(x, y)$ i standardnu devijaciju $s(x, y)$ inteziteta piksela, što je prikazano jednačinom:

$$t(x, y) = m(x, y) \cdot \left[1 + k \cdot \left(\frac{s(x, y)}{R} - 1 \right) \right]$$

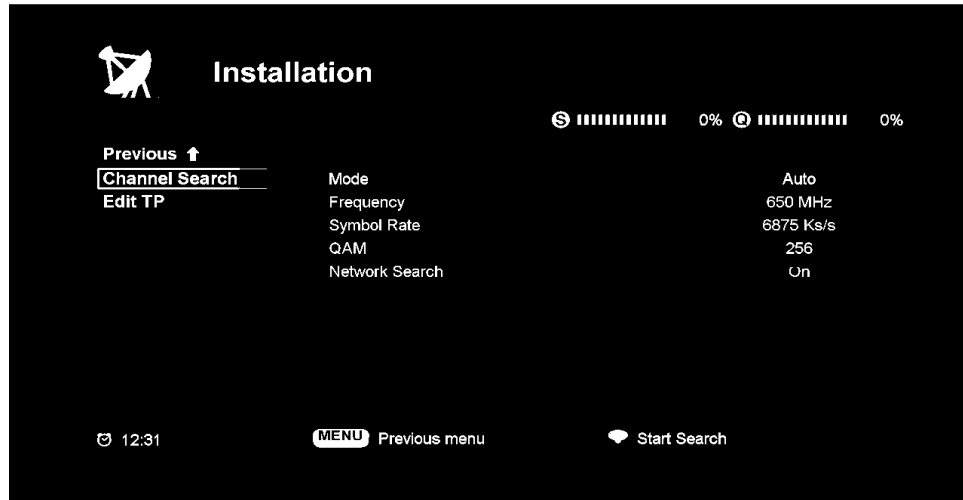
, gde je R maksimalna vrednost standardne devijacije (ili dinamički opseg standardne devijacije $R=128$ za sivu sliku), a k je parametar koji može imati pozitivne vrednosti u opsegu $[0.2, 0.5]$. Ako se uzme u obzir da je intezitet piksela sive slike na lokaciji (x, y) predstavljen sa $g(x, y) \in [0, 255]$, prag $t(x, y)$ se za svaki piksel računa tako da važi:

$$o(x, y) = \begin{cases} 0, & \text{ako je } g(x, y) \leq t(x, y) \\ 255, & \text{inače} \end{cases}$$

Standardna devijacija $s(x, y)$ i srednja vrednost $m(x, y)$ prilagođavaju vrednost praga $t(x, y)$ na osnovu kontrasta lokalnih suseda posmatranog piksela. Kada je kontrast u nekom delu slike veći, standardna devijacija je približno jednaka njenoj maksimalnoj vrednosti, što rezultuje time da je prag približno jednak srednjoj vrednosti. Kada je kontrast između posmatranog piksela i njegovih suseda nizak, prag pada ispod srednje vrednosti, uspešno uklanjajući relativno tamne regione pozadine. Parametar k kontroliše vrednosti praga lokalnog prozora, tako da se sa povećanjem vrednosti parametra k smanjuje prag lokalne srednje vrednosti. Usvojeno je da Sauvola binarizacioni algoritam uzima za k vrednost 0.2 iz razloga što je pomoću ove vrednosti dobijen kompletan sadržaj slike, a veličina prozora obrade je određena veličinom segmenta.

Sauvola algoritam sa adaptivnim pragom je široko primenjen algoritam za binarizaciju i najbolje performanse binarizacije su nad dokumentima, ali dobijena slika na izlazu je

relativno u celosti sačuvana. Konačni rezultat ovog algoritma dat je na slici 10.



Slika 10. Dobijena slika nakon primene adaptivnog praga

4.8 Funkcija CalculateLineHistogram

```
void CalculateLineHistogram(double *Y, double *TotalEdges, int Width, int Height)
```

Ovaj pristup nije bio zadovoljavajući, pa je time izostavljen iz konačnog algoritma, a princip je opisan samo ukratko.

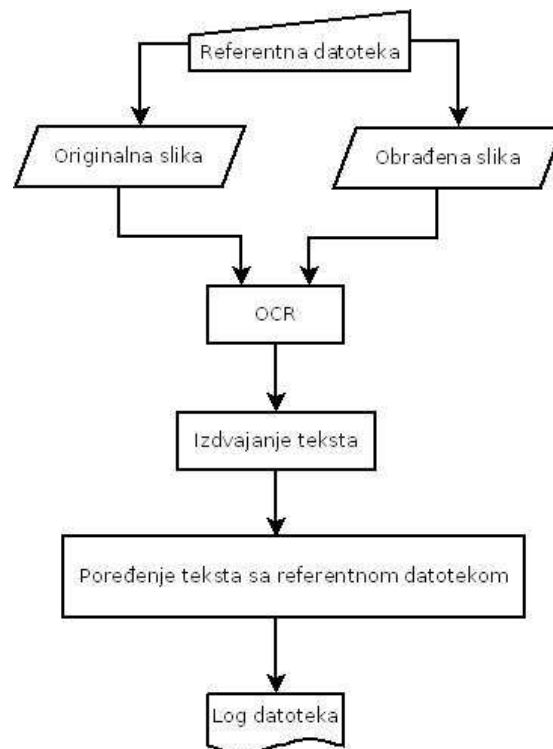
Nakon formiranog niza koji predstavlja horizontalnu projekciju, računa se srednja vrednost horizontalne projekcije, kao i prag na osnovu kojeg se postavljaju kriterijumi za potiskivanje nevažnih vrednosti, odnosno sve vrednosti piksela, manje od ovog praga i veće od praga pomnoženog sa odgovarajućim korakom, se uništavaju. Slično, za vertikalnu projekciju zanemaruju se one vrednosti piksela koje su veće od izračunatog praga i manje od praga podeljenog sa datim korakom. Ideja horizontalnog i vertikalnog projekcionog profila je dobijena iz rada [5].

Sledeći korak je upis prethodno dobijenih ivičnih slika u nove nizove koji predstavljaju horizontalne i vertikalne ivice u slici. Dobijena ukupna projekcija je upisana u datoteku, a pomoću Matlab simulatora su dobijeni histogrami za obe projekcije.

5. Rezultati

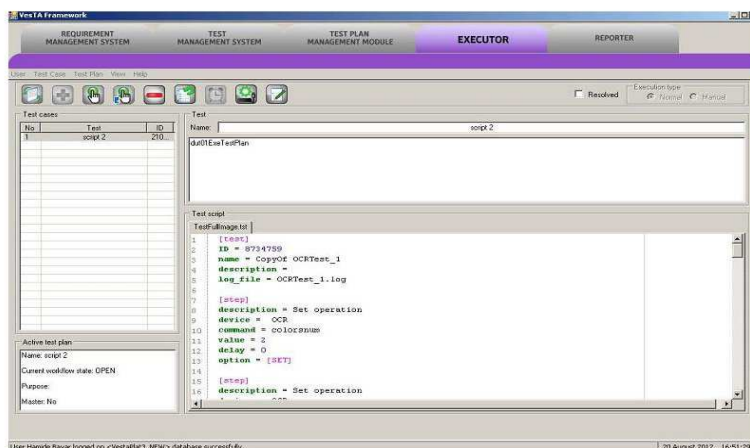
Pristup koji je predložen ovde daje samo još jednu od mnogih namena OCR-u. Pred-obrađena slika je fokusirana na prilagođavanje slike ovom popularnom sistemu kako bi imali prepoznatljive karaktere, a potom i dobijanje koordinata sadržaja meni-ja na svakom delu slike sa ekrana. Slike nad kojima je algoritam ispitivan su preuzete od TV meni-ja HD rezolucije.

Verifikacija teksta se vrši proverom da li karakteri imaju tipične karakteristike tekstualnog fonta, izbacivanjem onih delova slike u kojima su visina i širina datih objekata veće od određenih predefinisanih vrednosti. To je urađeno iz razloga što je OCR sistem napravljen tako da ne prepozna tekst čiji je font mali, ali i zato što pojedini detektovani objekti nisu zapravo tekst, već linije, ikonice ili slično. Predložen pristup izdvajanja teksta je u velikoj meri efikasan, sa višom tačnošću prepoznavanja karaktera nego u originalnoj slici, što se vidi iz priloženih primera. Krajnji cilj je da se ovakav metod primeni nad prihvaćenim i zapisanim slikama sa sadržajem TV ekrana, kao deo BBT sistema za automatsko ispitivanje i funkcionalnu verifikaciju DTV uređaja.



Slika 11. Poređenje dobijenih rezultata

Predstavljen sistem prikazan na Slici 11. služi kao verifikacija funkcionalnog izvršavanja TV uređaja, npr. preuzimanjem i obradom opcija u okviru meni-ja sa ekrana kako bi se verifikovala funkcionalnost televizora, tj. da li je otvoren željeni meni. Slike se dobijaju putem „grabber“-a, programske podrške koja preuzima izgled ekrana u datom momentu. OCR ima ulogu da pročita tekst u okviru obrađivane slike. Samo ispitivanje je odrađeno kontrolnom aplikacijom (eng. *Executor*, Slika 12.) i ispitnim planom koji se izvršava u okviru ovog alata. Pomoću njega su omogućeni kontrola, razvoj i izvršenje BBT ispitnih slučajeva. Ukoliko se napiše više ispitnih slučajeva u jednom ispitnom planu, moguće je izvršiti svaki ispitni plan pojedinačno, a potom se ispravnost ispitivanja proverava na osnovu očekivanih rezultata, odnosno referentnih datoteka. BBT skripte sadrže niz komandi na osnovu kojih se izvršavaju potrebne transformacije i prosleđuju koordinate blokova, tj. željenih segmenata slike.

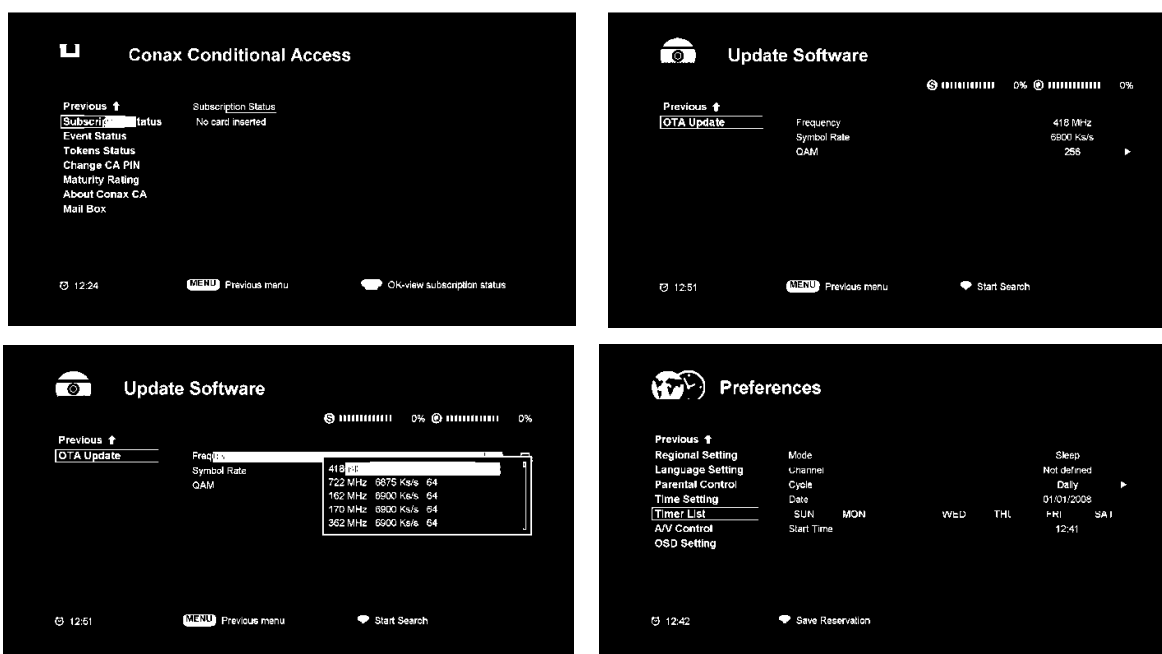


Slika 12. Aplikacija za ispitivanja

Dobijeni rezultati se smeštaju u log datoteke, nakon izvršenih ispitnih planova. Jedna log datoteka sadrži karaktere pronađene od strane OCR-a, dok se u drugoj, tzv. DIFF (od eng. *Difference*) datoteci dobijaju rezultati tačnosti detekcije. U ovoj datoteci se ispisuju rezultati predstavljeni sa dve različite metrike: LCS (eng. *Longest Common Subsequence*) koja dozvoljava dodavanje i brisanje, i Levenshtein-ovom distancom (minimalan broj promena između dve sekvence), a dobijen je i procenat poklapanja rezultata OCR-a sa očekivanim rezultatom. Rezultati dobijeni za ispitivane slike dati su tabelom 2.

Ime slike	Originalna slika		Obradena slika	
	Levenshtein-ova distanca	Poklapanje karaktera [%]	Levenshtein-ova distanca	Poklapanje karaktera [%]
DUT_0_6	77	55	28	91
DUT_0_25	47	81	37	93
DUT_0_26	77	80	65	86
DUT_0_36	86	72	66	80
DUT_0_60	73	72	53	90

Tabela 2. Poređenje tačnosti rezultata



Slika 13. Slike DUT_0_6, DUT_0_25, DUT_0_26, DUT_0_36, redom

Zaključak

U ovom radu je predložen metod za segmentaciju i detekciju teksta u DTV slici. Ovaj postupak je važan prilikom automatskog ispitivanja funkcionalnosti DTV prijemnika. U toku tog postupka je neophodno ispitivanje sistema TV meni-ja i pri tome je potrebna segmentacija teksta i prepoznavanje karaktera koji sačinjavaju tekst. Na početku pred-obrade date slike, isprobane su različite maske s ciljem da se dobije što jasnija slika, i ispostavilo se da od svih filtera, s namenom uklanjanja šuma, najbolje performanse daje Gauss-ov. Pre detektovanja ivica u slici ne može se primeniti više maski. Ali, sasvim je dovoljno da se primeni jedna od maski koja će ukloniti samo nečistoće, a ostaviti ivice netaknute.

Realizovani algoritam s ciljem pripreme slike za OCR, obuhvata korake NF filtriranja, pronalaska ivica u slici i obeležavanja pronađenih segmenata gde se nalaze kandidati za tekst, kao i korekcije lokalizovanih segmenata sa tekstom. Na kraju je primenjen Sauvola adaptivni metod lokalne binarizacije, čime je dobijena slika pripremljena za OCR, a potom su upoređene tačnosti prepoznavanja karaktera između obrađene i originalne slike. Tačnost prepoznatih karaktera nad ispitivanim obrađenim slikama je veća za prosečno 16 % u odnosu na tačnost kod originalnih slika i time su postignuta zadovoljavajuća poboljšanja u performansama OCR-a.

Bilo je neophodno primeniti lokalni, odnosno adaptivni prag iz razloga što je odrađena binarizacija po segmentima daleko preciznija od metode sa globalnim pragom, koji je ranije primenjen kako bi se prikazale ivice. Što se tiče OCR-a i njegovih performansi, detekcija je prihvatljiva i za manje fontove slova, kao i za drugačije tipove, s obzirom da se čak i slovo, u potpunosti izdvojeno od ostatka teksta, detektuje.

Moguće je primeniti neku drugu kombinaciju metoda/algoritama koji bi dali još bolje rezultate, tako da je predstavljen pristup samo jedna od brojnih mogućnosti realizacije algoritma za pripremu slika namenjenih za prepoznavanje teksta u slikama.

Slike koje imaju mali kontrast mogu se transformisati u slike sa većim kontrastom širenjem histograma na ceo opseg, odnosno remapiranjem vrednosti piksela sive slike. Ovakvo širenje kontrasta se često naziva podešavanje dinamičkog opsega (eng. *Dynamic Range Adjustment - DRA*). Međutim, pojačavanjem kontrasta u delovima slike gde je kontrast manji, moguće je da se pojave neželjeni artefakti.

Druga opcija je povećati kontrast u delovima gde je veća jačina ili gustina ivica primenom ECP (eng. *Edge Clustering Power*) ili ESS (eng. *Edge Strength Smoothing*) operatora, iako bi lokalni prag trebao da zadrži tekst koji ima nizak kontrast u odnosu na pozadinu [10]. ESS operator koristi se za ujednačavanje ivica teksta, dok se ECP operator koristi za njihovo pojačavanje.

Detektovani blokovi sa tekstom koji imaju mali kontrast, i iz tog razloga se nakon binarizacije ne vidi tekst u tim delovima slike, podjednako su bitni kao i sav ostali tekst. Moguće je povećati razliku inteziteta između pozadine i objekata, pa je zbog toga potrebno odraditi, upravo na takvim kritičnim delovima slike, adaptivnu ekvilizaciju histograma (eng. *Histogram Equilization - HE*). Takav segment slike se posmatra kao pod-slika, pa se pri ovakvoj obradi ona posmatra kao jedna slika u celosti, te se nad njom vrši proširivanje histograma i mora se izračunati histogram posebno za ovakve slučajeve. Međutim, HE je globalni metod, iako veoma efikasan za povećanje inteziteta detalja slike.

Još jedna ideja vezana za preciznije prepoznavanje karaktera po blokovima je određivanje koordinata svih segmenata slike i prepoznavanje teksta sa većom tačnošću. Međutim, cilj pripreme slike i izdvajanja teksta iz slika kod BBT-a je verifikacija takvog sadržaja pri ispitivanju uređaja i time je bolji pristup pamćenje svih preuzetih slika, a potom njihovo poređenje sa trenutno preuzetim slikama metodom poređenja blokova (eng. *block matching*).

Literatura

- [1] David J. Crandall, “Extraction of unconstrained caption text from general-purpose video”
- [2] HP Labs & Google: Tesseract-OCR
<http://code.google.com/p/tesseract-ocr/>
- [3] Datong Chen, “Text detection and recognition in images and video sequences”
- [4] Raman Maini, Dr. Himanshu Aggarwal, “Study and Comparison of Various Image Edge Detection Techniques”, International Journal of Image Processing (IJIP), Volume (3) : Issue (1)
- [5] Julinda Gllavata, Ralph Ewerth and Bernd Freisleben, “A Robust algorithm for Text detection in images”, Proceedings of the 3rd international symposium on Image and Signal Processing and Analysis, 2003.
- [6] J. Sauvola and M. Pietikainen, “Adaptive document image binarization,” Pattern Recognition 33(2), 2000., pp. 225–236
- [7] J. Sushma and M. Padmaja, “Text Detection in Color Images”, 2009.
- [8] Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti, “Text Line Segmentation Based on Morphology and Histogram Projection”, Center of Informatics, Federal University of Pernambuco, 10th International Conference on Document Analysis and Recognition, 2009.
- [9] David J. Crandall, “Extraction of unconstrained caption text from general-purpose video”
- [10] Hao-Chao Chang, Jia-Ching Cheng, “Automatic Text Extraction in Video using Artificial Neural Network”, Thesis for Master of Science, 2005.