Prepoznavanje rukopisa

Dalibor, Šuplika

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:748617

Rights / Prava: Attribution-NonCommercial 3.0 Unported/Imenovanje-Nekomercijalno 3.0

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Dalibor Šuplika

PREPOZNAVANJE RUKOPISA

ZAVRŠNI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Dalibor Šuplika

Matični broj: 0016120010

Studij: Poslovni sustavi

PREPOZNAVANJE RUKOPISA

ZAVRŠNI RAD

Mentor/Mentorica:

Doc. dr. sc. Petra Grd

Varaždin, rujan 2019.

Dalibor Šuplika

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Cilj ovog završnog rada je prepoznati ručno napisani tekst i odrediti koji pisac iz baze je napisao taj tekst na temelju skenirane slike teksta korištenjem neuronske mreže i algoritama ugrađenih u računalni program. Kod prepoznavanja rukopisa razlikujemo dvije metode: online i offline metodu prepoznavanja. Obje metode su opisane u radu kako bi se shvatila razlika između njih, te je opisana njihova primjena u stvarnom životu. Također u radu su opisana svojstva pisanja, te prednosti i nedostatci prepoznavanja rukopisa. Praktični dio rada koristi offline metodu. Za izradu programa korišten je Python 3.6.1., neuronska mreža za prepoznavanje autora koju sam sam istrenirao na temelju slika preuzetih s „IAM Handwriting Database“, SimpleHTR neuronska mreža za prepoznavanje napisanog teksta te moduli Keras 2.0.9 i Tensorflow 1.8.0.

Ključne riječi: rukopis; prepoznavanje; segmentacija; normalizacija; uklanjanje zrnatosti; digitalizator; značajke;

Sadržaj

Sadržaj iii
1. Uvod 1
2. Prepoznavanje rukopisa 2
2.1. Svojstva pisanja 2
2.2. Prednosti prepoznavanja rukopisa 3
2.3. Problemi prepoznavanja rukopisa 4
3. Offline prepoznavanje rukopisa 5
3.1. Offline prepoznavanje rukopisa s višedimenzionalnim rekurentnim neuronskim mrežama 7
3.2. Primjena offline metode 9
4. Online prepoznavanje rukopisa 10
4.1. Tehnologija digitalizatora 10
4.2. Pred obrada 11
4.3. Prepoznavanje oblika 12
4.3.1. Analiza značajki 12
4.3.2. Vremenski slijed zona, ekstrema ili smjerova 13
4.3.3. Podudaranje krivulja 13
4.3.4. Prepoznavanje nizova znakova i pisanih slova 14
4.4. Naknadna obrada 14
5. Aplikacija za prepoznavanje rukopisa i određivanja autora 15
5.1. IAM Handwriting Database 15
5.2. Korištene tehnologije i alati 17
5.3. Prikaz aplikacije 18
5.3.1. Poznati problemi kod aplikacije 20
5.4. Testiranje neuronske mreže 20
6. Zaključak 23
Popis literature 24
Popis slika 26

1. Uvod

Rukopis je osobna vještina svakog pojedinca. Rukopis se sastoji od umjetnih grafičkih oznaka kojima je svrha prenijeti poruku. Svaka riječ sačinjena je od skupa ikona koje nazivamo slovima. Smatra se da je pisanje omogućilo veći dio kulture i civilizacije.[1]

Nekoliko vrsta analiza, prepoznavanja i tumačenja može se povezati s rukopisom. Prepoznavanje rukopisa je pretvaranje jezika predstavljenog u njegovom prostornom obliku grafičkih oznaka u njegov simbolički prikaz. Tumačenje rukopisa ili interpretacija rukopisa je određivanje značenja ručno napisanog pojma, na primjer da je ručno napisani tekst adresa. Identifikacija rukopisa je zadatak određivanja autora ručno napisanog teksta iz skupa autora, te provjeravanje potpisa je utvrđivanje da li dani potpis pripada odgovarajućoj osobi. Identifikacija i verifikacija koriste se u forenzičkoj analizi te određuju posebne osobine u pisanju određenog pisca, dok prepoznavanje rukopisa i tumačenje rukopisa su procesi kojima je cilj filtrirati varijacije kako bi se odredila poruka.[1]

2. Prepoznavanje rukopisa

Prepoznavanje rukopisa je sposobnost računala da zaprimi i interpretira ručno napisani tekst. Za engleski pravopis, kao i za sve ostale jezike koji koriste Latinsku abecedu, simbolički prikaz je 8-bitni ASCII prikaz znakova. Znakovi većine pisanih jezika svijeta danas su prikazani u obliku UTF-16.[1]

Napisani podatak je pretvoren u digitalni oblik skeniranjem papira na kojem je podatak napisan ili pisanjem uz pomoć posebne olovke po elektroničkoj površini. Ovisno o metodi korištenja razlikujemo offline metodu prepoznavanja rukopisa i online metodu prepoznavanje rukopisa. Kod offline metode sve što je dostupno je gotova slika teksta[1], dok se kod online metode pohranjuju vremenski nizovi koordinata, koji predstavljaju kretanje vrha olovke.[3] Uz pomoć vremena može se ustanoviti redoslijed unosa. Online metoda bavi se prostorno-vremenskom reprezentacijom unosa, dok offline metoda analizira prostorno osvjetljenje slike.[1] Izdvajanja relevantnih značajki kod online prepoznavanja je mnogo lakše, te online metoda obično dalje bolje rezultate od offline metode. Također, važna razlika je između prepoznavanja izoliranih znakova ili riječi i prepoznavanja cijelih redaka teksta. Prepoznavanje cijelih redaka teksta je znatno teže od prepoznavanja pojedinih znakova ili riječi, te izvrsni rezultati koji su dobiveni za prepoznavanje riječi i znakova nikada nisu bili usklađeni s potpunim linijama. [3] Neovisno o metodi koja se koristi, standardni postupak prepoznavanja je u biti isti. Prvo se izdvoji niz značajki iz podataka, zatim se taj niz značajki usklađuje s nizom slova korištenjem Skrivenog Markovljevog Modela (eng. *Hidden Markov Models - HMM*) ili hibrida između HMM i neuronskih mreža.[2]

2.1. Svojstva pisanja

Pisani jezik ima abecedu slova (ili znakova), interpunkcijske simbole i slično. Temeljno svojstvo pisanja koje omogućuje komunikaciju je da su razlike između različitih slova značajnije od razlika između različitih crteža istog slova. Tvrdi se da postoje iznimke ovome jer O i 0 mogu biti napisani identično, iako kontekst obično pruža informacije potrebne za razlikovanje slova od brojeva. Rukopis se sastoji od vremenskog slijeda poteza, pri čemu je se pod jedan potez smatra trenutka kada je olovka dotakla površinu po kojoj se piše do trenutka kada se olovka odvojila od površine po kojoj se piše. Znakovi pisanja obično se formiraju u slijedu, jedan znak se dovršava prije početka sljedećeg, a znakovi tipično slijede neki prostorni poredak, kao što je slijeva nadesno. Postoje neke iznimke ovome kod pisanih slova, npr. točke (kod j, i) , kvačice (kod č, ć, š ,ž) i crtice (kod t, đ) znaju biti odgođene na

način da se prvo napiše temelj riječi i tada se riječ završava dodavanjem točaka, kvačica i crtica.[5]

Uzmimo u obzir dva jezika: Engleski i Kineski. Engleska abeceda ima 26 slova, gdje svako slovo ima dva oblika: velika i mala slova. Engleski ima dva osnovna stila pisanja: tiskana i pisana slova. Engleske riječi sastoje se od nizova slova, prosječno pet po riječi. Velika tiskana slova pisana rukom u prosjeku su napisana u dva poteza po slovu, dok mala slova u jednom potezu po slovu. Na engleskom, mjesto i veličina slova su važni. Velika slova se nalaze na osnovnoj liniji i pune su veličine. Mala slova su manja, a većina su otprilike pola visine velikih slova. Iznimka kod malih slova su slova s uzlaznim potezima (npr. b, d, l) i silaznim potezima (npr. j, p). Kineski jezik ima mnogo veći skup znakova u abecedi. Kineski znak može predstavljati riječ. Postoji oko 50 000 znakova, a osnovni rječnik se sastoji od 3000 do 5000 znakova. Postoje dva osnovna stila pisanja znakova: blok i kurziv. Blok stil piše se pažljivo, s prilično strogim pridržavanjem pravilnog broja i redoslijeda poteza. Znak u blok stilu ima prosječno 8 do 10 poteza, najjednostavniji znak ima jedan potez, a najsloženiji više od 30. Kurzivni stil se piše brže i s manje poteza. To se postiže spajanjem nekih poteza blok stila i korištenjem jednostavnijih radikalnih oblika. Svi znakovi se razlikuju po statičkim i dinamičkim svojstvima. Statička varijacija može se pojaviti, na primjer, u veličini ili obliku slova, dok se dinamička varijacija može pojaviti u broju i redoslijedu poteza. Stupanj varijacije ovisi o stilu i brzini pisanja, pri čemu brže pisanje daje veću varijaciju. Promjenjivost rukopisa je proučavana i općenito i sa stajališta prepoznavanja rukopisa.[5]

2.2. Prednosti prepoznavanja rukopisa

Iako je postignut veliki napredak u načinu na koji se podaci prezentiraju korisniku, poput alata za vizualizaciju podataka, primarni način unosa podataka od korisnika do računala i dalje je tipkovnica. Iako se velik dio današnjih podataka unosi u računala pomoću tipkovnice, još uvijek postoji mnogo zadataka koje ljudi više preferiraju unositi rukom nego tipkovnicom, npr. bilješke s predavanja, izrada nacrtanih skica, itd. Osobni digitalni asistenti su džepni uređaji koji mogu pohraniti kalendare, pružaju pristup e-pošti i sadrže druge produktivne alati. Ti uređaji su premali da bi imali tipkovnice u punoj veličini, ili ponekad može biti premala za bilo koju tipkovnicu, pa za unos podataka zahtijeva olovka ili glasovna sučelja. Prepoznavanje govora i prepoznavanje rukopisa omogućuju druge prirodnije oblike komunikacije čovjeka i računala. Prepoznavanje rukopisa je superiornije prepoznavanju govora jer je neosjetljiv na buku iz okoliša, što je važna prednost za mnoge aplikacije.[9]

Neki jezici sadrže jako veliki broj simbola (npr. kineski Kanji, japanski Hiraganu i Katakanu) radi čega unos preko tipkovnice postaje još teži zadatak. Za ove jezike

prepoznavanje rukopisa može pružiti mnogo učinkovitiju metodu unosa podataka, međutim, kako bi bili korisni, trebaju biti ekonomični, korisnički prihvatljivi i njihova točnost prepoznavanja mora biti izrazito visoka.[9]

Još jedna prednost online prepoznavanja je prilagodba. Kad korisnik vidi da se neki njegovi likovi ne prepoznaju točno, može izmijeniti njihov crtež kako bi poboljšao prepoznavanje. S druge strane, neki alati za prepoznavanje rukopisa sposobni su se prilagoditi piscu obično pohranjivanjem uzoraka piščevih likova za kasnije prepoznavanje.[9]

2.3. Problemi prepoznavanja rukopisa

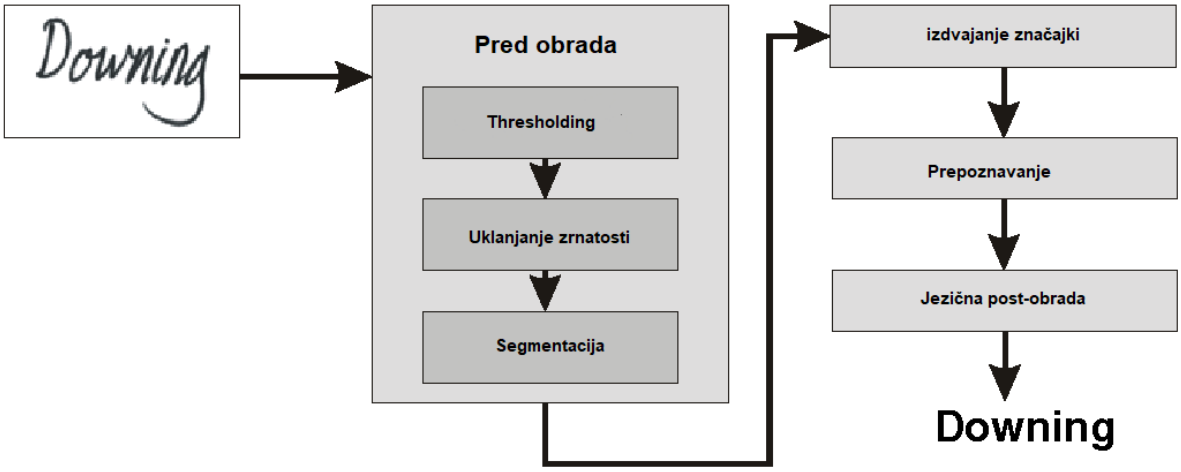
Problemi prepoznavanja uzoraka uključuju prepoznavanje simbola jezika, jednadžbi, crteža i gesta. Problemi prepoznavanja simbola jezika uključuju, na primjer, veliku abecedu kineskih znakova, japansku Hiraganu i Katakanu, korejski Hangul, arapski i stilove pisanja zapadnih jezika. Za rješavanje problema stilova pisanja potrebno je izvršiti segmentaciju znakova. Segmentacija znakova je proces razdvajanja znakova (slova) u riječi jedno od drugog tako da svako slovo bude zasebno. Segmentaciju je teže izvršiti kod stilova pisanja gdje su slova napisana jedno blizu drugog, te zahtijevaju napredne tehnike segmentacije, uključujući interakciju segmentacije i prepoznavanja znakova. Tiskana slova je lakše segmentirati od pisanih jer su znakovi diskretni i sastoje se od jednog ili više poteza, za razliku od pisanih slova koja su povezana i moguće je jednim potezom napisati više slova ili čak cijelu riječ. Također, problem prepoznavanja se javlja kod znakova koji su sličnog oblika (npr. U i V, a i d, n i h) i znakova koji su slični brojevima (npr. O i 0, Z i 2, S i 5). Slovo O i brojka 0 se mogu napisati identično, te se mogu razlikovati samo po kontekstu.[5]

3. Offline prepoznavanje rukopisa

Offline metoda prepoznavanja rukopisa općenito se smatra težom od online metode prepoznavanja rukopisa. U online metodi značajke se mogu izdvojiti iz putanje olovke i dobivene slike, dok kod offline metode sve što je dostupno je slika teksta.[2] Glavni zadatci offline metode su prepoznavanje znakova i prepoznavanje riječi. Neophodni korak kod prepoznavanja pisanog jezika je zadatak zvan analiza dokumenta. Analiza dokumenta je prostorni problem lociranja i registriranja odgovarajućeg teksta kada se koriste složeni dvodimenzionalni prostorni raspored. Potrebno je izvršiti nekoliko operacija analize dokumenata prije nego se može prepoznati tekst na skeniranom dokumentu. Neke od uobičajenih operacija su: thresholding, uklanjanje zrnatosti, segmentacija linija i segmentacija riječi i znakova.[1]

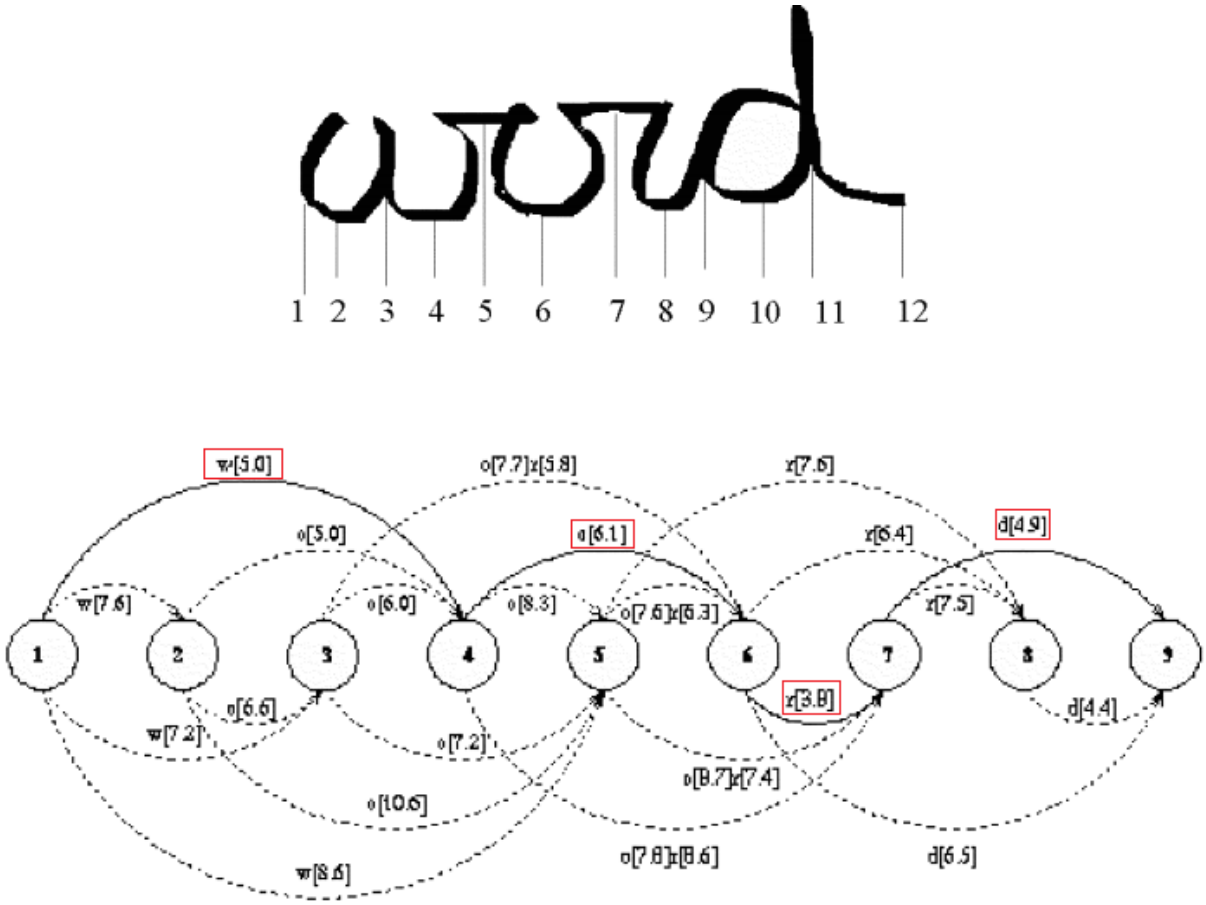
Thresholding je pretvaranje gray-scale slike u binarnu crno-bijelu sliku. Postiže se odvajanjem prednjeg dijela od pozadine, odnosno u ovom slučaju teksta od papira. Digitalnim snimanjem slika može doći do zrnatosti od skenera te treba primijeniti uklanjanje zrnatosti (eng. *noise removal*) kako bi se to uklonilo. Segmentacija linija je odvajanje pojedinih linija teksta jedne od druge, što može biti problem kod redova ručno napisanih riječi radi uzlaznih (npr. malo pisano d) i silaznih (npr. malo pisano j) slova koja mogu priječi iz jednog reda u drugi. Nakon što se redovi odvoje za svaki red posebno radi se segmentacija riječi koja odvaja jednu riječ od druge te segmentacija znakova koja odvaja svako slovo zasebno.[1]

Osnovni problem kod prepoznavanja rukopisa je dodijeliti digitalni znak njegovoj simboličkoj klasi. Kod rukopisa taj proces se naziva inteligentno prepoznavanje znakova (eng. *intelligent character recognition*). Tipične klase su velika i mala slova, deset znamenki brojeva (0-9), te specifični simboli kao što su točka, zarez, uskličnik, upitnik, zagrade, znakovi matematičkih operacija i slično. Koristi se algoritam za prepoznavanje uzoraka kako bi se izlučile značajke oblika i tom obliku dodijelila odgovarajuća klasa. Prepoznati isprintan znak na čistom papiru može se učiniti s visokom točnošću. Do problema dolazi kod ručno napisanog teksta iz razloga jer svaka osoba ima drugačiji rukopis te napisana slova nisu identična kao kod teksta koji je isprintan na papiru u određenom fontu. Kod određenih slučajeva prepoznavanja rukopisa postaje potrebno koristiti modele kako bi ograničili izbore na razini slova i riječi. Ovakvi modeli su nužni kod prepoznavanja rukopisa zbog velike varijabilnosti ručnog ispisa i pisanih slova. [1]



Slika 1: Blok dijagram offline sustava prepoznavanja rukopisa[4]

Kod offline prepoznavanja rukopisa postoje 2 pristupa: analitički pristup i holistički pristup. Analitički pristup prepoznaje znak po znak, dok holistički pristup uzima cijelu riječ u obzir, međutim, veća razina uspješnosti se dobiva kombiniranjem rezultata oba pristupa. Jedna od metoda prepoznavanja riječi temelji se na određivanju pre-segmentacijskih točaka nakon čega slijedi određivanje optimalnog puta kroz dijagram prijelaza stanja, što je prikazano na slici 2, dok drugi pristup koristi ideju redovitih i jedinstvenih obilježja iz razloga što se smatra da rukopis ima redoviti protok koji se mijenja povremenim pojedinačnim ukrasima. Neovisno o metodi koja se koristi, u svojoj temeljnoj strukturi sve metode su iste. Tipično metodologija sadrži se od: pred obrade, faze segmentacije (koja se može izbjeći ako se koriste globalne značajke riječi), prepoznavanja i post-obrade.[1]



Slika 2. Analitička metoda prepoznavanje riječi s pred-segmentacijskim točkama i odgovarajućim dijagramom prijelaza stanja.[1]

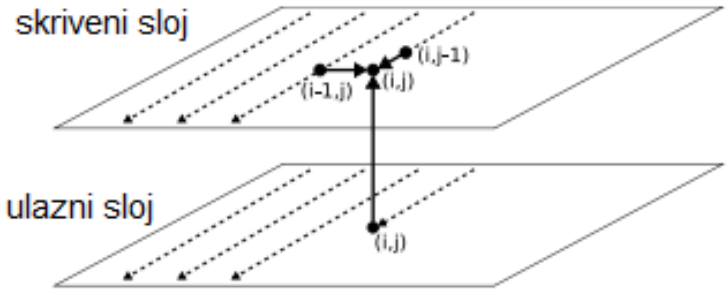
3.1. Offline prepoznavanje rukopisa s višedimenzionalnim rekurentnim neuronskim mrežama

Glavni nedostatak prepoznavanja rukopisa je da ulazne značajke moraju zadovoljiti stroge pretpostavke neovisnosti koje nameću HMM. Drugim riječima, to znači da značajke moraju biti redizajnirane za svaku abecedu i, u manjoj mjeri, za svaki jezik, te bi bilo nemoguće koristiti isti sistem za prepoznavanje Engleskog i Arapskog jezika.[2] Također, HMM-ovi pretpostavljaju da vjerojatnost svakog promatranja ovisi samo o trenutnom stanju, što otežava modeliranje kontekstualnih učinaka. Rekurentne neuronske mreže (eng. *recurrent neural networks - RNN*) ne pate od tih ograničenja i stoga se čine kao dobra alternativa HMM-ovima, međutim, kod prepoznavanje rukopisa, RNN-ovi su ograničene na izolirano prepoznavanje znakova [3] Rješenje ovog problema nude višedimenzionalne rekurentne neuronske mreže (eng. *multidimensional recurrent neural networks – MDRNNs*). MDRNNs su poseban slučaj

usmjerenih acikličkih grafičkih mreža. Generaliziraju standardne rekurentne neuronske mreže pružajući rekurentne veze duž svih prostorno-vremenskih dimenzija prisutnih u podacima. Ove veze čine MDRNNs robusnim prema lokalnim distorzijama duž bilo koje kombinacije ulaznih dimenzija i omogućuju im da modeliraju višedimenzionalni kontekst na fleksibilan način.[2]

Također, potrebno je pretvoriti dvodimenzionalnu sliku u jednodimenzionalni sekvencu oznaka. Rješenje tome je prosljeđivanje podataka kroz hijerarhiju MDRNN slojeva sa skupljenim blokovima aktivacija nakon svake razine. Pomoću tih blokova dvodimenzionalne slike se postupno kolabiraju u jednodimenzionalne sekvence, koje zatim mogu biti označene izlaznim slojem.[2]

Osnovna ideja MDRNNs je da zamijeni jedinstvenu rekurentnu vezu koja se nalazi u standardnim rekurentnim mrežama s toliko veza koliko ima prostorno-vremenskih dimenzija u podacima te tako omogući mreži da stvori fleksibilan unutarnji prikaz okolnog konteksta, koji je robustan prema lokaliziranim distorzijama. Skriveni sloj MDRNN-a skenira kroz unos u jednodimenzionalnim trakama, te pohranjuje svoje aktivacije u među spremnik. Trake su uređene na način da svaka točka koju je sloj već posjetio pokazuje na točku jedan korak unatrag po svakoj dimenziji. Skrivene aktivacije s prethodnih točaka se unose u trenutnu točku kroz rekurentne veze, zajedno s unosom. Jedan takav sloj dovoljan je da omogući pristup svim kontekstima u odnosu na smjer skeniranja od trenutne točke. Slika 2 prikazuje dvodimenzionalni MDRNN koji je upravo opisan. Debele linije prikazuju veze s trenutnom točkom (i, j). Veze unutar ravnine skrivenog sloja se ponavljaju. Veze unutar ravnine skrivenog sloja su rekurentne. Isprekidane linije prikazuju trake za skeniranje duž kojih su prethodne točke posjećivane, počevši u gornjem lijevom kutu.[2]



Slika 3. Dvodimenzionalna MDRNN[2]

3.2. Primjena offline metode

Primjena offline metode javlja se u obliku čitanja poštanske adrese, bankovnim čekovima i obrascima. Zadatak tumačenja ručno napisanih adresa je dodjeljivanje slike pošiljke adresi za isporuku. Pošiljci treba odrediti državu, grad, poštu, ulicu, kućni broj, te naziv tvrtke ili ime osobe. Ovaj sustav koristi znanje o poštanskim domenama kod prepoznavanja ručno napisanih adresa, te ga koristi poštanska služba SAD-a. Ovaj zadatak smatra se tumačenjem, a ne prepoznavanje, iz razloga što je zadatak ovog sistema dodijeliti adresu na ispravno odredište bez obzira na nepotpune ili kontradiktorne informacije prisutne na pošiljci.[1]

Priznavanje bankovnih čekova predstavlja nekoliko istraživačkih izazova u području analize i prepoznavanja dokumenata iz razloga što su pozadine čekova često obojene i imaju kompleksne uzorke. Iz čeka je potrebno iščitati: zakonski iznos, uslužnost, datum i potpis osobe koja je izdala ček. Analizom se slika filtrira i binarizira, segmentiraju se tekstualni blokovi, te se uklanjaju smjernice i zrnatost. Za prepoznavanje zakonskih i uslužnih iznosa koriste se HMM.[1]

4. Online prepoznavanje rukopisa

Online prepoznavanje rukopisa se odnosi na metode i tehnike koje se bave automatskom obradom poruke u trenutku kada je ona pisana uz pomoć digitalizatora ili instrumentalne olovke koja bilježi informacije o vršku olovke, općenito njegov položaj, brzinu ili ubrzanje kao funkciju vremena. Neke od tehničkih domena koje su se razvile za online prepoznavanje su: računala s olovkom (eng. *pen-based computers*), verifikatori potpisa i razvojni alati. Računala s olovkama se koriste za prepoznavanje rukom pisanih poruka i naredbi za interakciju s računalnim platformama, verifikatori potpisa se bave potpisima, vrlo specifičnom vrstom dobro naučenog rukopisa, u svrhu provjere identiteta osobe, dok razvojni alati uključuju različite sustave koji iskorištavaju neuromotorne karakteristike rukopisa za projektiranje sustava za potrebe obrazovanja i rehabilitacije.[1] Veliki interes za online prepoznavanje rukopisa javio se napredovanjem hardvera koji je omogućio unos i ispis na istu površinu kombiniranjem tableta i ravnih zaslona. Ova kombinacija omogućuje korištenje elektroničke tinte, pružajući trenutnu povratnu informaciju piscu. Elektronska tinta je prikaz traga kretanja vrha olovke izravno ispod vrha olovke. Također, interesu je pridonijelo želja za automatizacijom uredskog poslovanja za prirodnije metode unosa podataka u strojeve, te jasnije razumijevanje aplikacija koje su prikladne za prepoznavanje rukopisa.[5]

4.1. Tehnologija digitalizatora

Tabletni digitalizatori prvi puta su dokumentirani 1957. godine od strane T.L. Dimonda pod nazivom „Stylator“ u dijelu „Devices for reading handwritten characters“[6], međutim, RAND tablet je bio najpopularniji rani digitalizator i potaknuo je početnu aktivnost u prepoznavanju rukopisa na internetu. Dvije glavne tehnologije za digitalizator tableta su elektromagnetska ili elektrostatička i osjetljiva na pritisak. Elektromagnetski ili elektrostatički tableti imaju x i y rešetke vodiča u tabletu i petlju žica u vrhu olovke. Položaj vrha olovke određuje se na način da se ili mreža ili petlja pobuđuju elektromagnetskim impulsom, a druga detektira inducirani napon ili struju u sinusnom signalu. Vodiči tableta se skeniraju kako bi locirao par koji je najbliži petlji i vrši se interpolacija kako bi se odredio točan položaj između ova dva vodiča. Tableti osjetljivi na pritisak imaju slojeve vodljivog i otpornog materijala s mehaničkim razmakom između slojeva. Električni potencijal se primjenjuje preko jednog od otpornih slojeva, bilo u smjeru x ili y, kako bi se postavio gradijent napona koji odgovara položaju. Pritisak na vršku olovke rezultira time da vodljivi sloj izbacuje napon, a pri tome i

položaj, otpornog sloja. Tehnologija osjetljiva na pritisak ima prednost što ne zahtijeva uporabu posebnog olovke.[5]

4.2. Pred obrada

Pred obrada uzoraka izravno olakšava opis uzoraka i utječe na kvalitetu opisa. Zadaci pred obrade za online prepoznavanje uključuju: vanjsku segmentaciju, uklanjanje zrnatosti (eng. *noise reduction*) i normalizaciju znakova.[9]

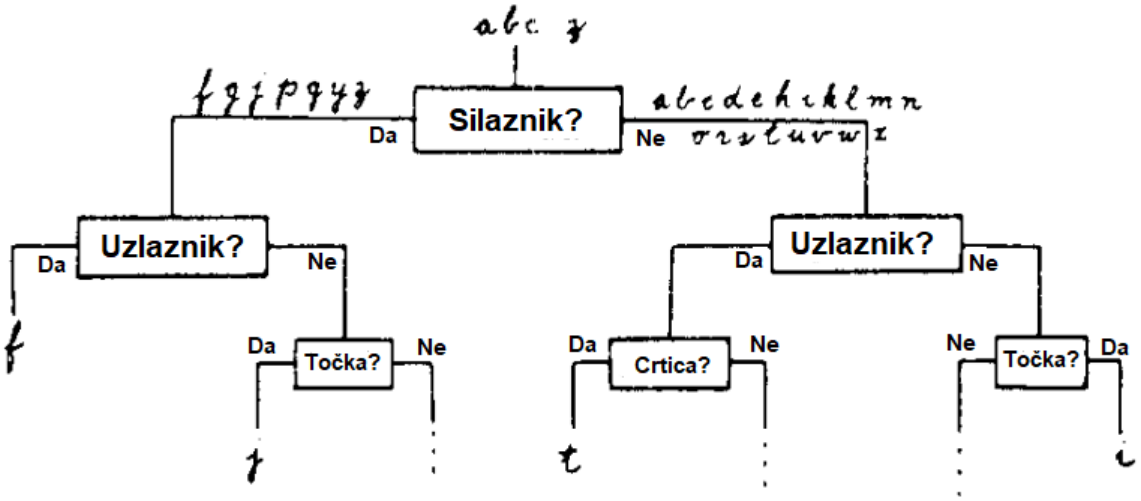
Vanjska segmentacija je izolacija raznih jedinica za pisanje kao što su znakovi ili riječi, prije samog prepoznavanja. U usporedbi s oslanjanjem na prepoznavanje za pružanje svih odluka o segmentaciji (kao što se to radi kod unutarnje segmentacije), vanjska segmentacija osigurava veću interaktivnost, uštedu izračuna i pojednostavljuje posao prepoznavatelja. Uklanjanje zrnatosti može se postići preko mnogih tehnika i algoritma. Zrnatost nastaje zbog ograničene točnosti tableta, procesa digitalizacije, neredovitog pokreta ruke i netočnosti indikacije dodira olovke (npr. ako slučajno dotaknemo površinu olovkom). Neke od tehnika uklanjanja zrnatosti su: zaglađivanje, filtriranje (ponekad nazivano prorjeđivanje, uklanja duplicirane podatkovne točke i smanjuje broj točaka), uklanjanje lažnih točaka (eng. *wild point correction,* zamjenjuje ili uklanja povremene lažne točke obično uzrokovane hardverskim problemima) i smanjenje točke (smanjuje točkice na pojedinačne točke). Normalizacija je proces koji mijenja raspon vrijednosti intenziteta piksela. Normalizacija putanje znakova na standardnu veličinu usvojena je u gotovo svakom sustavu prepoznavanja znakova. Normalizacija veličine prilagođava veličinu znaka tako da je znak zatvoren u standardni okvir. Normalizacija duljine udara prisiljava broj točaka na određeni broj. Nelinearna normalizacija dodjeljuje koordinate udarnim točkama prema raspodjeli gustoće linija sa ciljem izjednačavanja razmaka između poteza.[9]

4.3. Prepoznavanje oblika

Neke metode prepoznavanja oblika oslanjaju se na prethodnu analizu znakova abecede. Značajke (npr. kvačice) mogu biti specifične za abecedu. Druge metode, kao što je većina onih za obradu signala, u biti su neovisne o abecedi.[5]

4.3.1. Analiza značajki

Skup značajki može predstavljati znak. Značajke se mogu temeljiti na statičkim svojstvima znakova, dinamičkim svojstvima ili oboje. Značajke mogu biti binarne, npr. da li je znak silazan (eng. descender) ili ne, ima li točku ili je bez točke, itd. Kod binarnih značajki, prepoznavanje se često određuje stablom odlučivanja. Na primjer, za mala slova engleske abecede, prisutnost silaznika smanjuje izbore na f, g, j, p, q, y. Zatim, ako je točka prisutna, jedini izbor je j. Nedostatak ove metode je da ne može proizvesti alternativne izbore znakova, koji su obično poželjni za naknadnu obradu.[5]



Slika 4. Stablo odlučivanja kod binarnih značajki[5]

4.3.2. Vremenski slijed zona, ekstrema ili smjerova

Te se metode prvenstveno oslanjaju na dinamičke informacije. Niz kodiranih zona može predstavljati znak. Zone se određuju dijeljenjem pravokutnika koji okružuje pisani znak. Znak se postavlja na pravokutnik i određuje se niz zona koje prelazi vrh olovke. Ovaj slijed, ili odgovarajući slijed značajki, zatim dodjeljuje ime nepoznatom liku, često točnim podudaranjem iz rječnika sekvenci zona.[5]

Metoda ekstrema opisuje znak u smislu niza točaka lokalnih ekstrema (obično lijevo, desno, gore, dolje). Takvi se sljedovi nazivaju lančani kodovi (eng. *chain codes*).[5]

Slična metoda koristi niz smjerova kretanja vrha olovke tijekom pisanja određenog znaka. Koristeći četiri primitivna smjera (gore, dolje, lijevo, desno), sustav kodira prva četiri smjera sekvence, a zatim pomoću naredbe table lookup klasificira traženje po tablici koja ima 256 ($4 * 4 * 4 * 4$) unosa. Kako se broj smjerova i vremenskih intervala povećava, table lookup postaje manje praktičan, a sekvence se uspoređuju pomoću podudaranja krivulja (eng. *curve matching*).[5]

4.3.3. Podudaranje krivulja

Podudaranje krivulja je problem pronalaženja najboljeg uklapanja između dvije krivulje, te je izrazito važan u aplikacijama računalnog vida. Ovaj se problem pojavljuje u raznim montažama koji zahtijevaju da robot stavi dva dijela zajedno duž njihove međusobne granice. Podudaranje krivulja može se poistovjetiti sa slaganjem slagalica, jer računalo mora spojiti dvije krivulje koje su međusobno povezive i zajedno čine znak. Budući da su dvodimenzionalni objekti u potpunosti opisani (globalno i lokalno) svojim zatvorenim graničnim krivuljama, detekcija djelomično zatvorenih objekata koji djeluju u složenoj sceni može se izvesti usklađivanjem granične krivulje scene s graničnim krivuljama kandidata, te pokušati saznati koja kova krivulja kandidata odgovara graničnoj krivulji.[7]

Schwartz i Sharir, u znanstvenom radu „Identification of partially obscured objects in two dimensions by matching of noisy 'characteristic curves'“[8], riješili su problem podudaranja krivulje pod restriktivnom pretpostavkom da je jedna krivulja koja se podudara ispravna subkrivulja druge. "S obzirom na dvije krivulje, tako da je jedna prikladna subkrivulja druge, pronađite translaciju i rotaciju subkrivulje koje daju svoje najbolje najmanje kvadrate prikladne za dulju krivulju."[7][8]

4.3.4. Prepoznavanje nizova znakova i pisanih slova

Za neke probleme se segmentacija znakova ne može izvesti bez prepoznavanju znakova. Primjer ovog problema je engleski run-on printing na kojem se susjedni likovi mogu dodirnuti ili se preklapati. Sličan se problem pojavljuje kod kineskih znakova gdje segment može biti znak ili dio većeg znaka. Jedan od načina rješavanja ovog problema je segmentiranje i povezivanje svih jedinica pisanja koje mogu biti znakovi, a zatim rangiranje nizova znakova prema njihovim kumulativnim rezultatima prepoznavanja oblika.[5]

Ova metoda je također primijenjena za prepoznavanje pisanih slova. Prepoznavanje pisanih slova je teško iz razloga što se nekoliko znakova može napisati jednim potezom, a segmentacija počinje tek nakon što potez završi, odnosno, kada se olovka digne od površine na kojoj se piše. Uobičajeni pristup je analiza riječi pomoću segmenta poteza, gdje je segment poteza dio poteza ili cijeli potez (npr. ako je riječ o samo jednom slovu). Zatim se za identifikaciju slova koriste sekvence segmenata poteza. Segment poteza se može gledati kao putanja koja proizlazi iz jednog mišićnog djelovanja. [5]

4.4. Naknadna obrada

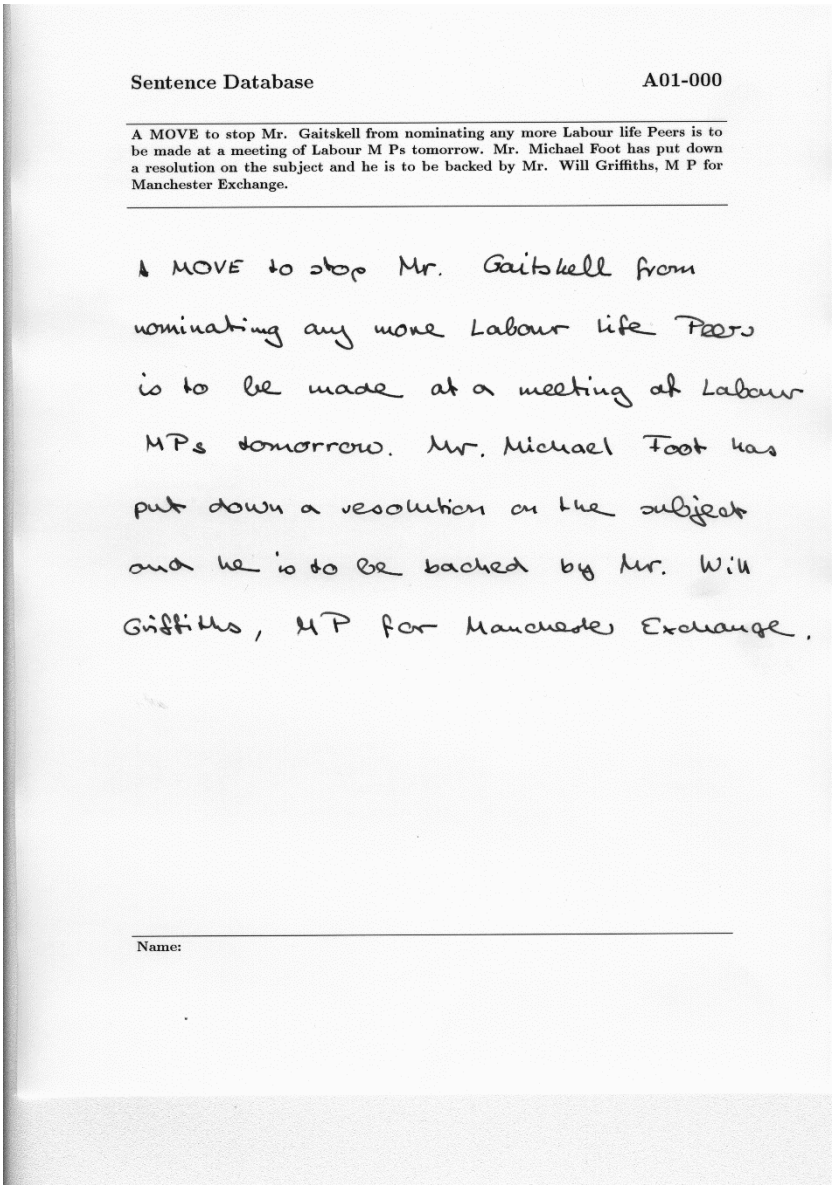
Faza naknadne obrade posljednja je faza sustava prepoznavanja. U strukturiranom tekstualnom obliku ispisuje odgovarajuće prepoznate znakove. Rezultati sustava obično sadrže pogreške zbog problema s klasifikacijom znakova i segmentacijom. Za ispravljanje pogrešaka pri prepoznavanju, sustavi za prepoznavanje rukopisa primjenjuju kontekstualne tehnike naknadne obrade. Dvije najčešće metode naknadne obrade za ispravljanje pogrešaka su pretraživanje rječnika i statistički pristup. Prednost statističkog pristupa u odnosu na pretraživanje rječnika je računsko vrijeme i korištenje memorije. Najjednostavniji način uključivanja informacija o kontekstu je korištenje rječnika za ispravljanje manjih pogrešaka. [10]

5. Aplikacija za prepoznavanje rukopisa i određivanja autora

Praktični dio rada je aplikacija koja je u mogućnosti prepoznati ručno napisani tekst, te odrediti autora ručno napisanog teksta u koliko se on nalazi u bazi. U aplikaciju su implementirane dvije neuronske mreže, istrenirana SimpleHTR neuronska mreža za prepoznavanje ručno napisanog teksta[11], te neuronska mreža za određivanje autora koju sam sam istrenirao uz pomoć preuzetog koda na githubu [12].

5.1. IAM Handwriting Database

Za izradu praktičnog dijela rada korištena je IAM Handwriting Database baza podataka [13]. IAM Handwriting Database sastoji se od forma ručno pisanog engleskog teksta koji se mogu upotrijebiti za treniranje i testiranje prepoznavanja rukopisa i za obavljanje eksperimenata identifikacije i provjere pisaca. Baza podataka sadrži ručno ispunjene forme, koje su skenirane u rezoluciji 300 dpi i spremljene u .png slike sa 256 razina sive boje (eng. *gray levels*). Ova baza podataka sastoji se od: 657 rukopisa pisaca, 1539 stranica skeniranog teksta, 5685 izoliranih i označenih rečenica, 13353 izoliranih i označenih linija teksta te 115320 izoliranih i označenih riječi. Sve slike forma, rečenica i riječi nalaze se u obliku .png datoteka, a odgovarajuće datoteke s oznakama formi, uključujući podatke o segmentaciji, nalaze se u nazivu slika i mapa kao meta-informacije u XML formatu, te je vrlo lako saznati koja rečenica ili riječ pripada kojoj formi. Slika 5 prikazuje jednu od ispunjenih formi, a slika 6. prikazuje izoliranu 1. rečenicu iz te forme. Za treniranje neuronske mreže korišteno je 65 formi i 708 redova teksta označenih s a01, a02, b05 i b06.



Sentence Database A01-000

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, M P for Manchester Exchange.

A MOVE to stop Mr. Gaitskell from nominating any more Labour life Peers is to be made at a meeting of Labour MPs tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, MP for Manchester Exchange.

Name:

Slika 5. Primjer ispunjene forme [13]



A MOVE to stop Mr. Gaitskell from

Slika 6. Primjer izoliranog reda[13]

5.2. Korištene tehnologije i alati

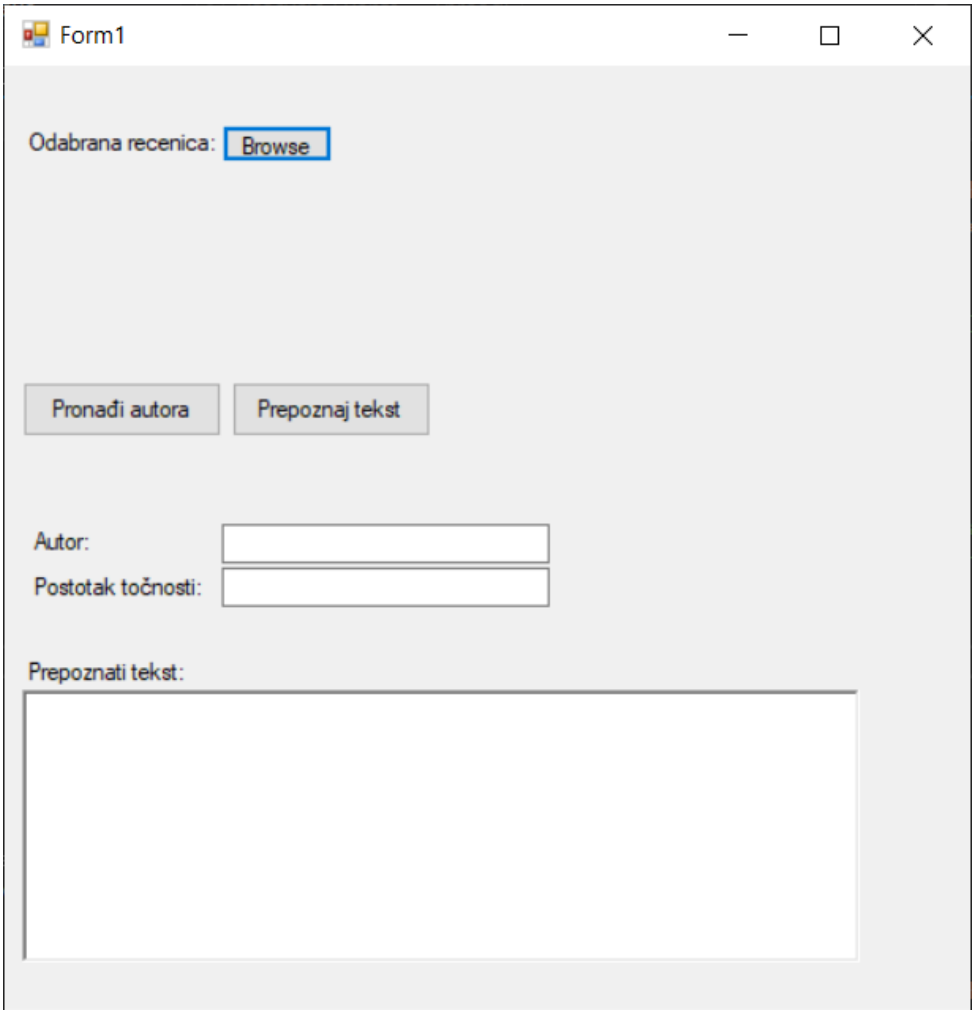
Kao što je već prije navedeno, izrada praktičnog dijela rada izrađena je pomoću simpleHTR neuronske mreže koja je već unaprijed bila istrenirana i neuronske mreže za prepoznavanje autora koju sam sam istrenirao. Neuronske mreže implementirane su pomoću TensorFlow-a, a za učenje neuronske mreže korišten je Keras. SimpleHTR neuronska mreža trenirana je na podacima iz IAM Handwriting Database-a koji su također korišteni kod treniranja neuronske mreže za prepoznavanje autora. Aplikacija koristi dio koda preuzetog od Github korisnika Groovypanda pomoću kojeg sliku rečenice rastavi na pojedine riječi te sliku svake pojedine riječi spremi u mapu[14]. Ovaj dio je bitan iz razloga što je SimpleHTR neuronska mreža istrenirana da prepoznaje samo jednu riječ a ne cijelu rečenicu.

Kako bi se smanjilo vrijeme potrebno za treniranje neuronske mreže, korišten je Google Colab koji pruža uslugu korištenja Tesla K80 grafičke kartice koja je smanjila vrijeme učenja sa otprilike 60 sati na otprilike 8 sati. Google Colab je besplatan za korištenje, sve što je potrebno je prenijeti python notebook na Google Drive te na početku koda povezati Google Drive s Google Colab-om u prvoj liniji notebook-a.

Za izradu grafičkog sučelja aplikacije korišten je Visual Studio 2019 i programski jezik C#. Izrađena je windows forms aplikacija koja omogućava običnom korisniku da koristi aplikaciju bez ikakvog razumijevanja ijednog programskog jezika, te nudi korisniku vrlo dobro iskustvo korištenja.

5.3. Prikaz aplikacije

Kao što je ranije navedeno, aplikacija za prepoznavanje rukopisa i određivanje autora izrađena je u alatu Visual Studio 2019. Aplikacija je vrlo jednostavna za korištenje i sastoji se od jednog glavnog izbornika na kojem su 3 gumba, 3 textbox-a za prikaz rezultata i jedne panele za prikaz odabrane slike. Kada se aplikacija pokrene izgleda kao na slici 7. Klikom na „Browse“ otvorit će se novi prozor preko kojeg korisnik odabire sliku koju želi prepoznati. Slika mora biti isključivo u .png formatu kako bi se mogla odabrati.

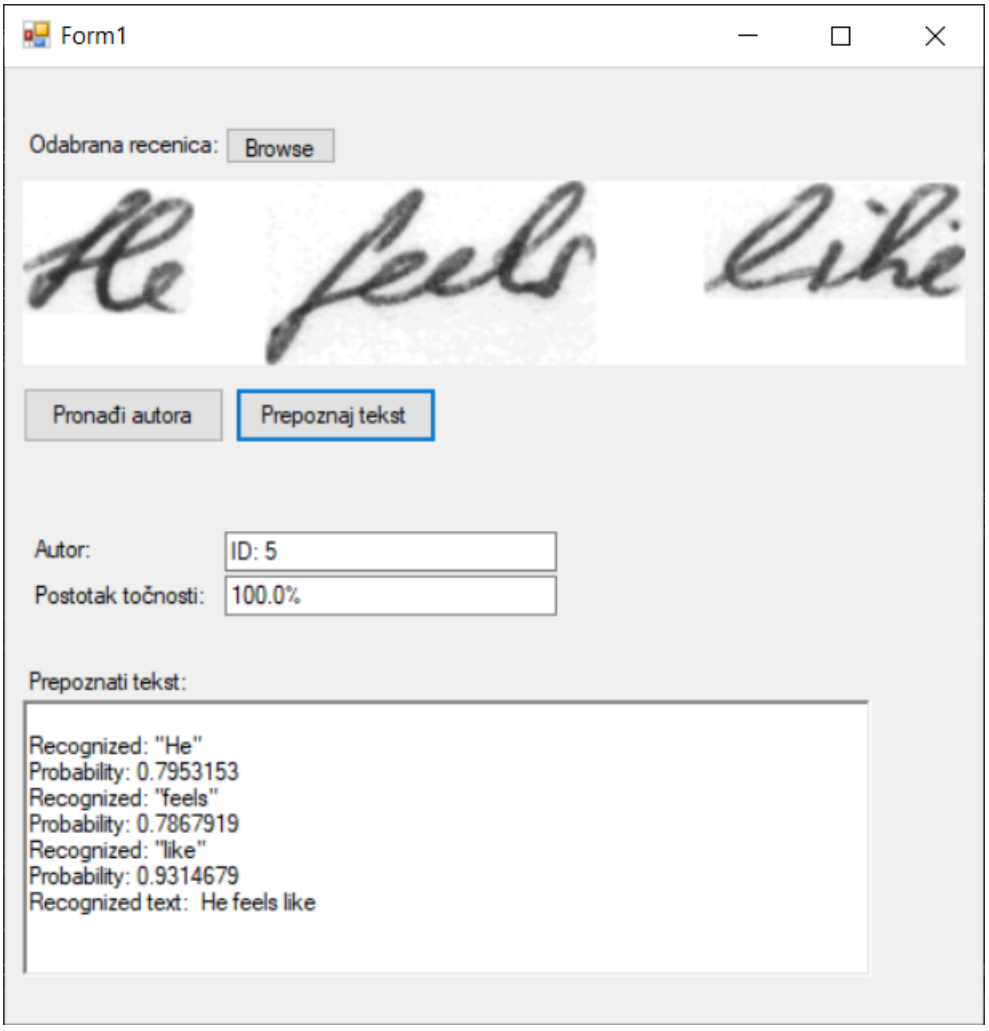


Slika 7. Aplikacija prilikom pokretanja

Nakon što korisnik odabere sliku, ta slika će se prikazati ispod gumba „Browse“, kao što je prikazano na slici 8, kako bi korisnik moga sam sebi potvrditi da je odabrao dobru sliku. Nakon toga korisnik ima dvije opcije: pronađi autora ili prepoznati tekst.

Klikom na gumb pronađi autora aplikacija će pokušati pronaći autora na temelju autora korištenih u treniranju te će ispisati prepoznatog autora i postotak sigurnosti da je autor pogođen. Ukoliko postotak točnosti bude 100% to znači da je odabrana slika rečenice bila korištena prilikom treniranja neuronske mreže.

Klikom na gumb prepoznaj tekst aplikacija će pokušati prepoznati napisani tekst te ispisati postotak sigurnosti da je svaka pojedina riječ točno prepoznata i na kraju ispisati cijelu rečenicu. Na slici 8. se može vidjeti primjer točno prepoznatog teksta i autora.



Slika 8. Aplikacija s prikazom rezultata

5.3.1. Poznati problemi kod aplikacije

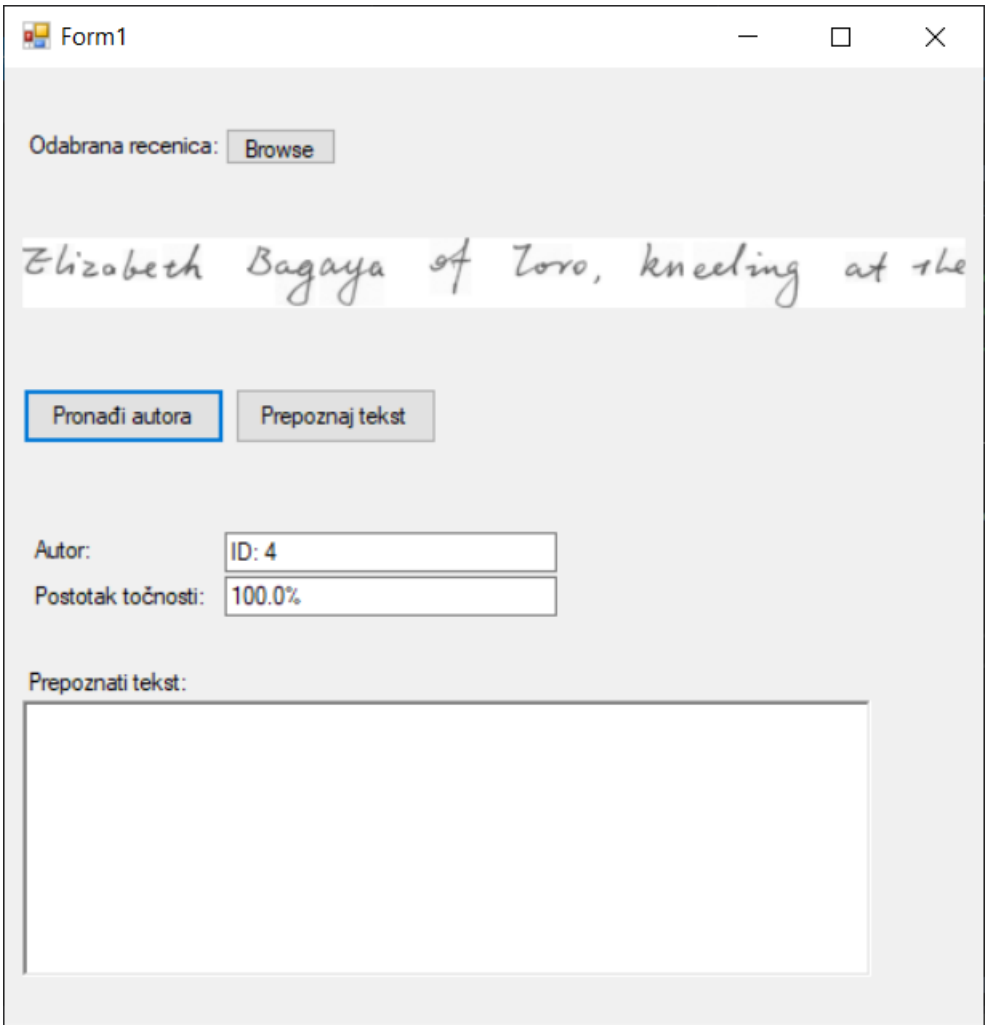
Opcija pronađi autora će raditi samo u slučaju ako je autor korišten kod treniranja neuronske mreže, te ukoliko korisnici žele da aplikacija prepoznaje njihov rukopis morat će ispočetka istrenirati neuronsku mrežu te prilikom treniranja priložiti svoj rukopis.

U koliko korisnici žele prepoznati tekst koji su sami napisali morat će ga skenirati i prilagoditi slikama s IAM Handwriting Database-a obzirom da su svi podaci kod treniranja neuronske mreže preuzeti iz te baze podataka. To znači da slike moraju biti skenirane u rezoluciji 300 dpi i spremljene isključivo u .png formatu sa 256 razina sive boje. Također slike moraju biti obrezane kao na slici 6, tako da slika bude samo rečenica koju korisnici žele prepoznati. Također, aplikacija samo prepoznaje engleske riječi.

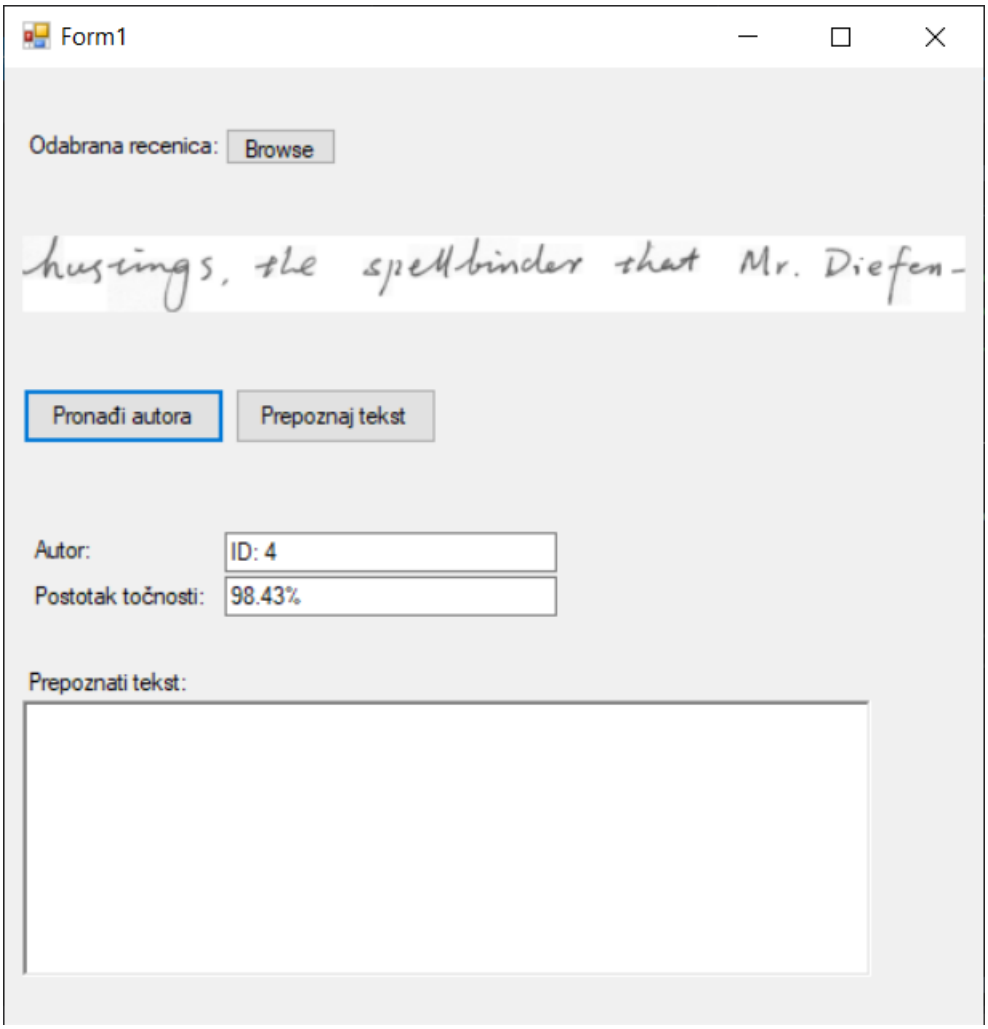
5.4. Testiranje neuronske mreže

Kod treniranja neuronske mreže skup podataka se uobičajeno podijeli na tri dijela: podaci za treniranje, podaci za validaciju i podaci za testiranje. Podaci za treniranje se koriste kod treniranja neuronske mreže, podaci da validaciju se koriste za provjeru točnosti mreže tijekom treniranja i podaci za testiranje koriste se za provjeravanje točnosti gotove neuronske mreže.

Kao što je navedeno ranije, za treniranje neuronske mreže za prepoznavanje autora, koristile su se forme s oznakama a01, a02, b05 i b06. Kako bi se provjerila točnost neuronske mreže, treba se uzeti rečenica iz forme koja nije korištena u treniranju i da pripada nekom od autora iz oznaka korištenih za treniranje. Forma a02-102 i forma a03-047 pripadaju istom autoru, što se lako može vidjeti jer se autor potpisao na obje forme. Kako forme s oznakom a03 nisu korištene za treniranje potrebno je uzeti jedan red teksta iz te forme i provjeriti da li će aplikacija prepoznati autora. Slika 9 prikazuje red teksta iz forme a02-102. Može se vidjeti da autor tog reda teksta ima dodijeljen ID 4. Slika 10. prikazuje red teksta iz forme a03-047. Može se vidjeti da je za taj red teksta isto prepoznat autor koji ima ID 4, te da je aplikacija 98.43% sigurna da se radi o istom autoru.



Slika 9. Autor forme a02-102



Slika 10. Autor forme a03-047

6. Zaključak

U radu su opisane dvije metode prepoznavanja rukopisa, online i offline metoda, način na koji funkcioniraju, njihova primjena i načini na koji se mogu implementirati. Napravljena je funkcionalna aplikacija koja može prepoznati ručno napisani tekst offline metodom te prepoznati autora napisanog teksta ukoliko je taj autor sudjelovao u procesu treniranja neuronske mreže. Neuronska mreža je korištena za prepoznavanje teksta i autora iz razloga što ima veliku funkcionalnost, a zbog raznih modula i alata jednostavna je za korištenje.

Neuronska mreža iz praktičnog dijela ima mjesta za poboljšanje. Svaka osoba ima jedinstven rukopis, te neuronska mreža nikada neće raditi sa 100% točnošću, međutim trebala bi imati što veću moguću točnost iz razloga što u njenoj primjeni u stvarnom svijetu greške nisu poželjne. Kako bi se točnost neuronske mreže povećala potrebno je trenirati neuronsku mrežu sa što više različitih primjera, odnosno različitih rukopisa.

U današnje vrijeme prepoznavanje rukopisa ima neke uporabe u stvarnom svijetu, kao što je spomenuto i u radu, međutim ima još primjena npr. u vještačenju, zabavi, itd. Obzirom da je rukopis jedinstven za svaku osobu, prepoznavanje rukopisa ima potencijalno beskonačno prostora za poboljšanje, te radi toga vjerojatno neće nikada u potpunosti zamijeniti čovjeka u stvarnom svijetu kod obavljanje zadatka prepoznavanja rukopisa, pogotovo u poslovima gdje nema nikakvog mjesta za greške.

Popis literature

[1] R. Plamondon, S. N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey", Ivizlab.sfu.ca, 2000. [Na internetu]. Dostupno: http://ivizlab.sfu.ca/arya/Papers/IEEE/PAMI/2000/January/Handwriting%20Recognition.pdf [Pristupano: 27. 5. 2019].

[2] A. Graves, J. Schmidhuber, “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks”, 2008. [Na internetu]. Dostupno: http://papers.nips.cc/paper/3449-offline-handwriting-recognition-with-multidimensional-recurrent-neural-networks.pdf [Pristupano: 27.5.2019].

[3] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition", Idsia.ch, 2009. [Na internetu]. Dostupno: http://www.idsia.ch/~juergen/tpami_2008.pdf [Pristupano: 10.6. 2019].

[4] G. Tímár, K. Karacs, C. Rekeczky, "Analogic Preprocessing And Segmentation Algorithms For Offline Handwriting Recognition", *Semanticscholar.org*, 2003. [Na internetu]. Dostupno: https://www.semanticscholar.org/paper/Analogic-Preprocessing-And-Segmentation-Algorithms-T%C3%ADm%C3%A1r-Karacs/5868707928fec9f894df6017c0611a734c8f7eca [Pristupano: 10. 6. 2019].

[5] C. C. Tappert, C. Y. Suen, T. Wakahara, "The State of the Art in On-Line Handwriting Recognition", *Csis.pace.edu*, 1990. [Na internetu]. Dostupno: http://www.csis.pace.edu/~ctappert/dps/pdf/pen-tappert.pdf [Pristupano: 13. 6. 2019].

[6] T. Dimond, "Devices for reading handwritten characters", *dl.acm.org*, 1957. [Na internetu]. Dostupno: https://dl.acm.org/citation.cfm?id=1457765 [Pristupano: 14. 6. 2019].

[7] H. Wolfson, "On curve matching", *dl.acm.org*, 1990. [Na internetu]. Dostupno: https://dl.acm.org/citation.cfm?id=81059 [Pristupano: 19. 6. 2019].

[8] J. Schwartz, M. Sharir, "Identification of partially obscured objects in two dimensions by matching of noisy 'characteristic curves'", *archive.org*, 1985. [Na internetu]. Dostupno: https://archive.org/details/identificationof00schw [Pristupano: 20. 6. 2019].

[9] V. Kumar, „Online Handwriting Recognition Problem: Issues and Techniques“, *Pdfs.semanticscholar.org*, 2014. [Na internetu] Dostupno: https://pdfs.semanticscholar.org/f957/3acd8405b5c594314491dedfbeb3bf40750f.pdf [Pristupano: 7. 9. 2019].

[10] Jaipur National University, „A Review: English Online Handwriting Recognition“, *Ijana.in.* bez. dat. [Na internetu] Dostupno: http://www.ijana.in/Special%20Issue/C29.pdf [Pristupano: 8. 9. 2019].

[11] SimpleHTR | githubherald | Github.com [Na internetu] Dostupno: https://github.com/githubharald/SimpleHTR [Pristupano: 10. 9. 2019.]

[12] P. Dwivedi, „Handwriting recognition using Tensorflow and Keras“, *towardsdatascience.com*, 2018. [Na internetu] Dostupno: https://towardsdatascience.com/handwriting-recognition-using-tensorflow-and-keras-819b36148fe5 [Pristupano: 2. 6. 2019.]

[13] U. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition", *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39-46, 2002. Dostupno: http://www.fki.inf.unibe.ch/databases/iam-handwriting-database [Pristupano: 2. 6. 2019.]

[14] Handwriting_Recognition | Groovypanda | Github.com [Na internetu] Dostupno: https://github.com/Groovypanda/Handwriting_Recognition?fbclid=IwAR1WeP3ZjkQhQmtzqXCKRReEIR62dSRgO1QKbdtx01qb4x1bIJNiYjtGCvo [Pristupano: 11. 9. 2019.]

Popis slika

Slika 1: Blok dijagram offline sustava prepoznavanja rukopisa[4] 6
Slika 2. Analitička metoda prepoznavanje riječi s pred-segmentacijskim točkama i odgovarajućim dijagramom prijelaza stanja.[1].....7
Slika 3. Dvodimenzionalna MDRNN[2] 8
Slika 4. Stablo odlučivanja kod binarnih značajki[5].....12
Slika 5. Primjer ispunjene forme [13] 16
Slika 7. Aplikacija prilikom pokretanja.....18
Slika 8. Aplikacija s prikazom rezultata 19
Slika 9. Autor forme a02-102 21
Slika 10. Autor forme a03-047.....22