UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET
KATEDRA ZA RAČUNARSTVO

Aplikacija za separaciju karaktera iz skeniranih tekstualnih dokumenata i generisanje baze karaktera različitih formata

Master rad - studijsko-istraživački rad

Smer: Računarstvo i informatika
Modul: Softversko inženjerstvo

Kandidat:
Marko Đokić 1022

Mentor:
Prof. dr. Vladan Vučković

Niš, 2024. God.

Sadržaj

1. Projektni zadatak

Aplikacija je tool odnosno kombinacija automatskih i polu-automatsih procedura. Aplikacija se sastoji iz narednih segmenata:

1.1. Loader skeniranog dokumenta (.jpg ili .bmp):

Treba da radi sa jednom ili nekoliko slika dobijenih skeniranjem dokumenta pod istim uslovima (npr. iskucanih istom pisacom masinom).

1.2. Filteri i binarizacija

Upotrebom filtera sa ručnim podešavanjem klizača, kao u photoshop-u, uklanjaju se šumovi i ostale "smetnje" na celoj slici. Takođe, uklanjaju se i boje tako da se na kraju dobije binarizovana slika (jedan bajt po pikselu, 0 pozadina, 255 element karaktera).

1.3. Separacija karaktera

Nad tako obrađenom slikom (slikama) primenjuje se neki od algoritama sa separaciju karaktera (slova, brojke, interpunkcijski znaci). Parametri za separaciju (ako ih ima) se takođe podešavaju ručno i vizuelno se prati proces - do maksimalno dobrog rezultata.

1.4. Formiranje baze karaktera

Od tako izdvojenih karaktera formira se baza karaktera. Mozemo odrediti da baza uvek bude standardna, prema ASCII tabeli i to od koda 33 do 126 (ukupno 94 karaktera). Ova oblast ukljucuje sve karaktere koji se mogu pojaviti u tekstu. U slucaju da neki karakter nedostaje u bazi (nije separisan iz teksta) u tabeli se ubacuje kod 32 (BLANKO).

1.5. Editovanje baze karaktera:

Listanje baze i editovanje pojedinih karaktera mišem (uključivanje ili isključivanje piksela) u bazi. Karakteri u bazi imaju različite x i y dimezije, o tome treba voditi računa u editoru.

1.6. Normalizacija:

Finalni korak, normalizacija sređene baze i snimanje na disk u binarnom formatu tako da je svaki karakter sveden na identičnu matricu NxM (visina x sirina). Treba da postoje dva fajla:

1. “normbaza.txt” Tekstualni fajl, sadrži samo jedan red (string dužine 94 karaktera) u kome se nalaze definisani karakteri. Oni koji nedostaju imaju na odgovarajućem mestu blanko znak. Npr: !"# 01234567890:;<=>? ABCDEF.....xyz{ }. Za svaki od registrovanih karaktera u ovom stringu ide binarna definicija u jedinsvenom binarnom fajlu

2. “normbaza.bsa”. Jedan bajt po jednom pikselu, bajtovi se ređaju kao na displeju odozgo levo, prema dole desno, 24x16=384 bajta po svakom karakteru, 0 je pozadina, 255 ispunjen karakter. Ako karakter nije definisan u ovoj bazi na odgovarajućim pozicijama su sve nule.

Izlaz osim u binarnom formatu se moze i ucitati posle generisanja u neki hex viewer, treba dopuniti i ASCII bazom.

2. Uvod

Digitalna obrada slike, kao zasebna disciplina, započela je svoj razvoj sredinom šezdesetih godina dvadesetog veka. Prvobitno pokrenuta potrebama svemirskih istraživanja, digitalna obrada slike omogućila je značajno poboljšanje kvaliteta slika dobijenih sa svemirskih sondi i satelita, uprkos tadašnjim ograničenjima u računskoj snazi i kapacitetima elektronike. Tokom godina, disciplina je doživela veliki napredak, od prvih vojnih i istraživačkih primena do široke komercijalizacije i dostupnosti običnim korisnicima.

Sa razvojem mikroelektronike i pojavom mikroprocesorskih komponenti, cena računarske opreme značajno je pala dok su njene mogućnosti dramatično porasle. Digitalna obrada slike postala je dostupna širem krugu korisnika, uključujući industriju, televiziju, telekomunikacije i robotiku. Oprema za obradu slike postala je periferija personalnih računara, omogućavajući njenu upotrebu u različitim komercijalnim primenama. Pojava digitalnih fotoaparata i napredak personalnih računara učinili su digitalnu obradu slike dostupnom široj publici, uključujući obične korisnike bez mnogo iskustva. Razvijeni su mnogi specijalizovani programi za obradu slike koji su često besplatni ili vrlo pristupačni.

Tema ovog rada je razvoj aplikacije za generisanje fontova, koja koristi OCR tehnologiju za prepoznavanje karaktera sa slika i kreiranje baze karaktera. Ova aplikacija omogućava korisnicima da snime ili učitaju slike sa tekstom, prepoznaju karaktere pomoću OCR sistema i automatski generišu digitalne fontove. Takva aplikacija može imati široku primenu u dizajnu, tipografiji, arhiviranju dokumenata i drugim oblastima gde je personalizacija fontova od značaja.

Aplikacija će biti razvijena korišćenjem naprednih tehnika obrade slike i mašinskog učenja, kako bi se obezbedila visoka tačnost prepoznavanja karaktera i generisanja fontova. Pored toga, aplikacija će imati intuitivno korisničko uputstvo koje će omogućiti jednostavnu upotrebu i prilagođavanje različitim potrebama korisnika.

3. Optičko prepoznavanje karaktera

Optičko prepoznavanje karaktera ili optički čitač karaktera (OCR) je elektronska ili mehanička konverzija slika kucanog, rukom pisanog ili štampanog teksta u mašinski kodirani tekst, bilo da je reč o skeniranom dokumentu, fotografiji dokumenta, scenskoj fotografiji (na primer, tekst na znakovima i bilbordima na pejzažnoj fotografiji) ili o tekstu titla prekrivenog preko slike (na primer: sa televizijskog prenosa).

Široko se koristi kao oblik unosa podataka sa štampanih papirnih zapisa – bilo da su to pasoški dokumenti, računi, bankovni izvodi, kompjuterski računi, poslovne kartice, pošta, štampani podaci ili bilo koja odgovarajuća dokumentacija – to je uobičajena metoda digitalizacije štampanih tekstova kako bi se mogli elektronski uređivati, pretraživati, kompaktno skladištiti, prikazivati online i koristiti u mašinskim procesima kao što su kognitivno računarstvo, mašinsko prevođenje, (izvađen) tekst-u-govor, ključni podaci i rudarenje teksta. OCR je oblast istraživanja u prepoznavanju šablona, veštačkoj inteligenciji i kompjuterskoj viziji.

Rane verzije su morale biti trenirane sa slikama svakog karaktera i radile su sa jednim fontom u isto vreme. Napredni sistemi sposobni da postignu visok stepen tačnosti za većinu fontova sada su uobičajeni, uz podršku za različite formate ulaznih slika. Neki sistemi su sposobni da reprodukuju formatirani izlaz koji blisko odgovara originalnoj stranici uključujući slike, kolone i druge ne-tekstualne komponente.

4. Tehnologije i biblioteke

Tesseract

Tesseract je open-source OCR (Optical Character Recognition) mehanizam koji se koristi za prepoznavanje i konverziju teksta sa slika u mašinski kodirani tekst.

Tesseract je sposoban za prepoznavanje teksta na mnogim jezicima, a takođe podržava i prepoznavanje različitih fontova i stilova teksta. Jedna od njegovih glavnih prednosti je sposobnost prepoznavanja višerednog i višekolonskog teksta, kao i tekstova koji sadrže složene strukture kao što su tabele i kolone.

Pored toga, Tesseract može da radi sa različitim formatima ulaznih slika, uključujući TIFF, PNG, JPEG, i druge. Njegova modularna arhitektura omogućava lako prilagođavanje i integraciju u različite aplikacije i sisteme.

Tesseract je moćan i fleksibilan open-source OCR (Optical Character Recognition) mehanizam koji se koristi za prepoznavanje i konverziju teksta sa slika u mašinski kodirani tekst. Razvijen je od strane Hewlett-Packarda tokom 1980-ih i ranih 1990-ih, a Google ga je preuzeo i održava ga od 2006. godine. Tesseract je postao jedan od najpopularnijih OCR alata zbog svoje preciznosti i sposobnosti prepoznavanja velikog broja jezika.

Glavne Karakteristike Tesseract-a

1. Podrška za Više Jezika: Tesseract prepoznaje više od 100 jezika, uključujući ideografske jezike i jezike koji se pišu zdesna nalevo. Korisnici mogu dodavati nove jezike i trenirati sistem za specifične potrebe.

2. Dva OCR Motora: Tesseract koristi dva glavna model za prepoznavanje teksta: LSTM (Long Short-Term Memory) Model: Koristi napredne tehnike dubokog učenja za prepoznavanje teksta. Legacy Model: Tradicionalni motor koji prepoznaje obrasce karaktera.

3. Različiti Izlazni Formati: Tesseract može generisati različite formate izlaznog teksta, kao što su plain text, hOCR (HTML for OCR), PDF, i TSV. Ovo omogućava fleksibilnu upotrebu u različitim aplikacijama i sistemima.

4. Podrška za Različite Formate Slika: Tesseract može raditi sa različitim formatima ulaznih slika, uključujući TIFF, PNG, JPEG, BMP, i druge.

5. Integracija i Prilagodljivost: Tesseract se može lako integrisati u aplikacije korišćenjem različitih programskih jezika kao što su Python, C++, i Java.

6. Njegova modularna arhitektura omogućava prilagođavanje specifičnim potrebama korisnika.

Emgu.CV

Emgu.CV je cross-platformska .NET biblioteka (wrapper) za OpenCV, omogućavajući jednostavnu integraciju OpenCV funkcionalnosti u .NET aplikacije koristeći jezike kao što su C# i VB.NET. Ova biblioteka nudi bogat skup funkcija za obradu slike, od kojih su mnoge ključne za prethodnu obradu slika pre nego što se proslede OCR sistemima za prepoznavanje teksta. Jedna od najčešće korišćenih funkcija u okviru ove biblioteke je binarizacija i thresholding, koje se koriste za poboljšanje kvaliteta slike i povećanje tačnosti prepoznavanja karaktera.

Binarizacija i Thresholding u Emgu.CV

Binarizacija i thresholding su osnovne tehnike u obradi slike koje se koriste za konverziju sive slike u binarnu sliku, pri čemu se pikseli dele na bele i crne vrednosti na osnovu određenog praga. Ove tehnike su posebno korisne u OCR aplikacijama, jer pomažu u uklanjanju šuma i povećavaju kontrast između teksta i pozadine, što olakšava prepoznavanje karaktera.

Emgu.CV pruža nekoliko metoda za binarizaciju i thresholding, uključujući globalni thresholding, adaptivni thresholding, i Otsu-ov metod. U nastavku su neke od primena ovih metoda.

Globalni Thresholding: Globalno thresholding primenjuje jedan prag za celu sliku. Svi pikseli ispod praga se postavljaju na crno, dok se svi pikseli iznad praga postavljaju na belo.

```
Mat src = CvInvoke.Imread("input_image.png", ImreadModes.Grayscale);
Mat dest = new Mat();
double threshValue = 128.0;
CvInvoke.Threshold(src, dest, threshValue, 255, ThresholdType.Binary);
```

Slika 1. Globalni Thresholding

Adaptivni Thresholding: Adaptivno thresholding koristi različite pragove za različite delove slike, što je korisno kada slika ima promenljivu osvetljenost.

```
Mat src = CvInvoke.Imread("input_image.png", ImreadModes.Grayscale);
Mat dest = new Mat();
CvInvoke.AdaptiveThreshold(src, dest, 255, AdaptiveThresholdType.MeanC, ThresholdType.Binary, 11, 2);
```

Slika 2. Adaptivni Thresholding

Otsu-ov Metod: Otsu-ov metod automatski izračunava optimalni prag za binarizaciju slike na osnovu histograma piksel vrednosti.

```
Mat src = CvInvoke.Imread("input_image.png", ImreadModes.Grayscale);
Mat dest = new Mat();
CvInvoke.Threshold(src, dest, 0, 255, ThresholdType.Binary | ThresholdType.Otsu);
```

Slika 3. Otsu-ov metod

U okviru ove aplikacije, Emgu.CV se koristi za prethodnu obradu ulaznih slika kako bi se poboljšala tačnost OCR prepoznavanja. Primena binarizacije i thresholding-a omogućava jasniji kontrast između karaktera i pozadine, što rezultira boljim prepoznavanjem karaktera i tačnijim generisanjem fontova.

Windows Forme

Windows Forme (Windows Forms) predstavljaju deo .NET Framework-a koji omogućava razvoj bogatih grafičkih korisničkih interfejsa (GUI) za Windows aplikacije. Kao razvojna platforma, Windows Forme nude niz prednosti koje ih čine odličnim izborom za izradu desktop aplikacija, posebno u kontekstu aplikacija koje zahtevaju složene interakcije korisnika, kao što je ova aplikacija.

Prednosti Korišćenja Windows Formi

1. Jednostavnost i Produktivnost: Windows Forme omogućavaju brzo i jednostavno kreiranje GUI elemenata pomoću drag-and-drop vizuelnog dizajnera u Visual Studio okruženju. Ovaj pristup omogućava programerima da brzo razviju i iteriraju dizajn korisničkog interfejsa, povećavajući produktivnost i skraćujući vreme razvoja.
2. Bogata Biblioteka Kontrola: Windows Forme dolaze sa bogatim skupom ugrađenih kontrola, kao što su dugmad, tekstualna polja, liste, tabele, meniji i mnoge druge. Ove kontrole mogu se lako prilagoditi i kombinovati kako bi se kreirao funkcionalan i atraktivan korisnički interfejs.
3. Dobra Podrška za .NET Framework: Kao deo .NET Framework-a, Windows Forme imaju pristup svim pogodnostima ovog okruženja, uključujući robusnu biblioteku klasa, podršku za više jezika (C#, VB.NET, F#), i napredne funkcionalnosti kao što su mrežna komunikacija, rad sa bazama podataka i obrada slike.
4. Kompatibilnost i Stabilnost: Windows Forme su već dugo prisutne i dokazale su se kao stabilna platforma za razvoj aplikacija. Kompatibilnost sa starijim verzijama Windows operativnog sistema čini ih pouzdanim izborom za širok spektar korisnika.
5. Vizuelni Dizajner: Visual Studio nudi napredni vizuelni dizajner za Windows Forme, omogućavajući programerima da intuitivno rasporede i konfigurišu

kontrole na formi. Ovo čini razvoj interfejsa brzim i jednostavnim, bez potrebe za ručnim pisanjem velikih količina koda.

Izbor Windows Formi za razvoj aplikacije omogućava jednostavno kreiranje bogatih i interaktivnih korisničkih interfejsa, uz iskorišćavanje prednosti .NET Framework-a. Kroz korišćenje Windows Formi, aplikacija postaje pristupačna širokom krugu korisnika, pružajući intuitivne alate za prepoznavanje karaktera i generisanje personalizovanih fontova.

Verzije i pokretanje

Kompletna aplikacija je realizovana pomoću .NET okruženja odnosno framework-a. Okruženje je odabrano zato što poseduje veliku podršku za obradu slike i takođe tip aplikacije Windows form aplikacija ima odličnu pordšku za prikaz i rad sa slikam. Aplikacija ima četiri osnovne Windows forme za prikaz na kojima se nalaze sve ostale kontrole. Vezija okruženja korišćenog u aplikaciji je 4.7.2.

```
<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
```

Slika 4. Verzija .NET Okruženja

Glavne biblioteke koje su bile neophodne za realizaciju projekta su biblioteka Tesseract i EmguCV. Tesseract biblioteka se koristi kao OCR za prepoznavanje teksta, karaktera, pozicije karaktera itd. EmguCV je biblioteka koja u osnovi koristi OpenCV i koristi se za različite obrade slika u aplikaciji, kao što je na primer binarizacija slike.

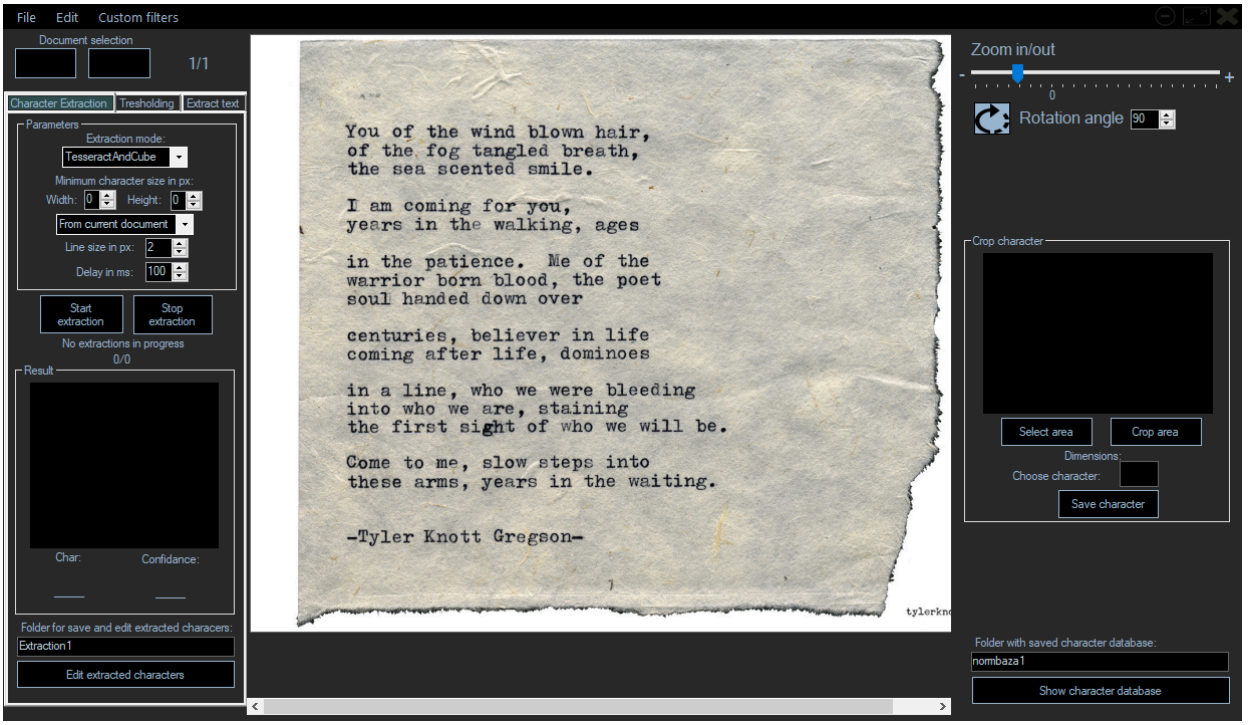
```
<package id="EMGU.CV" version="4.1.1.3497" targetFramework="net472" />
<package id="Tesseract" version="3.3.0" targetFramework="net472" />
```

Slika 5. Verzije glavnih biblioteka

Za pokretanje aplikacije i za dalji njen razvoj dovoljno bi bilo izbildovati je u Visual Studiju i pokrenuti. Komande u konzoli za bildovanje i pokretanje bile bi **`dotnet build`** i **`dotnet run`**, ove komande bi trebale rešiti sve i dodati sve pakete. U slučaju da dođe do neke greške moguće je dodati pakete sa nekom od narednih komandi **`nuget restore`** i **`dotnet restore`**. Ukoliko korisnik želi da koristi aplikaciju porebno je samo pokrenuti instalaciju i pratiti korake instalacije i aplikacija će biti spremna za upotrebu. Testirana je na operativnm sistemima Windows 7 i 10.

5. Korisničko uputstvo

Kao što se može videti na slici 6. glavna forma se sastoji od velikog broja delova i kontrola. Iz glavne fome se stiže do svih ostalih formi. U nastavku ce redom biti opisane sve mogućnosti aplikacije idući redom od početnih zadataka i zahteva aplikacije.

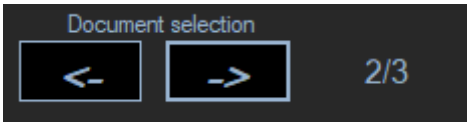


Slika 6. Izgled početne forme aplikacije sa učitanom slikom

5.1. Loader skeniranog dokumenta (.jpg ili .bmp)

Učitavanje slike koja je u .jpg ili .bmp formatu se obavlja u menu-strip kontroli odabradi opciju File -> Get images, nakon toga otvoriće se Dialog pomoću koga je moguće očitati jednu ili više slika.

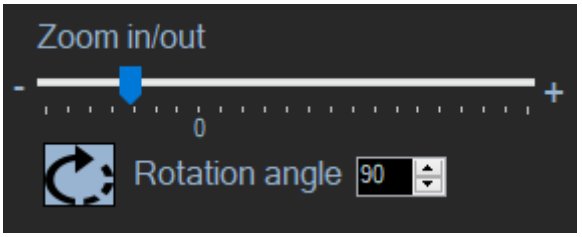
Nakon što su jedna ili više slika učitane, u glavnoj formi u kontroli na sredini aplikacije prikazaće se prva učitana slika, ostale slike je moguće listati pomoću dugmića, na kojima su nacrtane strelice, u gornjem levom uglu (slika 7). Prva brojka ispod ovih dugmića predstavlja koji je dokument trenutno prikazan, a druga predstavlja ukupan broj dokumenata.



Slika 7. Izbor dokumenata

U gornjem desnom uglu postoji track-bar kontrola koja služi za zumiranje i odzumiranje dokumenta koji je trenutno prikazan (slika 8.). Takođe ovom kontrolom je moguće upravljati i nakon što se klikne na površinu očitane slike, dovoljno je samo okretati

točkić tj. scroll taster na mišu i kontrola će pratiti točkić. Ispod ove kontrole postoji kontrola dugme sa ikonicom u obliku strelice, ova kontrola služi za rotiranje slike pod određenim uglom. Desno od ovog dugmeta je kontrola u kojoj se nalazi određena numerička vrednost koja predstavlja ugao pod kojim će se trenutno izabrana slika rotirati.

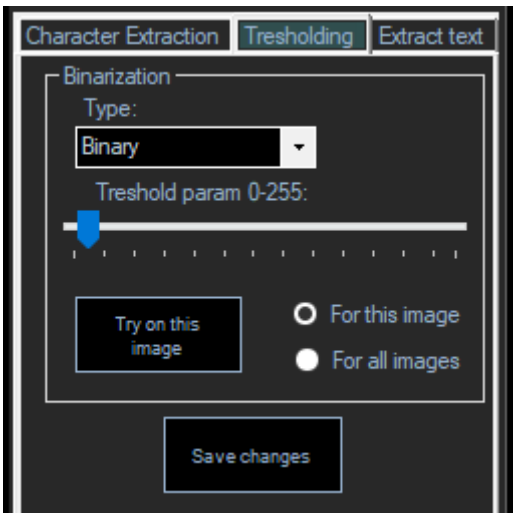


Slika 8. Track bar

5.2. Filteri i binarizacija

Deo unutar drugog taba u tab kontoli koji se nalazi levo od prikazane centralne slike, služi za binarizaciju/tresholding slike, prikazan je na slici 9. Padajući meni predstavlja glavni parametar koji određuje na koji način će se izvršiti treshold slike. Najvažniji izbori u padajućem meniju bi bili Binary i BinaryInv, osim ovog parametra potrebno je izabrati i prag za thresholding koji se kreće od 0-255, pomeranjem ručke na Track-bar kontroli menjaće se ovaj parametar, a samim tim će se i primenjivati ovaj filter na slici sa promenjenim parametrom. Binary opcija je najbolje da se koristi kada su tamna slova na svetloj pozadini, a u slučaju da su slova svetlija od pozadine poželjno je iskoristiti BinaryInv opciju, da bismo dobili binarizovanu sliku sa crnim (0) slovima i belom (255) pozadinom.

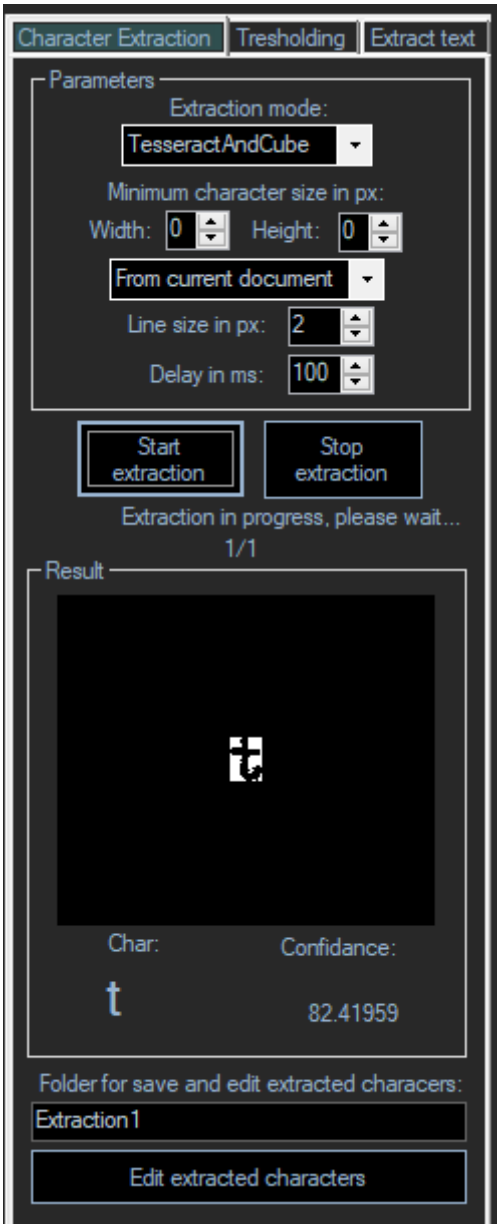
Najbolje je pre svega isprobati ovaj filter. Dugme sa natpisom “Try on this image” se koristi da se isproba filter i da se binarizovana slika prikaže, slika se tada neće sačuvati u aplikaciji, korisnik ima mogućnst da više puta isproba filter sa različitim pragom.



Slika 9. Tresholding i binarizacija

Kada se korisnik odluči za parametre koj mu odgovaraju potrebno je da klikne na dugme sa oznakom “Save changes“ da bi se binarizovana slika trajno sačuvala u programu. Izborom radio dugmića korisnik će odlučiti da li želi da uradi binaraizaciju sa istim parametrima na sve slike ako odabere „For all images“ ili samo za trenutno izabranu sliku, ako izabere opciju „For this image“.

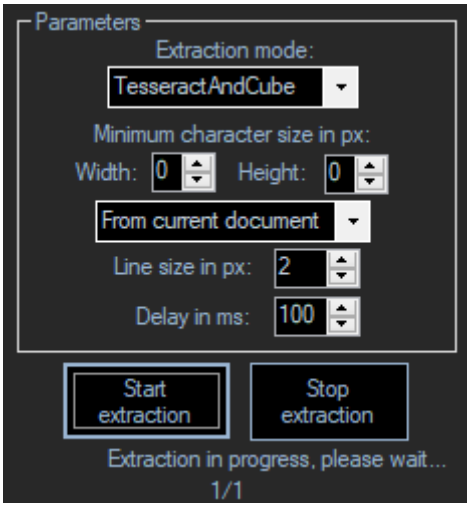
5.3. Separacija karaktera



Slika 10. Tab - Separacija karaktera

Na slici 10. Predstavljen je ceo odeljak koji služi za separaciju karaktera, na slici 11. je izdvojen deo za parametre u kome se viši odabir parametara za separaciju i ekstrakciju karaktera. Najpre je potrebno odabrati Extraction mod u padajućem meniju, TesseractAndCube mod je do sada kreatoru aplikacije davao najbolje rezultate pa ga je zato postavio kao defaultni mod. Nakon toga moguće je odabrati kolika bi bila minimalna veličina karaktera koji se ekstrakuje u pikselima,

moguće je uneti minimalnu šitinu i visinu. Drugi padajući meni predstavlja izbor koji se odnosi na to da li korisnik želi da obradi sve dokumente redom ili želi da obradi samo dokument koji je trenutno prikazan. Ako izabere opciju „From current document“ obradiće samo taj, a opcija „From all document“ obradiće sve dokumente redom.

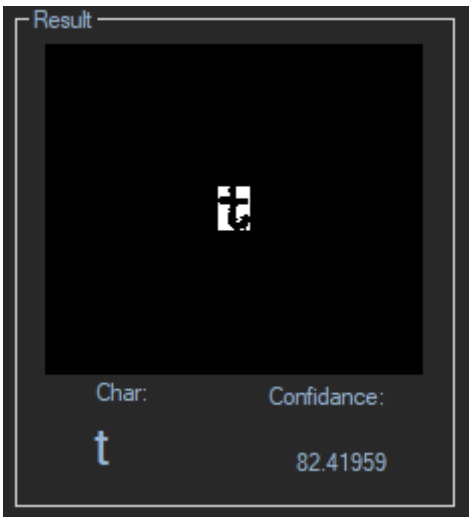


Slika 11. Parametri za ekstrakciju i separaciju karaktera

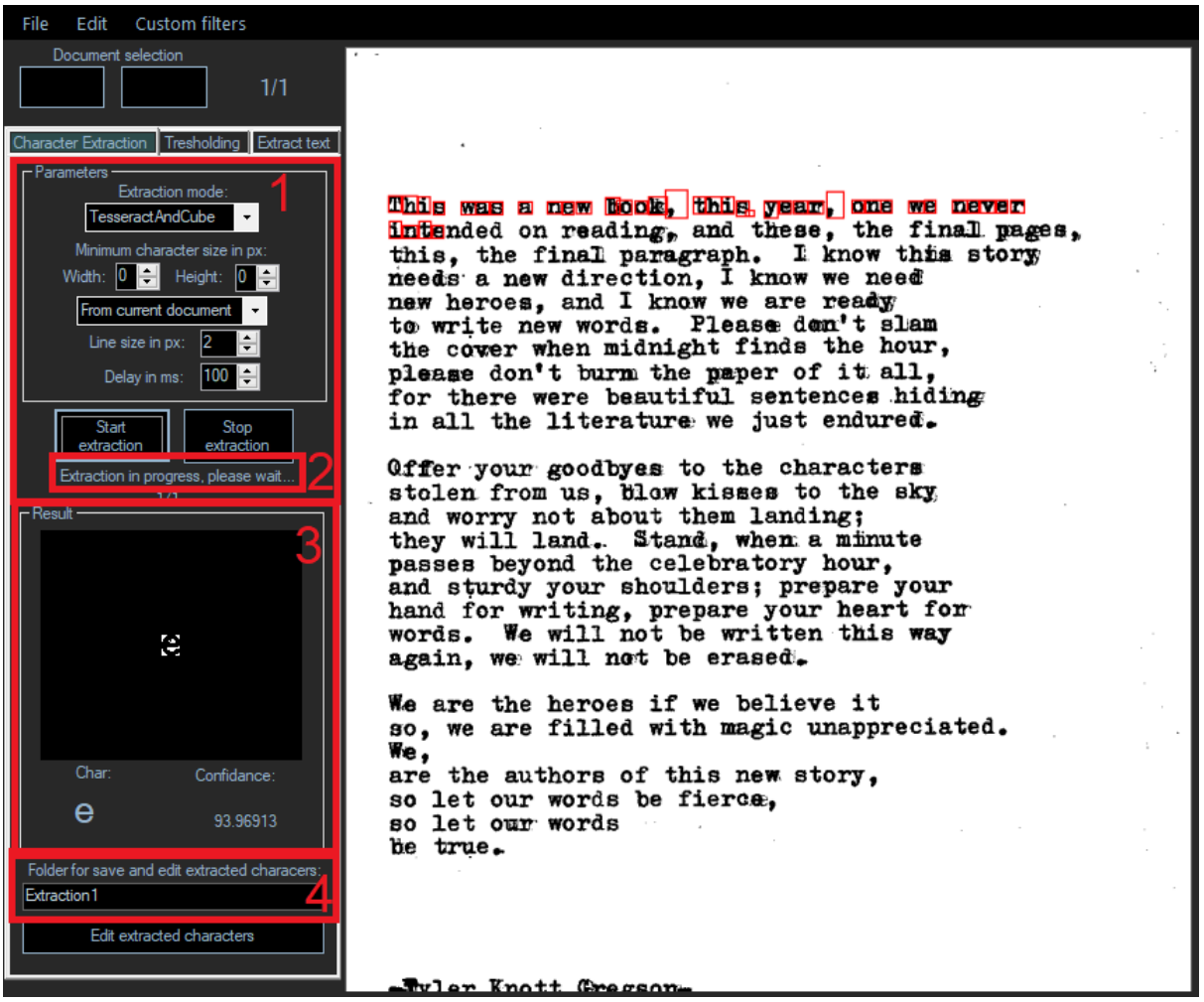
Naredne dve brojke predstavljaju - veličinu crvene linije u pikselima koja će uokvirivati karakter po karakter onim redom kojim ih obrađuje aplikacija i – kašnjenje u milisekundama koje postoji da bi korisnik vizuelno mogao da prati obradu dok se karakteri pojavljuju u kontroli na slici 12. o kojoj će kasnije biti reči.

U nastavku postoje dva dugmeta sa oznakama „Start extraction“ i „Stop extraction“, klikom na prvo pokrenućemo proces izdvajanja i čuvanja karaktera u fajl. Potrebno je sačekati od nekoliko sekundi do nekoliko minuta, u zavisnosti od dokumenta i računara, dok se dokument obradi i dok separacija karaktera krene. U toku ovog procesa moguće ga je stopirati klikom na drugo dugme. Ispod dugmića postoji labela koja označava napredak procesa i da li je proces gotov, a labela ispod nje označava redom najpre broj dokumenta koji se trenutno obrađuje i ukupan broj učitanih dokumenata.

Slika 12. predstavlja deo u kome se prikazuju karakteri redom koji su izdvojeni i koji će se sačuvati, ispod oznake „Char:“ nalazi se karakter koji je prepoznat, a ispod oznake „Confidance:“ procena koliko je program siguran da je to baš taj karakter na slici. Fajl u kome se čuvaju karakteri je imenovan ispod svega ovoga, na slici 10. I njegov naziv je „Extraction1“.



Slika 12. Prikaz izdvojenih karaktera, prepoznat karakter i oznaka koliko je program siguran da je to baš taj karakter



Slika 13. Program u toku ekstrakcije

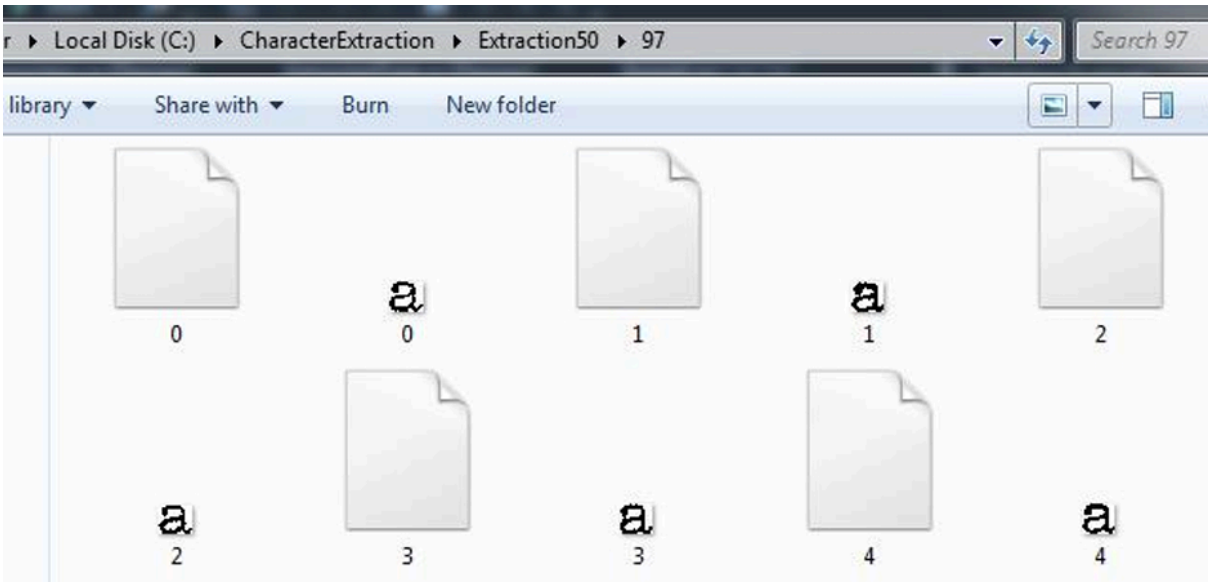
Na slici 13. prikazan je program u toku ekstrakcije karaktera, crvenim okvirima obeleženi su regioni od interesa vezan za proces ekstrakcije:

1. Izbor parametara:
2. Oznaka vezana za obradu
3. Deo za prikaz izvučenih karaktera
4. Fajl u kome se čuvaju karakteri

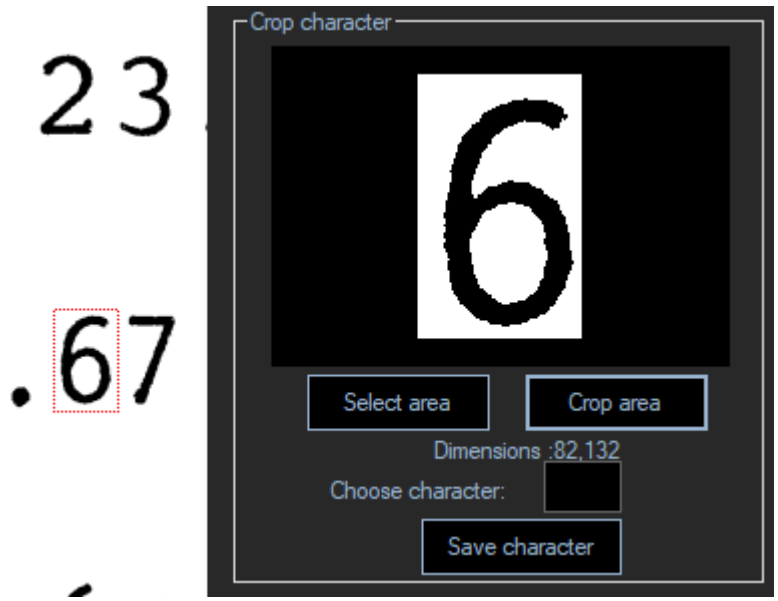
Karakteri na slici su obeleženi takođe crvenim okvirima, program obeležava redom karakter po karakter koji se obrađuje trenutno i prikazuje ga u delu pod brojem 3. Na ovoj slici je sačuvan snimak ekrana u toku obrade, pa nisu svi karakteri još oveleženi u tom trenutku.

5.4. Formiranje baze karaktera

Nakon ekstrakcije u odabranom folderu se kreira poseban folder koji dobija ime po ascii vrednosti karaktera koji se čuva, u tom folderu se čuva bitmapa izvučenog karaktera i poseban tekstualni fajl koji ima isti naziv kao bitmapa ali bez ekstenzije. Ovaj tekstualni fajl sadrži brojku koja predstavlja sigurnost da je taj karakter odgovarajući. Na slici 14 može se videti kako izgleda jedan folder u kome se čuva jedan određeni karakter.



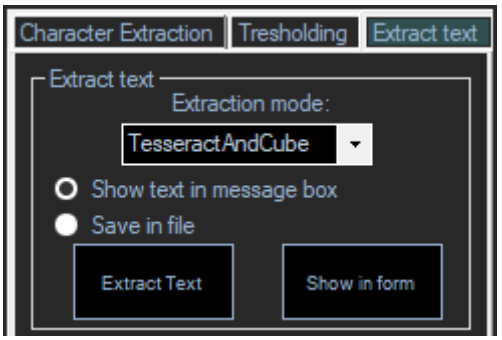
Slika 14. Izgled foldera (97 ascii je ’a’ karakter)



Slika 15. Ručno odsecanje karaktera

Odeljak u desnom delu na glavnoj formi, odnosno deo na slici 15. služi da ručno izdvojimo karakter sa slike i da ga dodamo u bazu. U slučaju da želimo da izdvojimo karakter potrebno je kliknuti najpre na dugme sa oznakom „Select area“, zatim možemo da odaberemo deo na slici koji želimo da izdvojimo, deo sa slike će se obeležiti isprekidanom crvenom linijom kao na slici 15. u levom delu, nakon toga je potrebno kliknuti na dugme sa oznakom „Crop area“. Posle svega toga izdvojeni deo će se očitati kao na slici 15., potrebno je još dodati koji karatker je izdvojen i kliknuti na dugme „Save character“ i on će biti sačuvan u istom folderu gde se čuvaju i oni karakteri koji su automatski dodati.

Dodatni deo aplikacije

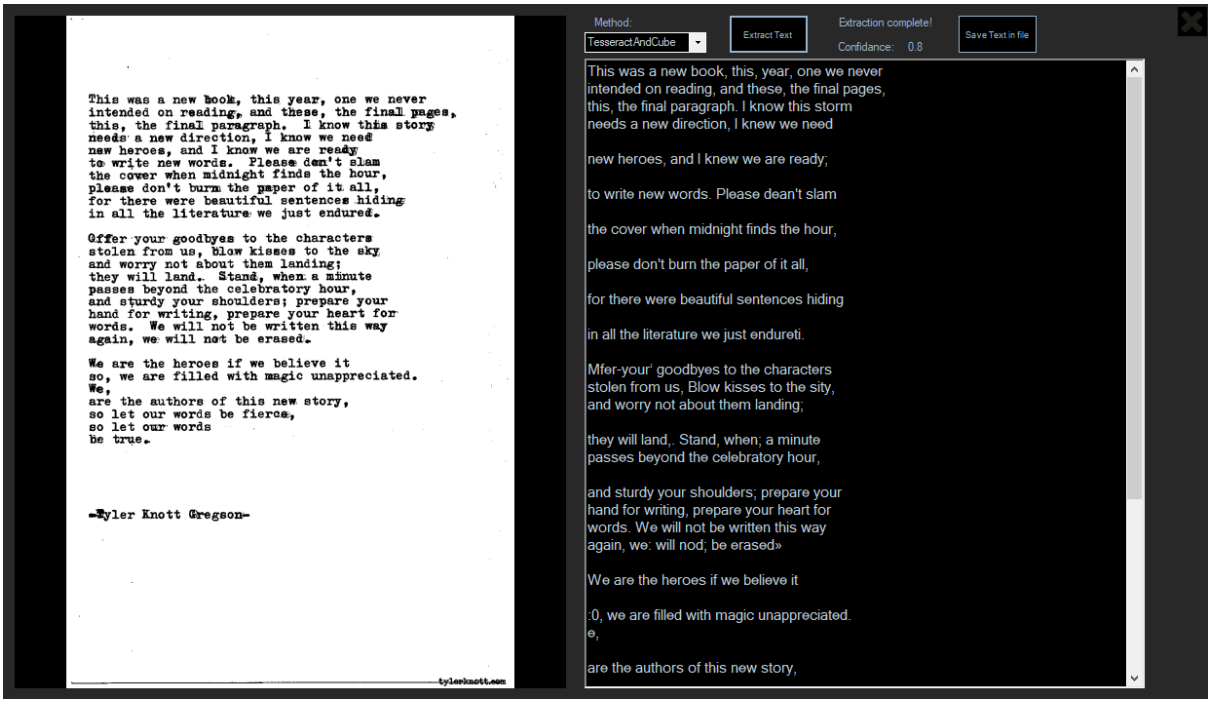


Slika 16. Odeljak za ekstrakciju kompletnog teksta sa dokumenta

Pre nego što se pređe na ostale stvari vezane za krajnje obrade dobijenih karaktera iz teksta biće opisan i poslednji deo na glavnoj formi. Deo sa slike 16. služi da bi se dobio kompletan tekst sa dokumenta. Moguće je najpre izabrati mod za ekstrakciju, a zatim izabrati jedan od dva ponuđena radio dugmeta, možemo izabrati da tekst iz trenutno izabranog dokumenta prikažemo u Message box formi, a

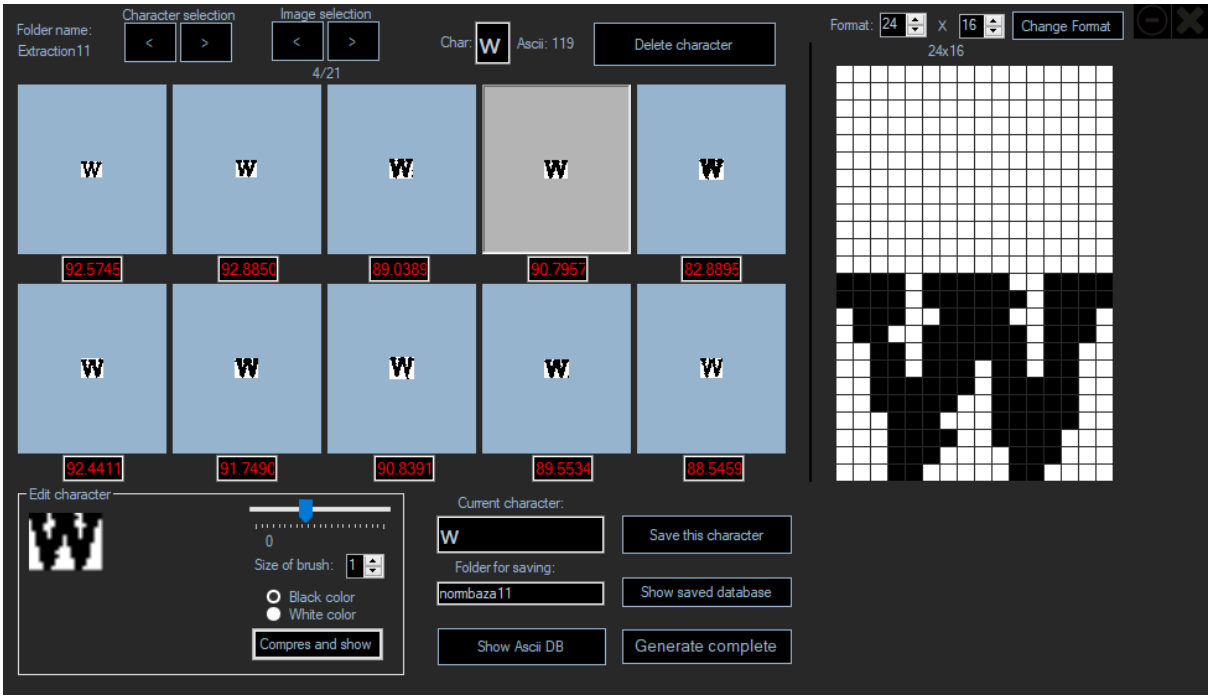
možemo ga snimiti odmah direktno u fajl. Nakon izbora potrebno je pritisnuti dugme sa oznakom „Extract Text“. Pre kreiranja fajla otvoriće se dialog u kome ćemo izabrati mesto, ime i ekstenziju fajla u koji želimo da snimimo tekst.

Ukoliko želimo da imamo bolji pregled svega toga, klikom na dugme sa oznakom „Show in form“ otvorićemo novu formu u kojoj će biti prikazani izabrani tekstualni dokument i polje u kome će se ispisati tekst koji se bude ekstrakovao. Potrebno je opet izabrati metod i potvrditi. Nakon dobijenog tekstra postoji mogućnost editovanja tog teksta ručno kako bi ispravili greške programa, ukoliko ih ima. Kada završimo dodatno editovanje moguće je snimiti test izborom dugmeta sa oznakom „Save Text in file“, opet će se pojaviti dijalog u kome ćemo odabrati lokaciju i naziv fajla u koji će se upisati tekst.(Slika 17.)



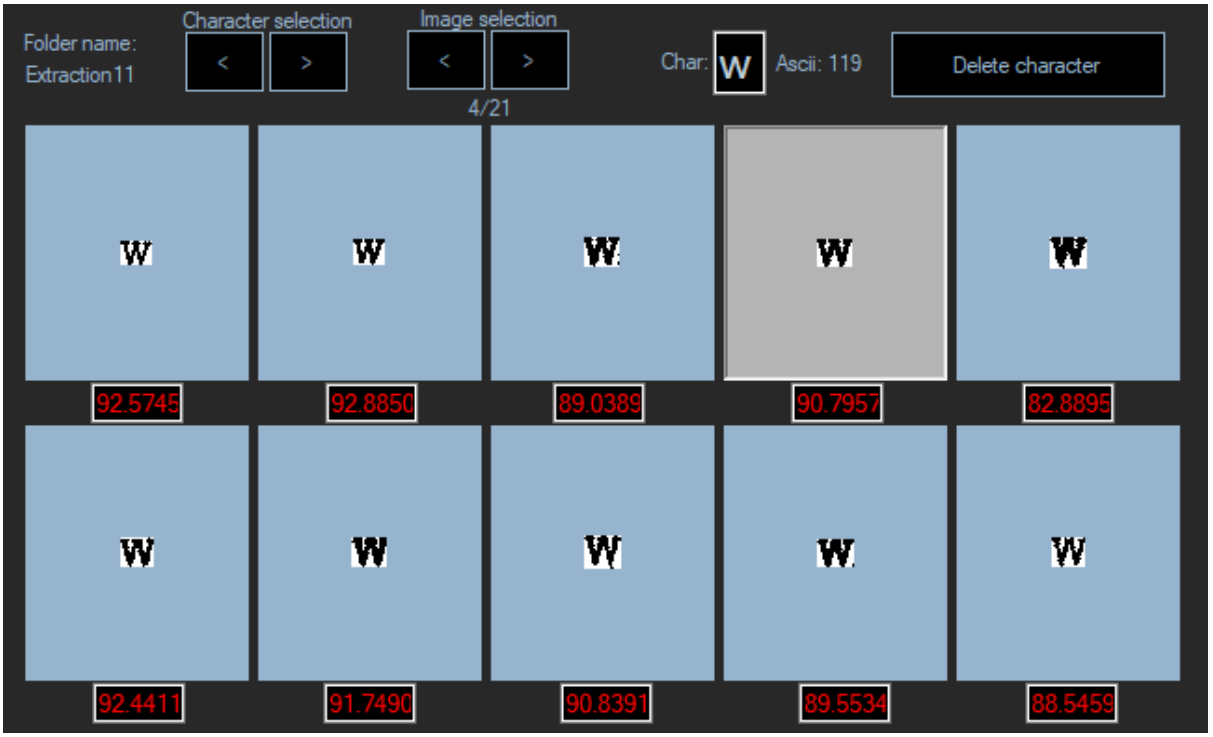
Slika 17. Uporedni prikaz dokumenta i ekstrakovanog teksta

5.5. Editovanje baze karaktera



Slika 18. Forma za obradu karaktera i kreiranje normalizovane baze karaktera

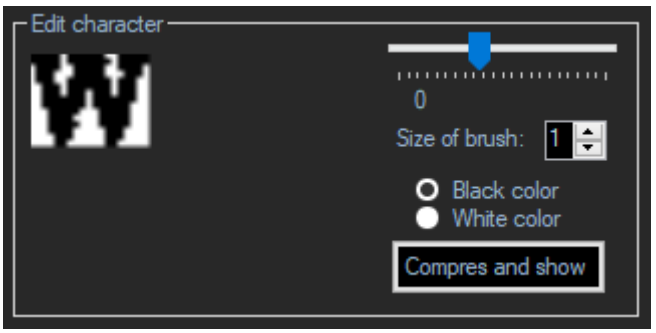
Na slici 18. predstavljena je druga najvažija forma u kojoj će automatski ili sa nekom od ručniih metoda biti odabrano koji će karakter biti odabran kao predstavnik u bazi karaktera.



Slika 19. Deo za prikaz, selekciju i brisanje karaktera

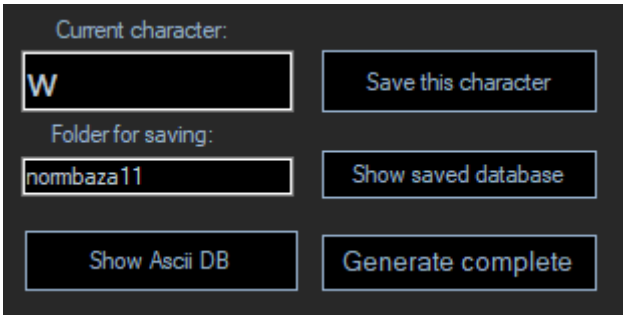
Slika 19. predstavlja deo u kome se izvučeni karakteri mogu listati a takođe i brisati. Duglići ispod oznake „Character selection“ služe da listaju različite foldere, jedan folder se ondnosi na jedan karakter. Dugmići ispod oznake „Image selection“ služe da listaju izvučene karaktere iz jednog foldera, brojevi ispod tih dugmića predstavljaju redom broj karaktera koji je trenutko selektovan i ukupan broj karaktera koji se nalaze u tom folderu. Selektovani karakter je moguće izbrisati klikom na dugme sa oznakom „Delete character“. Svaki karakter ispod slike sadrži i odgovarajući skor, tj. broj, što je skor veći to je veća verovatnoća da je to baš taj predstavljeni karakter odnosno to je bolja reprezentacija tog karaktera.

U trenutku kada karakter biva očitan slika ovog karaktera se automatski prebacuje u deo za editovanje na slici 20. i automatski se kompresuje u sliku veličine 24x16 piksela (ili neke korisnički odabrane veličine) i iscrtava se u delu koji se nalazi na slici 22. Moguće je editovati izabrani karakter, najpre podesiti zoom pomoću track-bar-a ili točkićem miša, odaberite boju i veličinu četkice u pikselima i zatim se može crtati po kontroli u kojoj se nalazi karakter, slika 20. Na kraju je moguće opet kompresovati karakter klikom na dugme sa oznakom „Compres and show“ i kompresovani karakter će se iscrtati u delu sa slike 22.

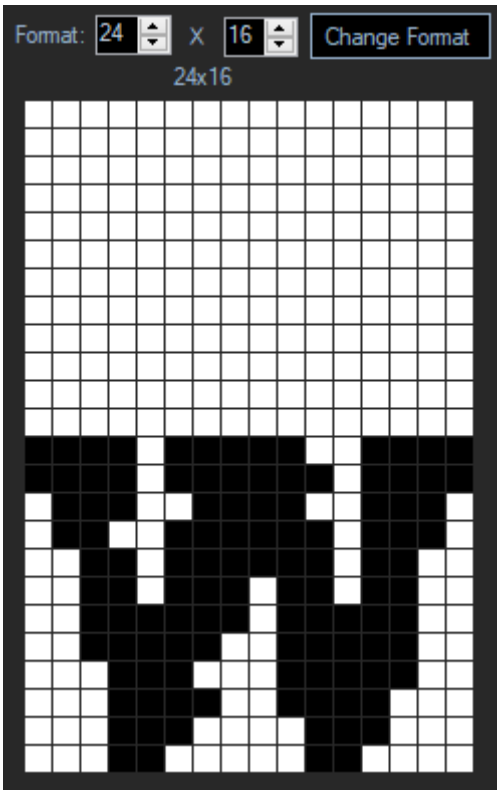


Slika 20. Editovanje binarizovanog karaktera

Kada korisnik izmeni karakter i kompresuje ga opet u sliku odabrane veličine MxN potrebno je da unese koji je to karakter u polje ispod oznake „Current character“ i da klikne na dugme sa oznakom „Save this character“ i karakter će biti snimljen u normalizovanu bazu. O normalizovanoj bazi više u delu 6.

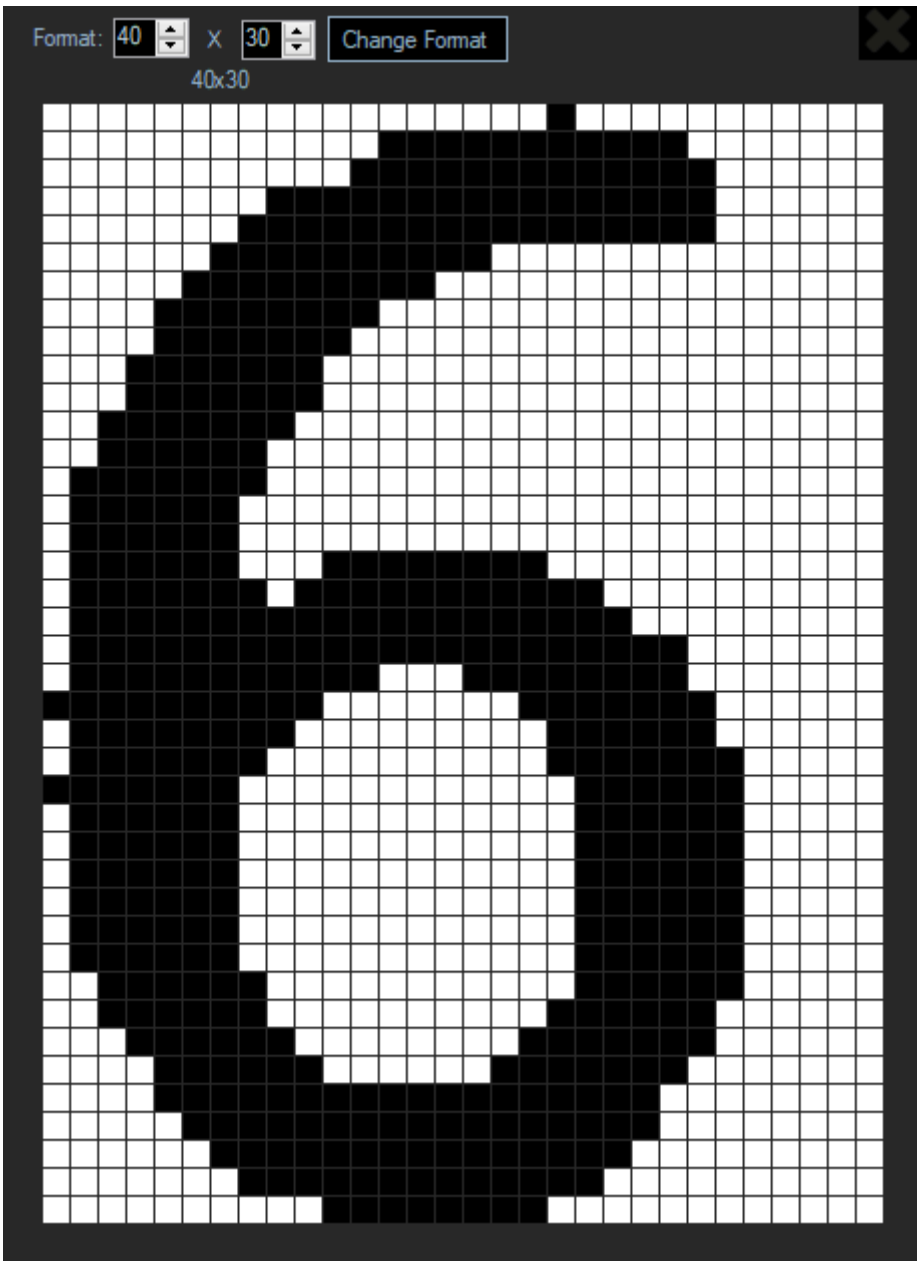


Slika 21. Snimanje u bazu



Slika 22. Kompresovan binarizovani karakter 24x16 piksela

Deo na slici 22. gde se nalazi kompresovani karakter moguće je editovati klikom na kvadratiće bele i crne boje, kvadratići će zameniti boju inverznu boju.



Slika 23. Kompresovan binarizovani karakter 40x30 piksela

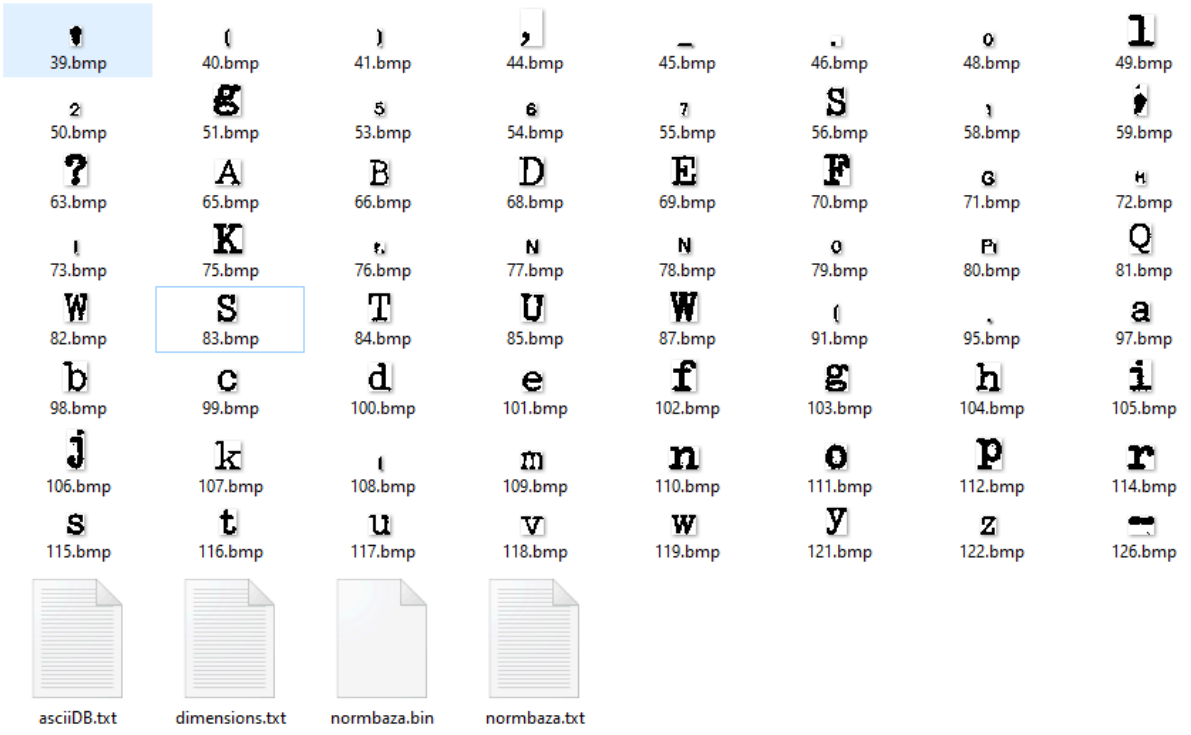
Slika 23. Predstavlja kompresovani binarizovani karakter 40x30 piksela kome je standardna veličina iz aplikacije promenjena. U zaglavlju slike 23. može se primetiti da postoje dve kontole za format koje predstavljaju redove i kolone i kontrola koja je dugme sa labelom “Change Format”. Postoji mogućnost da se u aplikaciji promeni format karaktera koji se čuva u bazi, potrebno je izmeniti redove i kolone u ovim kontrolama i klikom na dugome naredni karakteri će se kompresovati u novu veličinu. Stara baza biće izbrisana, jer nije moguće čuvari karaktere različitih dimenzija.

5.6.Normalizacija

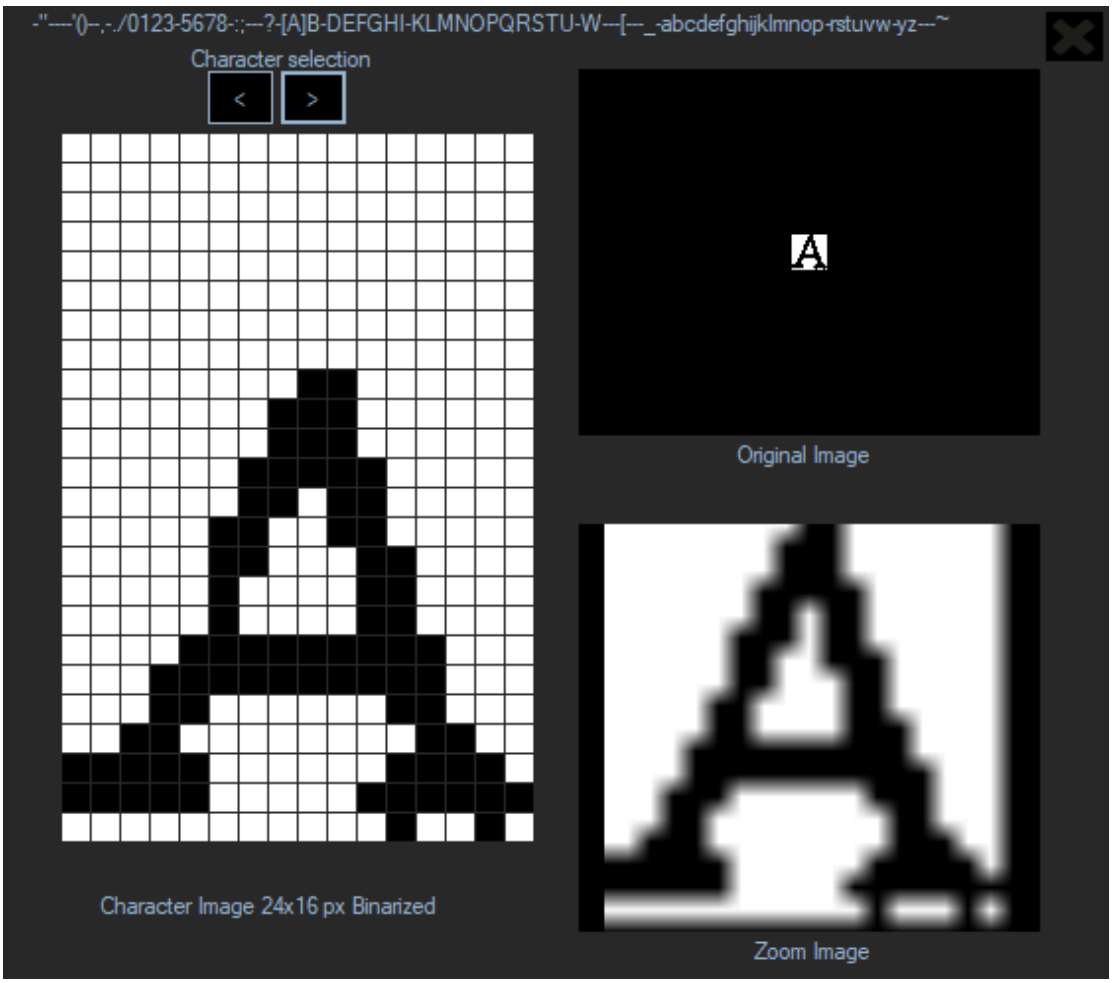
Da bi se editovani karakter snimio u bazu potrebno je odabrati ime baze tako što u delu sa slike 21. ispod oznake „Folder for saving:“ korisnik unese ime tog foldera za snimanje.

Pored ručnog dodavanja karaktera moguće je generisati kompletnu normalizovanu bazu, za generisanje kompletne normalizovane baze karaktera koristi se dugme sa oznakom „Generate complete“ sa slike 21. Ova opcija će izabrati karaktere sa najboljim skorom i dodati ih u normalizovanu bazu. Nakon izbora opcije izaćiće par MessageBox sa par pitanja vezanih za kreniranje nove baze, da li da ostanu sačuvani prethodni karakteri i tome slično.

Normalizovana baza se čuva u obliku koji je opisan u početnom odeljku pod nazivom Zadatak. Pored fajlova normbaza.bin, normbaza.txt, asciiDB.txt i dimensions.txt čuva se i slika karaktera u bmp formatu koja ima naziv kao ascii vrednost tog karaktera (Slika 24).



Slika 24. Izgled foldera normalizovane baze



Slika 25. Forma za prikaz normalizovane baze karaktera

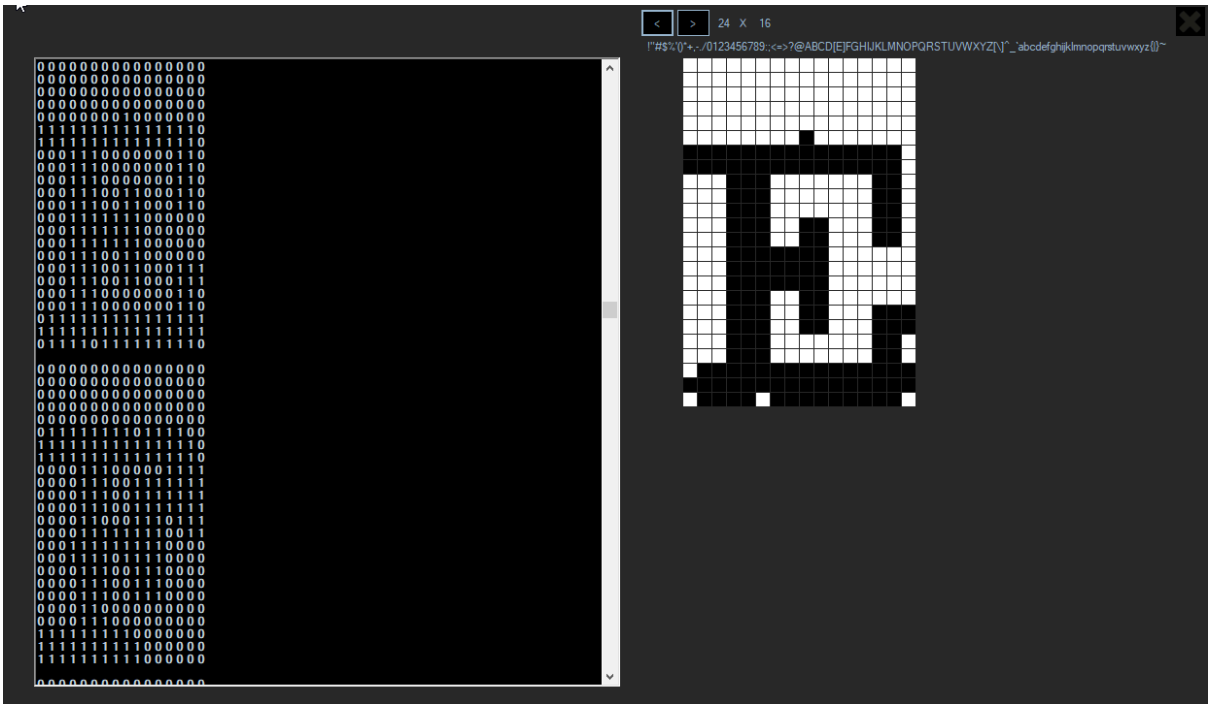
Prikaz normalizovane baze Na slici 21. u u sredini desno nalazi se deo preko koga se pristupa normalizovanoj radi pregleda sačuvane baze. Kontrolu odnosno dugme „Show saved database“, potrebno je kliknuti i otvoriće se nova forma za pregled sačuvanih karaktera, forma sa slike 25.

Ispod oznake „Character selection“ nalaze se dugmići koji se koriste za listanje karaktera iz baze. Sa desne strane prikazan je kompresovan karakter, a levo su prikazani redom originalna slika i zumirana slika.

Svi folderi i fajlovi koji se koriste kao podaci u programu se čuvaju u folderu C:\CharacterExtraction koji će biti automatski kreiran od strane aplikacije.

5.7.Prikaz formata u Ascii redosledu

Pored svih do sada opisanih kontrola sa slike 21. postoji i dugme sa labelom “Show Ascii DB” koje će najpre otvoriti dijalog za pretragu tekstualnih u po folderima sačuvanih baza u C:\CharacterExtraction gde je moguće izabrati neki od tekstualnih fajlova posebnog Ascii formata. Moguće je otvoriti i fajl ovog formata iz bilo kog dela file sistema u računaru. Ako je fajl ovog pravog formata otvoriće se forma sa slike 26. Levo će biti prikazan deo sirovog fajla ovog formata, a desno su prikazani karakteri koji se listaju pomoću kontrola u gornjem delu koji su izvučeni iz ovog formata. Takođe i ovaj format će biti zapamćen u formatu iste veličine MxN kao i binarni kompresovani format i u prva dva reda ovog tekstualnog fajla ispisane su broj redova i broj kolona.



Slika 26. Forma za prikaz formata Ascii tipa

6. Zaključak

Razvoj digitalne obrade slike i OCR tehnologije omogućio je značajne napretke u mnogim oblastima. Kroz ovaj rad, istražiće se mogućnosti i izazovi razvoja takve aplikacije, sa ciljem da se doprinese daljem napretku u oblasti digitalne obrade slike i OCR tehnologije.

Ova aplikacija kombinuje napredne tehnike obrade slike i OCR prepoznavanja kako bi omogućila korisnicima da jednostavno ekstraktuju tekst iz slika i kreiraju jedinstvene fontove. Ova tehnologija ima širok spektar primena, uključujući digitalizaciju starih rukopisa, personalizaciju dizajna, te olakšavanje rada u kreativnim industrijama.

U zaključku, razvoj aplikacije ne samo da demonstrira praktičnu primenu naprednih tehnologija obrade slike i OCR-a, već i otvara vrata za nove inovacije i mogućnosti u digitalnom svetu. Iako postoje tehnički izazovi, kao što su preciznost prepoznavanja karaktera i optimizacija performansi, kontinuirani napredak u ovoj oblasti obećava još veće potencijale za budućnost. Integracijom ovih tehnologija u svakodnevne alate, olakšava se proces kreiranja digitalnih sadržaja, omogućavajući korisnicima da brzo i efikasno ostvaruju svoje ideje.

Dalje istraživanje i razvoj u ovom domenu će nesumnjivo doneti nove metode i poboljšanja, čime će se proširiti mogućnosti i unaprediti kvaliteta aplikacija koje se oslanjaju na digitalnu obradu slike i OCR tehnologiju. Ovaj rad predstavlja značajan korak u tom pravcu, istovremeno služeći kao inspiracija za buduće projekte i inovacije, sledeca napradnija i proširenija verzija ovog projekta biće tema Master rada.

7. Literatura

[1] Basic tresholding OpenCV
https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html

[2] Tesseract software
https://en.wikipedia.org/wiki/Tesseract_(software)

[3] Tesseract Github
https://github.com/tesseract-ocr/tesseract

[4] EmguCV
https://www.emgu.com/wiki/index.php/Main_Page

[5] EmguCV Github
https://github.com/emgucv/emgucv

[6] Windows Forms
https://en.wikipedia.org/wiki/Windows_Forms

[7] Odsecanje dela slike
https://www.youtube.com/watch?v=7IR6J8Kw8cE

[8] Izdvajanje karaktera
https://github.com/halanch599/Emgucv/blob/master/Text%20character%20extraction%20from%20images/formcharactsegmentation.cs

[9] Tesseract
https://www.youtube.com/watch?v=rAkf8S1G-Aw

[10] Ekstrakcija teksta
https://www.youtube.com/watch?v=6jmhv52_iJM

[11] Ekstrakcija teksta - saveti
https://www.youtube.com/watch?v=rAkf8S1G-Aw

[12] Binarizacija
https://www.youtube.com/watch?v=KpCQp_rd-Nk

[13] Adaptive thresholding
https://www.youtube.com/watch?v=Bjtg0RFm6po

[14] Ekstrakcija kontura objekta
https://www.youtube.com/watch?v=fT9o3F4g3rE

[15] Jednostavan projekat za crtanje
https://www.youtube.com/watch?v=iAC2d5zUjzU

[16] Jednostavan grafički editor
https://www.youtube.com/watch?v=igJNruoPPh4

[17] Snimanje slike
https://www.youtube.com/watch?v=T3jds4To7k4

[18] Promena Veličine slike
https://www.youtube.com/watch?v=eiakPE9R7aw

[19] Tesseract diskusije
https://github.com/charlesw/tesseract/issues/64

[20] Tesseract OCR Pozicija teksta
https://stackoverflow.com/questions/51282214/tesseract-ocr-text-position

[21] Tesseract Istraživanja
https://research.aimultiple.com/ocr-accuracy/