MARIO MIKŠIĆ

USPOREDBA ALGORITAMA ZA OCR PRI RAZLIČITIM REZOLUCIJAMA

DIPLOMSKI RAD

Zagreb, 2012
Optičko prepoznavanje znakova (OCR) je kombinacija mehaničke i elektoničke komunikacije za prijenos rukom pisanih ili otisnutih tekstova u računalni kod pomoću optičkih uređaja. Njegova povijest seže i sto godina unazad i do danas se razgranala tj. dio je jedne široke skupine optičkog prepoznavanja. Primjena njegovih mogućnosti dovela nas je do toga da nam se optičko prepoznavanje uvuklo u sve životne prostore kao što su prepoznavanje poslovnih kartica, interaktivne igre novog doba, osobna identifikacija itd., a najzaslužniji za to su sve bolji algoritmi. Oni omogućavaju sve bržu i precizniju obradu (unos, raščišćavanje i usporedbu) no uvijek postoji mogućnost razvoja pogotovo kombinacijom više algoritama. Bitna stavka je da pri prepoznavanju algoritmi zadrže mogućnost "gledanja" na sliku bez obzira na rezoluciju i transformaciju slike (translacija, kaliranje i rotacija) tj. da stvori slikovne invarijacije i da se prepoznavanje dovede do razine ljudskih osjetila.

**Ključne riječi:** Optičko prepoznavanje, slikovne invarijacije, rezolucija

**ABSTRACT**

Optical Character Recognition (OCR) is combination of mechanical and electronic communication for transmission of handwritten, typewritten or printed text into machine-encoded text using optical devices. It's history goes hundred years back and today has become part of the group of optical recognition devices. Using it's possibilities led us to point where we have optical recognition in every area of our lives such as the recognition of business cards, interactive games of new age, personal identification, etc., and most responsible for it's progress are algorithms. They allow faster and more accurate processing (input, clearing and comparison), but there is always possibility of more development, especially with the combination of two or more algorithms. The important fact is that the algorithms for recognition keep the possibility of "seeing" the image, regardless of image resolution and transformation (translation, scaling and rotation) and to create image invariants and to bring recognition to the level of human senses.

**Key words:** Optical recognition, Image invariants, moment, resolution
### SADRŽAJ

1. **UVOD** .......................................................................................................................................... 1

2. **OCR** ............................................................................................................................................. 2
   2.1. Što je OCR? .......................................................................................................................... 2
   2.2. Povijest OCR-a ................................................................................................................. 4
   2.3. Osnovni pojmovi .................................................................................................................. 8
   2.3.1. Pismo i njegove karakteristike .................................................................................. 8
   2.3.2. Uređaji za unos ............................................................................................................ 9
   2.3.3. Rezolucija slike .......................................................................................................... 10
   2.3.4. Postupak prepoznavanja .......................................................................................... 10
   2.3.5. OCR softver .............................................................................................................. 13

3. **SVOJSTVA SLIKE** ........................................................................................................... 15
   3.1. Predobrada slike ............................................................................................................. 15
   3.2. Usporedba oblika ............................................................................................................. 16
   3.3. Srodne transformacije .................................................................................................. 16
   3.4. Proces OCR-a .................................................................................................................. 17
   3.5. Komponente OCR sistema ........................................................................................... 18
   3.5.1. Skeniranje .................................................................................................................. 18
   3.5.2. Lociranje i segmentacija ......................................................................................... 19
   3.5.3. Predprocesuiranje .................................................................................................... 21
   3.5.4. Izvlačenje značajki .................................................................................................. 22
     3.5.4.1. Podudaranj predložaka i tehnika ................................................................... 23
     3.5.4.2. Metode na osnovi značajki ........................................................................... 24
   3.5.5. Klasifikacija ................................................................................................................ 26
     3.6.5.1. Metode teorijskog odlučivanja ................................................................. 27
     3.6.5.2. Strukturne metode ....................................................................................... 28

4. **SLIKOVNE INVARIJANTE** .......................................................................................... 31
   4.1. Invarijacije na osnovi granica ..................................................................................... 32
     4.1.1. Lančani kod .......................................................................................................... 32
     4.1.2. Fourierovi deskriptori ....................................................................................... 33
   4.2. Invarijacije na osnovi prostora .................................................................................... 35
     4.2.1. Regularni i centralni moment ........................................................................... 37
     4.2.2. Huvih sedam momenata .................................................................................. 39
1. UVOD

"Tečan sam u govoru preko šest milijuna različitih oblika komunikacija."

*Robot C-3PO "Zvjezdani ratovi VI: Povratak Jedija (1983.) 20th Century Fox*

Možda to zvuči kao znanstvena fantastika koja se pojavljivala u maštama crtača stripova i filmskih scenarista, ali odavnina težimo razumijevanju svega što je oko nas i kako god komunicirali da ne postoje jezične barijere, stoga smo s vremenom, uvidjevši mogućnosti sve jačih računala, počeli stvarati uređaje koji će naš san moći ostvariti. "Robot" i ne zvuči tako teško za sastaviti ali težnja da taj robot ima sposobnost biti čovjek, uređaj s umjetnom inteligencijom, e tu nastaje problemi, tj. više granice koje svaki dan pomičemo učeći i razvijajući tehnologiju te primjenjujući neke nove ideje i načela.

Među prvim problemima koji su se pojavili u stvaranju umjetne inteligencije bio je unos informacija iz okoline tj. kako stroj vidi svijet oko sebe i kako da najbolje kategorizira informacije koje prikupi. Iako u samim početcima optičko prepoznavanje nije imalo ni približnu namjenu, što ćemo vidjeti u povijesnom razvoju OCR-a, s vremenom je postala osnova računalnog gledanja našeg svijeta.

Gledajući danas, OCR se razgranao u različite oblike optičkog prepoznavanja te je našao primjenu u gotovo svim granama tehnologije, toliko je postao integriran u naš život da najčešće i ne primjećujemo da koristimo načela koja proizlaze iz njegovog razvoja, od svakodnevnih kupovine u dućanu, primjene osobnih iskaznica, parkirnih kartica, polaganja ispita do fotoaparata s prepoznavanjem osmijeha, čitača za slijepe osobe itd.

Ipak, koliko god primjenjujemo pomoć tehnika automatskog prepoznavanja i dalje ono nije svemoguće i nailazi uvijek na slične probleme koji se moraju istražiti, analizirati i pronaći riješenje. Jedan od njih je i "nečitkost" tj. niska kvaliteta unosa informacija. Upravo s jednim takvim problemom ćemo se susresti u ovom radu te ćemo ga pokušati razraditi i postaviti neke zaključke.
2. OCR

Optičko prepoznavanje znakova (eng. Optical Character Recognition - OCR) je računalno prepoznavanje tj. prevodenje slike pisanog ili otisnutog teksta, znakova, simbola i uzoraka s papira, digitaliziranog pomoću skenera, kamere ili fotoaparata, u tekst ili znakove za računalo prepoznatljiv oblik, ASCII ili Unicode kod kojih se može naknadno editirati. [4]

2.1. Što je OCR?

OCR je važan dio sustava tehnika automatskog prepoznavanja u koje još spada i prepoznavanje govera, radio frekvencija, vizualni sistem prepoznavanja oblika, magnetske trake, bar kodovi, magnetske tinte i optičko prepoznavanje oznaka. Cilj OCR sustava je prepoznavanje slova abecede, brojeva i drugih znakova iz digitalne slike bez našeg utjecaja. To se postiže tako da izdvojeno slovo na slici pretražimo i uspoređimo sa odgovarajućim predloškom slike znakova iz postojeće pohranjene baze modela. Samo računalo ne može provesti prepoznavanje bez popratnog softvera i hardvera te baš u tome se nalazi kvaliteta samog OCR-a. Kvalitetni softveri i algoritmi uvršteni u njih omogućavaju prebacivanje knjiga u oblik koji se može pohraniti na računalo bez da moramo ručno unositi podatke riječ po riječ. Sam proces oslanja se na obradu skeniranih slika koje se analiziraju na svijetla i tamna područja teksta znak po znak, u svrhu prepoznavanja svakog pojedinog slova ili broja. Kad je znak prepoznat dodjeljuje mu se ASCII kod u današnje vrijeme najčešće korišten kodni sustav u obradi podataka.

S vremenom je OCR dobio sve značajniju ulogu i sve više utjecao na način spremanja, editiranja i dijeljenja podataka. Najviše se koristi u knjižnicama pri digitalizaciji knjiga kako bi se očuvala vrijedna,pisana,povijesna baština. [27] U jednostavnjem obliku se primjenjuje kod identifikacije raznih bankovnih i podatkovnih kartica te pri sortiranju velikog broja pošiljki u poštama dodijeljujući im odgovarajuća odredišta uz minimalan broj pogrešaka. Koliko god OCR napreduje i usavršava se s vremenom još uvijek daje slabije rezultate u prepoznavanju rukom pisanog teksta ili tiskanog teksta koji sliči rukopisu. Ti napredniji softveri tražili su se najviše pri
prepoznavanju potpisa i upisanih novčanih vrijednosti kod skeniranja bankovnih čekova. Pošto su čekovi već nekako stvar prošlosti bolji bi primjer bio velika količina ručno ispisanih sudskih ili policijskih spisa koje bi bilo mnogo lakše pretraživati i kategorizirati u kvalitetno arhiviranom digitalnom obliku. Iako bi sam proces digitalizacije svih tih spisa bio dugotrajan i skup moramo gledati prednosti koje nam donosi kao što su brzina pretraživanja, manji prostor i kvaliteta skladištenja tj. pohranjivanja, umnožavanje samih podataka i jednostavnost proslijeđivanja i pristupanja ako treba i sa drugog kraja svijeta.[1][28][29]

Ovisno o tome kad se vrši obrada OCR se dijeli na "On-line" prepoznavanje i na "Off-line" prepoznavanje (1). "Off-line" prepoznavanje je obrada koja se vrši nakon što završimo unos kompletnog sadržaja dok kod "On-line" prepoznavanja obrada se obavlja trenutačno dok se pisani znakovi ispisuju. Sama kvaliteta izvedbe tog prepoznavanja izravno ovisi o kvaliteti ulaznih podataka tj. što je ujednačenija izvedba svakog pojedinog znaka kod rukom pisanog teksta to je prepoznavanje točnije. Sve u svemu tehnika prepoznavanja znakova je još uvijek daleko od načina na koji čovjek prepoznaje i čita. Jedina prednost koja se daje uređajima je brzina, no sve bržim napretkom tehnologije i konstantnim razvijanjem novih kompjuterskih jezika i algoritama, prepoznavanje se dovodi do idealnog.[8]

Slika 1: Podijela prepoznavanja znakova prema obradi
2.2. Povijest OCR-a

Prepoznavanje znakova (OCR) je podkategorija tehnike prepoznavanja uzoraka ali je samo prepoznavanje znakova bilo osnova na kojoj su se kasnije bazirali i razvili svi ostali sustavi automatskog prepoznavanja. Ako bi smo gledali gdje je sve to počelo mogli bi se vratiti do prvih pokušaja da se izradi uređaj koji oponaša ljudska osjetila u našem slučaju oko. Još prije 2 stoljeća pokušalo se doći do uređaja koji bi pomagali slijepima u čitanju. 1870-te amerikanac C. R. Carey je patentirao uređaj za prijenos slike koristeći mozaik fotoćelija (Skener mrežnice), dva desetljeća kasnije poljak P. G. Nipkow patentirao je sekvencialni skener kod kojeg se slika skenirala red po red što je bio velik pomak.[1][2]

Tjekom prvog dijela 20. stoljeća bilo je mnogo pokušaja da se izradi bolji uređaj za pomoć pri čitanju. Jedan od značajnijih pokušaja pripisuje se austrijanцу, Gustavu Tauscheku, samoukom inžinjeru iz Beča, koji 1928. patentira prvi uređaj s osnovnim konceptom današnjih OCR-a koji naziva "Uređaj za čitanje" (2). Tauschekovi patenti su se s vremenom sve više razvijala te je za ondašnji IBM patentirao 169 od kojih su najznačajniji uređaj za identifikaciju pomoću bušenih kartona i magnetski bubanj za pohranu podataka preteća današnjim tvrdim diskovima u računalima. [12][24]

Slika 2: Patent G.Tauschek – Uređaj za čitanje
Razvojem digitalnih kompjutera sredinom 1940-ih postavljaju se temelji OCR-a i pronalazi se sve veća primjena pogotovo u poslovnom svijetu. Od 1950-ih tehnologija zahvaća sve veći zamah, a elektronsko obrađivanje podataka postaje sve važnije područje. Unos podataka pomoću bušenih kartica postaje neekonomičan jer nastaje potreba za rukovanjem sa sve većom količinom podataka. U potrazi za boljim riješenjem vidi se mogućnost u OCR uređajima koji postaju dostupniji i dorasli tadašnjim zahtjevima. U to vrijeme David H. Shepard razvija i patentira sustav prepoznavanja podataka pisanih na pisačim mašinama (Gismo), a prvi takav uređaj je instaliran u Reader's Digest-u u to vrijeme jedana od jačih izdavačkih kuća. [6]

**Prva generacija**

Tijekom 60-ih se počinju vidjeti prednosti OCR-a te se počinje sve više komercijalizirati, ali raste i potreba za novim i bržim instalacijama u državnim sektorima. Ta prva generacija OCR uređaja oslanjala se samo na jedan oblik slova i to uz uvijet da nema varijacija u izgledu. Primjenu su našli u poštama pri prepoznavanju odredišnih adresa i u administraciji tj. identifikaciji kod očitavanja socijalnog broja. Znakovi su bili posebno dizajnirani i neprirodni no s vremenom su se počeli pojavljivati uređaji koji su mogli čitati i do desetak različitih fontova. Ograničenje su im stvarale male baze uzoraka za uspoređivanje, koje su morale sadržavati svako slovo i znak za pojedini font.

**Druga generacija**

Sve širom upotrebom u drugoj polovici 60-ih početkom 70-ih rastao je razvoj OCR uređaja koji počinju čitati i standardne otisnute fontove, a neki su čak mogli očitavati i rukom pisane znakove. Na tehnološkim sajmovima jedva se čekalo vidjeti nove uređaje i dostignuća, tako je na sajmu u New Yorku 1965. dojam ostavio "IBM 1287" razvijen u laboratoriju Rochester, prvi uređaj sposoban čitati rukom pisane brojeve. OCR uređaji druge generacije imali su ograničenja u prepoznavanju samo brojeva i nekoliko jasno rukom pisanih slova i simbola, ali i to je bio velik pomak. Nedugo zatim Toshiba proizvodi prvi automatizirani uređaj za sortiranje pošte po poštanskom broju, a Hitachi vidi priliku u masivnoj proizvodnji te izbacuje uređaje visokih performansi za malu cijenu. [30]
Američki institut za standardizaciju (American National Standards Institute – ANSI) 1968 standardizira tip fonta OCR-A (3) razvijen od strane ATF-a (American Type Founders) udruženja 23-ju ljevaonica metala, te postavlja kriterije dizajnirajući font čitljiv ljudskom oku i uređajima. S druge strane u europi Adrian Frutiger razvio je OCR-B za Europsku udrugu proizvođača računala (ECMA), europski font dizajniran je da bude prirodniji te je pokušano da se oba tipa ujedine ali s vremenom su uređaji počeli prepoznavati oba standarda. [2][19][20]

![OCR-A i OCR-B fonta](Slika 3: Izgled OCR-A i OCR-B fonta)

**Treća generacija**

Drugom polovicom 70-ih postavljaju se novi izazovi OCR-u, a to je čitanje nisko kvalitetnih ispisa, ispisa na malim i velikim površinama i što veći postotak točnosti rukom pisanih znakova. Također je bio cilj zadržati visoke performanse za što manju cijenu, a u korist im je išao sve brži i brži razvoj hardvera. Tržište je počelo tražiti specijalizirane OCR uređaje no razvojem računala i standardni uređaji su ponovo dobili primjenu pogotovo u pomoći slabovidnim i slijepim osobama ali su bili ograničeni na fontove na latinici. Razvojem tehnologije pojavili su se i novi tipovi pisača, laserski pisači mogli su otisnuti širi raspon fontova što je postavilo nove izazove u razvoju OCR-a, no amerikanac Raymnd Kurzwell 1974 nalazi rješenje i razvija prvi OCR sustav (omni-font OCR system), računalni program koji prepoznaje većine normalno otisnutih tekstova.

**Četvrta generacija**

OCR još 50-tih postaje komercijalan no broj prodanih primjeraka je bio vrlo malen isključivo radi cijene, prodano je samo nekoliko stotina primjeraka do 1986. Daljnjim razvojem tehnologije hardver postaje pristupačniji, a sam softver za OCR se poklanjao uz kupljene uređaje za skeniranje. Državne institucije i privatni razvojni
sektor zahtijevaju sve sofisticiranije uređaje da klasični OCR za prepoznavanje teksta postaje zanemariv već se prepoznavanje vrši pomoću bar kodova, magnetskih tinti, zvukova, skeniranjem prsta, dlana ili oka, hologramskih otisaka, UV otisaka itd. [2]

Današnja generacija

Metode prepoznavanja znakova su se kroz godine znatno poboljšala, kad se uzme u obzir da su početci bili samo prepoznavanje standardiziranih slova i brojeva, a današnje metode mogu razaznati razne rukom pisane tekstove i netipične znakove. Razvojem tehnologije i brzih računala, sve manjih i moćnijih procesora, mobilnih uređaja, visokorezolucijskih kamera, svijeta brzo dostupnih informacija, interneta na dlanu, bežičnosti, robota koji zamijenjuju čovjeka i još mnogi uređaja o kojima smo do samo prije par godina samo sanjali ili vidali u filmovima znanstvene fantastike razvijaju se i istražuju novi pristupi prepoznavanja.

Slika 4: Beyo OCR softver – direktni prijenos slike u tekst

Na korak smo od trenutka kad bi trebala nastati umjetna inteligencija ili možda ne, problem je još uvijek tu. Kako svladati otiske niske kvalitete, mnogobrojne rukom pisane tekstove, svladavanje jezika koji nisu na latiniči, otiske s visokim postotkom šuma itd. sve njih čovjek još uvijek svladava mnogo lakše nego OCR, no nove generacije OCR brzo uče. Baze fontova više nisu pohranjene na lokalnim računalima već rasprostranjene po svijetu, adaptiraju se u trenu i pronalaze rješenja, prepoznaju cijele riječi ili rečenice i prevode ih na odabrani jezik. Optičko prepoznavanje je toliko rasprostranjeno i trenutno se nalazi u svim našim domovima u nekom obliku da toga nismo ni svjesni (4). OCR je do nedavno bila jedna od najistraživanijih grana informatičkog razvoja no polako gubi tu titulu jer većina komunikacije i publikacije se obavljaju već na računalima, sve je digitalizirano i nema potrebe za prepoznavanjem, tako da se bit istraživanja premjestila na srodne grane OCR-u.[1][16]
2.3. Osnovni pojmovi
2.3.1. Pismo i njegove karakteristike

Pismo je vrsta sustava simbola koje služi za zapisivanje jezika na površinu, služi nam kao oblik komunikacije, koja se jedino može ostvariti ako svaki simbol ima dogovoreno značenje u okviru pojedinog jezika. U različitim kulturama pisma predstavljaju lingvističke jedinice (glasove, slogove riječi) na različitim strukturnim razinama, dok u slovnim pismima (latinici, čirilici) simboli predstavljaju slova koja se slaganjem formiraju u riječi.

Slika 5: Raspodijela pisma u svijetu

Svaka kultura kroz vrijeme je razvila svoj sustav simbola s kojima je prenosila značenja svog jezika, neke su posudile tude, a najbolji primjer je japansko pismo koje koristi kineske simbole - ideograme (Kandi), slogovno pismo (Kana), te uz to i svoja dva slovna pisma Katakana i Hiragana. Na svijetu se danas koristi oko 30-tak različitih pisama (5), ako se ne uzimaju u obzir male razlike u pravopisu. Svakodnevno nastaju i novi simboli koji se uvrštavaju u pismo zahvaljujući potrebama računalnog razvoja.

[21]
2.3.2. Uređaji za unos

Od prvih rukom pisanih znakova, tiskarske preše, pa do danas, uvjet da se otisnuti ili pisani simboli vide je kontrast simbola u odnosu na površinu. Na tome također leži osnova samih optičkih uređaja (skenera, kamere, fotoaparatata) za unos dokumenata u računalo.

- **Skener**

Skener je ulazni uređaj za digitalizaciju tj. za neposredni unos slike, crteža ili teksta najčešće sa papira u računalo. Princip rada zasniva se na pretvorbi svjetlosti, koja se odbije sa skeniranog predmeta, u električni impuls. Odbijene zrake svjetlosti usmjeravaju se u sustav leća i ogledala prema senzorima za pretvorbu u električnu struju. Slika je u tom postupku podjeljena na točke, u računalu zapisane kao brojevi, a veći broj točaka predstavlja i veću kvalitetu. Broj unesenih točaka po prostoru predstavlja razlučivost tj. rezoluciju načrtno izraženu u točkama po inču (DPI – dots per inch) ili u broju točaka po milimetru (mm-1). Danas se najčešće koristi razlučivost od 300DPI (11.8mm-1) s nivoem osvijetljenosti od 8 bita (256 razina). [18]

-**Digitalna kamera i fotoaparat**

Digitalna kamera i fotoaparat su uređaji koji umjesto na klasičan film sliku "hvataju" na specijalizirani mikročip CMOS ili 3CCD. Mikročip digitalizira na njega projiciranu sliku te definira horizontalnu i vertikalnu poziciju svake točke, nijansu boje i intenzitet osvjetljenja. Sliku u digitalnom obliku potom pohranjuje na adekvantan medij (CD, DVD, SD...). Kod digitalnih kamere je jedino specifično to da se slike hvataju u određenom broju u sekundi (najčešće 25 slika po sekundi – eng. Frame per second FPS) u obliku video zapisa, da bi snimka dobila što prirodniji pokret.

-**Elektronska osjetljiva površina**

Elektronske osjetljive (6) površine su uređaji za digitalizaciju rukom pisanog teksta u računalo pomoću za to namjenjenim olovkama. Te površine su najčešće ekrani sa tekućim kristalima ili digitalizacijske ploče koje u kontaktu sa olovkom, pisanjem, šalju elektronske signale u računalo. Postoje dva pristupa unos, vansistemsko (off-line)
i unutarsistemsko (on-line) pisanje. Kod unutarsistemskeko postupka očitavaju se dvodimenzionalne koordinate dodira vrha olovke po površini u određenom vremenskom redoslijedu čime se dobiva niz poteza koji se naprave tijekom pisanja, dok kod vansistemskeko dobivamo samo sliku napisanog pisma bez da znamo kako se olovka kretala.

Sama količina podataka koja se obrađuje je mnogo veća kod vansistemskeko unos, a prepoznavanje je s manjim postotkom točnosti. Radi toga se unutar sistemskeko prepoznavanje na osjetljivim površinama najčešće koristi u bankama, poštama i javnom upravama pri identifikaciji potpisa.

Slika 6: Površina za unos elektronskih potpisa
2.3.3. Rezolucija slike

Digitalne slike se sastoje od malih kvadratića zvanih pikseli. Ta kvaliteta slike izravno ovisi o slikovnoj rezoluciji. Postoje dva tipa rezolucija: prostorna i izlazna. Prostorna rezolucija se definira brojem piksela po visini i širini, npr. Slika od 600x400 piksela sadrži 240 000 piksela. Što je veća prostorna rezolucija to je veći broj stvorenih piksela, a time je najčešće moguć veći broj detalja i oštrija slika.

Slika 7: Degradacija slike sa 512x512 na 16x16

Slika (7) nam pokazuje promijene u dimenziji slike ali se teško zamjećuje "efekt" na znaku jer su promijene u skladu sa veličinom. Tek kad sliku razvučemo na jednake dimenzije i napravimo usporedbu (8), tek tad možemo uočiti gubitke u kvaliteti slike. Rezultat su izgubljeni krajevi, iskrivljeni serifi, istanjenost ili zadebljanost određenih površina i gubitak broja piksela na površini oblika.

Slika 8: Prikaz razvučenih niskorezolucijskih slika na orginalnu veličinu
Izlazna rezolucija se definira po broju piksela po dužini inča (dpi). Cilj je da se lakše poklope rezolucije slike sa rezolucijom izlaznog uređaja poput kompjuterskog zaslona, mobilnog zaslona, printera itd. Što je veći dpi slike, naravno do granica izlaznog uređaja, to je bolja kvaliteta ispisa slike. Standardni injet i laserski printeri imaju rezoluciju izlaza od 1200 dpi dok monitori najčešće se kreću oko 96 dpi.

2.3.4. Postupak prepoznavanja

Prepoznavanje pisma je pretvorba pisma iz prostornog oblika sastavljenog od grafičkih znakova u računalne simbole koji su u engleskom pravopisu predstavljeni u obliku ASCII koda, dok se kod većine ostalih jezika najčešće koristi Unicode. Cilj je pri prepoznavanju pisma prepoznati slova i riječi (9) pri čemu je neophodan korak predobrada gdje se određuje mjesto na kojima se nalazi tekst u dokumentu sa složenim rasporedom teksta i grafike. Sam postupak prepoznavanja naziva se analiza slike dokumenta.

Slika 9: Prikaz konvencionalnog pristupa OCR-u

Osnovni zadatak je određivanje kojoj skupini simbola pripada digitalizirani znak. Sam proces prepoznavanja otisnutih simbola naziva se optičko prepoznavanje znakova, dok se kod prepoznavanja rukom pisanih simbola naziva inteligentno prepoznavanje znakova. Osnovne skupine simbola su mala i velika slova, brojevi te posebni znakovi (interpunkcije, valute, itd.). Da bi se odredila skupina u koju spadaju koriste se neki od algoritama za prepoznavanje uzoraka. Jedna od lakših metoda je klasifikacija rezolucije pomoću umjetne neuronske mreže.
Algoritam zasnovan na metodi najveće sličnosti su su precizniji, no ako imamo veći broj skupina simbola, ti algoritmi su znatno sporiji. Složeniji algoritmi prepoznaju i cijele riječi, a oni najnapredniji mogu i same riječi analizirati unutar rečenice i odabiru onu koja najbolje gramatički odgovara (10). To su jezični modeli koji se koriste za rekonstrukciju niza riječi nakon što su prošli kroz kanale sa "šumom". Najznačajniji model pri prepoznavanju je u biti riječnik. Riječnik sadrži lingvistička ograničenja, tako da je tijekom prepoznavanja teksta skup mogućih riječi za svaku sliku riječi ograničen sintaksom, semantikom i pragmatikom rečenice. Uključivanje statističkih informacija na razini niza riječi povećava performanse sustava za prepoznavanje. Do povećanja performansi dolazi zbog mogućnosti izbora riječi.

Slika 10: Shematski prikaz diagrama za pretraživanje dokumenata i riječnika
2.3.5. OCR softver

Način na koji OCR sustav radi mogao bi se opisati kroz par faza. Prepoznavanje počinje prvo učitavanjem slike pomoću određenog OCR softvera u kojem se vrši obrada slike pri čemu se dobivaju određena slikovna svojstva. Svojstva se uspoređuju s postojećom bazom uzoraka sa sličnim ili istim svojstvima tj. uspoređivanje matrice (eng. Matrix matching) i pronalaženje i izvlačenje ključnih značajki (eng. Feature extraction).

Softveri se sve više razvijaju te radi lakše klasifikacije dijelimo ih u par grupa. Prva podijela je na "Desktop" i "Server" softvere. Desktop softveri se instaliraju na osobna računala ili servere gdje im se i instalira i baza uzoraka, samo prepoznavanje se odvija izravno na računalu tako da brzina i kvaliteta ovisi i o karakteristikama računala tj. hardvera. Kod Server softvera baza uzoraka se nalazi na serveru te se njoj može pristupati s više računala istovremeno, a time se najčešće i nadopunjavanje odvija brže. Sami serveri su najčešće jača računala koja prepoznavanje i uspoređivanje obave brže od standardnih računala kučne upotrebe.

Druga grupa bi bila Web i Online OCR softveri koji se zahvaljujući napretku IT tehnologije i novim platformama za rad (pametni telefoni, tableti, igraće konzole, televizori...) u današnje vrijeme sve više primjenjuju. Baze se nalaze na glavnom serveru firme koja izdaje softver te se njima pristupa direktno preko internet stranice ili preko dodatka softveru koji se instalira na računalo ili novu platformu te se povezuje sa bazom uzoraka na serveru. Te baze su daleko opširnije i svakodnevno se nadograđuju novim uzorcima, a serveri su jaka računala koja mnogo brže odrade prepoznavanje i uspoređivanje tako da nam brzina prepoznavanja najčešće ovisi o samoj internet vezi i njenoj brzini. [8][17]
Optičko prepoznavanje je izravno uspoređivanje slike znaka iz teksta koji skeniramo ili fotografiramo sa slikom znaka u postojećoj bazi uzoraka. Uzorci u bazi podataka kojima pristupa sustav prilikom uspoređivanja znakova moraju biti što bolje definirani i kategorizirani da bi se ubrzala i olakšala njihova pretraga. Ako usporedba ne daje nikakve rezultate, baza se mora nadopuniti. U tom slučaju nastupa naša intervencija te mi unosimo traženi znak koji se sprema u bazu i služi kao novi slikovni uzorak za uspoređivanje. Svaki uzorak u procesu pretvorbe sadrži zamnimljive elemente koji tada dobivaju određene smislene i prepoznatljive značajke koje sužuju prostor pretrage. Na kraju uzorci slike imaju jedinstvena svojstva koja određuju taj pojedini znak te mu se dodjeljuju određene vrijednosti za kompjuter prepoznatljiv oblik tj. poziciju u ASCII ili Unicode kodu.

Svojstva slike se najčešće okarakteriziraju na dva načina, prvi je sveukupno svojstvo tj. prosječna razina sivog tona svih piksela u sivoj slici. Drugo je specifično svojstvo tj. dio slike sa jedinstvenim prepoznatljivim dijelom. Ponekad značajke slike nisu jasno povezane s nekim dijelom slike ali uvijek odražavaju dio njenih svojstava. Svako svojstvo je povezana sa različitim algoritmom, ako nije onda nam je svojstvo beskorisno. Postoje uvijetima koje mora zadovoljiti svako svojstvo slike:

1. postojano pri transformaciji – svojstvo slike mora biti prepoznatljivo iako se slici primjene različite transformacije poput translacije, rotacije, skaliranja itd.

2. postojano pri šumu – svojstvo slike mora biti prepoznatljivo iako se u slici pojavi šum poput zrnatosti, izblijeđenih piksela, moare-a itd.

3. jednostavna ekstrakcija – svojstvo slike da se brzo i učinkovito prepoznaju

4. efikasno uspoređivanje – svojstvo slike gdje algoritam prepoznavanja zahtijeva razumne računalne troškove
Odabir slikovnih značajki i odgovarajućih metoda je najvažniji korak u ostvarenju kvalitetnih performansi pri optičkom prepoznavanju. U isto vrijeme slikovna svojstva i metode odabira određuju izlazne korake obrađene slike. Neka svojstva slike i algoritmi su namijenjeni za rad na slikama u boji dok se druga svojstva i algoritmi se isključivo primjenjuju na binarnim ili slikama u sivom tonu. Gledano općenito svaka slika mora imati svoja specifična svojsta i dodijeljene algoritme da bi što točnije bila definirana i klasificirana. [8]

3.1. Predobrada slike

Slika koja se analizira često nije savršena i odmah spremna za prepoznavanje jer sadrži nepravilnosti poput šuma, zamućenosti, nagiba itd. te se mora prije obraditi. Obrada slike prije procesa prepoznavanja znakova smatra se posebnom predfazom kod koje se koriste posebni algoritmi. Slika se mora prilagoditi da bi se lakše mogli primjeniti standardni algoritmi za pronalaženje i izvlačenje ključnih značajki. Neki se algoritmi baziraju više na vanjskim obrisima (konturama) slike, dok drugi izračunavaju vrijednosti svakog piksela u slici.

Slika 11: Pripreme poravnavanja slike u predobradi

Faze predobrade uključuju algoritme za detektiranje rubova (11), filtriranje, binarizaciju, analizu prekoračenja rubova, popunjavanje praznina, segmentaciju itd. Sve to pogođuju da slika bude pripremljena za nove algoritme prepoznavanja i samom koraku klasifikacije i uspoređivanja sa bazom uzoraka.
3.2. Usporedba oblika

Osnova na koji se odabire uzorak iz baze je sličnost mjera. Kad se učitanom znaku, još uvijek obliku, dodijele karakteristike, uspoređuju se uzorcima i onaj s najvećim stupnjem sličnosti se odabire. Bilo koja dva oblika se mogu međusobno usporediti pomoću izračuna koristeći "metrične sličnosti". One se definiraju kao:

\[ \text{Jedinica na prostoru } A \text{ je funkcija } d \text{ na } A \times A \rightarrow R \text{ gdje je } R \text{ prostor stvarnih brojeva koji zadovoljavaju slijedeće uvijete za sve } (x, y) \in A \times A \text{ i } z \in A:} \]

\[
\begin{align*}
(a) & \quad d(x, y) \geq 0 \\
(b) & \quad d(x, y) = 0 \text{, if } x = y \\
(c) & \quad d(x, y) = d(y, x) \\
(d) & \quad d(x, y) \leq d(x, z) + d(z, y)
\end{align*}
\]

Euklidska udaljenost je jedan od najčešćih metričkih udaljenosti funkcija i definirana je kao:

\[
d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
\]

u gornjoj definiciji se pretpostavlja da su vrijednosti značajki slike numeričke. [8]
3.3. Srodne transformacije

Slika u predobradi dobije značajke koje nam olakšavaju danju analizu, slika je
dobila na kvaliteti no simboli na slici su još uvijek oblici koji nemaju definiciju. Ti
oblici se lakše definiraju ako im se otkrije transformacija tj. ako ustvrdimo da je na
njima primijenjena nekakva promijena poput:

- skaliranja (*promijene dimenzije po visini i širini*)
- translatacije (*pomak oblika po osi bez promijene dimenzije*)
- rotacije (*promijena položaja s jednom fiksnom točkom*)
- nagib (*promijena oblika s jednom fiksnom ravninom*)

Srodne transformacije su bitan dio za linearnu dvodimenzionalnu geometrijsku
transformaciju (12). Koristi se najviše da bi se lakše mapiranje vrijednosti prilikom
ulaznih i izlaznih slika. Na primjer, piksel kod ulazne slike ima poziciju (x,y) dok kod
izlazne dobiva novu vrijednost (x’,y’) no u odnosu na sveukupnu sliku zadržava ravnu
liniju s ostalim pikselima u obliku.

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} =
\begin{bmatrix}
    d_{11} & d_{12} & d_{13} \\
    d_{21} & d_{22} & d_{23} \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

Slika 12: Dvodimenzionalna geometrijska transformacija
3.4. Proces OCR-a

Glavni princip automatskog prepoznavanja znakova je prvo naučiti uređaj koje skupine uzoraka se mogu pojaviti i kako izgledaju. U OCR-u uzorci su slova, brojevi te posebni znakovi. U slučaju pojavljivanja nepoznatog ili ne prepoznatog uzorka, može se uređaj naučiti u koju skupinu spada ili mu dodijeliti kompletno novu skupinu. Kad uzorak nije prepoznat uređaj se uči tako da nam na ekranu ispiše neprepoznati uzorak i moguće uzorke od kojih mi odabiremo onaj koji se najbolje podudara.

3.5. Komponente OCR sistema

Standardni OCR sistem sastoji se od nekoliko komponenti (13) kao što se vidi na slici. Prvi korak je digitalizacija analognog dokumenta koristeći optičke uređaje za unos. Kad se pronađu područja sa tekstom svaki simbol se obrađuje u segmentacijskom procesu. Izvučeni simbol se zatim može predprocesuirati tj. mogu se ukloniti šumovi u slici radi lakšeg definiranja ključnih značajki simbola u slijedećem koraku.

Identitet svakog simbola se pronašao pomoću uspoređivanja ključnih značajki sa unesenim značajkama iz prethodnih unosa. Na kraju se informacije konteksta koriste za rekonstrukciju riječi i brojeva orginalnog teksta.
3.5.1. Skeniranje

Kroz proces skeniranja analogna slika se pretvara u digitalnu. Skener se sastoji od prenosnog mehanizma i senzora koji pretvaraju svjetlosne intenzitete u sive levele. Otisnuti dokument se najčešće sastoji od crnog otiska na bijeloj pozadini, a u slučaju da se slika sastoji od više nijansi sivih tonova, konvertira se u jednostavniju bilevelnu crno-bijelu sliku. Ovaj proces zna imati i pojave zvane "trasholding" zacrnjenje (14), što je u biti sivi ton koji se pretvorbom umijesto u bijelu prenese u crni dio slike. Količina zacrnjenja najčešće ovisi o kvaliteti same slike koja se skenira i načinu na koji se vrši skeniranje. Kod slika sa jednoličnom pozadinom vrlo ga je jednostavno ukloniti no znaju se pojaviti slike kod kojih se moraju upotrijebiti napredniji algoritmi za uklanjanje tih zacrnjenja da bi se dobila kvalitetna slika.

Najbolji algoritmi su oni koji mogu automatski prilagoditi i uklaniti zacrnjenja najčešće uz pomoć svojstava kontrasta i svijetline. Te metode zahtijevaju pohranu višelevelnih slika. To znači više memorije i sporiji rad radi veličine datoteke, no krajnji ishod je bolja slika. [13] [2]
3.5.2. Lociranje i segmentacija

Segmentacija je proces koji određuje od čega je slika sastavljena. Zato je bitno da se locira, odredi prostor na dokumentu gdje su podatci otisnuti i razdvojit ih od grafika. Najbolji primjer je kod sortiranja pošte gdje se pri skeniranju omotnice treba izdvoljiti odredišna adresa i pri tom prepoznavanje paziti da se ne umiješa markica ili logo firme.

Kad se primjeni na tekst, segmentacija je u biti izoliranje znakova ili riječi. Većina OCR algoritama segmentiraju riječi u izolirane znakove koji se zasebno prepoznaju. Najčešće se ta segmentacija prikaže kao izoliranje svake zasebne komponente tj. svako povezano crno područje. Ova metoda je uglavnom jednostavna ali se problem pojavi kad se znakovi dodiruju ili se znak sasoji od više dijelova. Ti problemi se mogu razdvojiti u četiri grupe:

- **Izvlačenje znakova koji se dodiruju ili se sastoje od više dijelova**

- **Razdvajanje šuma od teksta**
  Kod ovakvog slučaja najčešće su greške da se točke, navodnici i drugi manji simboli zamijene za šum i obrnuto.

- **Zamijena grafike ili geometrijskih simbola za tekst**
  Ovakva greška vodi do zamijene i prepoznavanja onog što nije tekst u tekstu.

- **Zamijena teksta za grafiku i geometrijske simbole**
  Tekst u ovom slučaju neće biti detektiran što se zna dogoditi kad su znakovi otisnuti u sklopu grafike. [2][11]
3.5.3. Predprocesuiranje

Dobivene slike nakon skeniranja mogu sadržavati određenu količinu šuma, također ovisno o rezoluciji skenera i uspjehu primjene metoda za uklanjanje zacrnjenja simboli mogu biti zamučeni ili razbijeni (sastavljeni od više dijelova). Neki od ovih oštećenja koja mogu kasnije uzrokovati slabije prepoznavanje (15), mogu se eliminirati koristeći predprocesuiranje da bi se poboljšali digitalizirani znakovi.

Pod poboljšanje smatramo ispunjavanje i stanjivanje znakova. Ispunjavanje eliminira male pukotine, razdvojenost i rupice u znakovima dok stanjivanje smanjuje debljinu linija. Tehnika poboljšanja se radi tako da se pomiče prozor preko binarne slike znaka, te se primjeni određena metoda tj. algoritam na sadržaj prozora.

Kao dodatak poboljšanju u predprocesuiranje se dodaje i normalizacija, to je algoritam koji se primjenjuje na cijeli znak mjenjajući mu nagib i rotaciju bez da se promijeni struktura znaka. Da bi se rotacija uspješno napravila treba prvo odrediti točan kut za koji je rotacija napravljena. Za određivanje otklona rotirane stranice i linije teksta koristi se najčešće varianta Houghove transformacije. Ipak da bi se pronašao kut pojedinog znaka moramo prvi otkriti njegovu vrijednost tj. prepoznati ga. [2]
3.5.4. Izvlačenje značajki

Cilj izvlačenja značajki je da se pronađu bitne karakteristike svakog simbola, a to je u biti najveći problem kod prepoznavanja uzoraka. Najbrži način je da se znak opiše kao rasterska slika, drugi pristup je da se izvuku određene značajke koje okarakteriziraju znak da se izbace nebitni atributi. Metoda za izvlačenje takvih značajki se najčešće dijeli u tri glavne grupe u koima se značajke mogu naći:

- raspodjela točaka
- transformacija i serija proširenja
- strukturna analiza

Različite skupine značajki mogu se procijeniti prema njihovoj osjetljivosti na šum i deformacije te prema jednostavnosti implementacije i uporabe. Rezultati takve usporedbi prikazane su u tablici /1/. Kriteriji koji se koriste u ovoj procjeni su sljedeći:

**Robusnost**

- 1) šum (Osjetljivost na nepovezane segmente, izbočine, praznine itd.)

- 2) distorzije (Osjetljivost na varijacije poput zaobljenih kuteva, nepravilnih izbočina, skupljanje itd.)

- 3) varijacije stila (Osjetljivost na varijacije u stilu kao što je uporaba različitih oblika za prikaz istih slova korištenjem serifa, nakošenosti i sl.)

- 4) translacija (Osjetljivost na pomak cijelog znaka ili njegovih komponenti)

- 5) rotacija (Osjetljivost na promjenu orijentacije (zakrenutosti) znaka)
• Praktična upotreba
  – 1) brzina prepoznavanja
  – 2) složenost primjene
  – 3) neovisnost

<table>
<thead>
<tr>
<th>Izvlačenje značajki</th>
<th>Robusnost</th>
<th>Praktična upotreba</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5</td>
<td>1 2 3</td>
</tr>
<tr>
<td>Podudaranje predložaka i tehnika</td>
<td></td>
<td>o o o o o</td>
</tr>
<tr>
<td>Transformacija</td>
<td></td>
<td>o o o o o o o</td>
</tr>
<tr>
<td>Raspodijela točaka: Zoniranje</td>
<td></td>
<td>o o o o o o o o o</td>
</tr>
<tr>
<td>Moment</td>
<td></td>
<td>o o o o o o o o o</td>
</tr>
<tr>
<td>N-tovi</td>
<td></td>
<td>o o o o o o o o o</td>
</tr>
<tr>
<td>Karakteristične lokacije</td>
<td></td>
<td>o o o o o o o o o</td>
</tr>
<tr>
<td>Presjeci i udaljenosti</td>
<td></td>
<td>o o o o o o o o o</td>
</tr>
<tr>
<td>Strukture značajke</td>
<td></td>
<td>o o o o o o o o o</td>
</tr>
</tbody>
</table>

● Jednostavna   ● Srednja   ○ Niska

Tabela 1: Procjena tehnika za izvlačenje značajki

3.5.4.1. Podudaranje predložaka i tehnika

Ove se metode razlikuju od drugih jer u biti nema izvučenih značajki. Umjesto toga imamo matricu sa slikom unešenog znaka koja se direktno uspoređuje sa protutipom znaka iz baze uzoraka. Izračunava se razlika između unešenog uzorka i svih protutipova uzoraka te se odabire onaj s najboljim poklapanjem i dodjeljuje mu se vrijednost znaka.

Tehnika je jednostavna i lako se implementira u hardver te se koristi u većini komercijalnih softverima za OCR. Problem nastaje jer je metoda osjetljiva na šum, varijacije stila i ima najviše problema sa zarotiranim znakovima (16). [2]
3.5.4.2. Metode na osnovi značajki

Kod ovih metoda iz znakova se izvlače mjere i razni proračuni koji će se koristiti za njihov opis i usporedbu s drugim znakovima. Znak se u biti prepoznaje prema najsličnijem opisu tj. značajkama. One su date kao brojevi u vektoru, a značajke vektora se koriste za predstavljanje simbola.

- Raspodjela točaka

Ova kategorija obuhvaća tehnike koje izvlače značajke na temelju statističke distribucije točaka. Te značajke su obično otporne na iskrivljenost i varijacije stila. Neke od metoda su navedene ispod.

- Zoniranje

Kod zoniranja znak se obuhvaća pravokutnikom koji je podijeljen na polja koja se preklapaju (17) ili ne preklapaju sa samim znakom, zatim se izračunava gustoća crnih točaka unutar polja te se rezultati koriste kao značajke.

- Moment

Moment predstavlja odabranu točku na crnim poljima, npr. Za značajku se može uzeti centar gravitacije ili određena koordinata sustava

- Presjeci i udaljenosti

Značajke presjeka se odabiru tako što se gleda broj koliko je puta presječen određeni vektor u nekom smjeru. Ova se tehnika najčešće koristi u komercijalnim softverima jer nije kompliciran algoritam i znatno je brži. Kad se koristi značajka udaljenosti mjere se razmaci između sjecišta na vektoru.

- N-tovi

Za značajku se uzimaju spojevi crnih i bijelih točaka po određenom redoslijedu.

- Karakteristične lokacije

Za svaku točku u pozadini znaka generira se vertikalni i horizontalni vektor. Za značajke se uzima broj koliko puta vektor presjeca liniju koja opisuje znak.

Slika 17: Zoniranje
- Transformacija i ekspanzija

Ove tehnike pomažu smanjiti dimenzionalne značajke vektora i nove značajke se izvlače neovisno od sveukupne deformacije kao što je translacija i rotacija. Transformacije koje se koriste su na osnovi osne transformacije. Najznačajniji su Fourierov, Walshov, Haarov, Hadamardn, Karhunen-Loeveev i Houghov princip.

Slika 18: Eliptični Fourierov deskriptor

Većina ovih transformacija zasniva se na krivulji koja opisuje znak. To znači da je tehnika izručito osjetljiva na šum koji se javlja na rubovima znaka ili na dijelovima znaka koji imaju razmaka. Sam šum na površini znaka je zanemariv.

- Strukturna analiza

Tijekom analize strukture izvlače se značajke koje opisuju geometrijsku i topološku strukturu simbola. S ovim značajkama pokušavaju se opisati dodaci na znaku poput linije preko slova, završne točke, razmaci između linija, lukovi i udoli (18). U usporedbi s drugim značajkama ova tehnika je iznimno tolerantna na šum i varijacije stila, ali ne toliko na rotaciju i pomak no još uvijek se radi na usavršavanju ove tehnike I algoritama.[2]
3.5.5. Klasifikacija

Klasifikacija je proces identifikacije svakog pojedinog znaka i njegovo smještanje u točnu klasu. Trenutno se razmatraju dva različita pristupa za klasifikaciju pri prepoznavanju znakova. Prvi je teorijsko odlučivanje pri prepoznavanju. Ova metoda se koristi kad se za opis znaka mogu koristiti brojevi koji predstavljaju vektore.

Također imamo karakteristike uzorka izvučene iz fizičke strukture znaka. U tom slučaju veza između karakteristika je vrlo bitna pri odabiru. Na primjer: znamo da se znakovi sastoje od vertikalnih i horizontalnih poteza, tako "L" i "T" imaju po jednu vertikalnu i jednu horizontalnu liniju te se u tom slučaju uzima strukturni princip za prepoznavanje. [2]

3.5.5.1. Metoda teorijskog odlučivanja

Osnovni pristup za teorijsko odlučivanje pri prepoznavanju je klasifikacija pomoću minimalne udaljenosti, strukturna klasifikacija i neuronske mreže. Sve tehnike će se ukratko opisati ispod.

Podudaranje

Podudaranje obuhvaća skupine tehnika na temelju sličnosti mjera gdje se izračunava udaljenost između značajki vektora koje opisuju izvučen znak i opisa svake klase. Mogu se koristiti različite mjere ali im je svima zajednička Euklidenova udaljenost. Taj minimalni klasifikator udaljenosti odlično radi kad su klase dovoljno razmaknute.

Kad se unese cijeli znak u kalsifikator i ne pronađe se nijedna značajka u bazi znakova za usporedbu koristi se koleracijski pristup. Tad se uspoređuju detalji između slike znaka i prototipa slike znaka u svakoj klasi. [2]
Optimalni statistički klasifikator

U statističkom klasifikatoru primjenjuje se pristup vjerovatnosti. Ideja proizlazi iz koristenja klasifikacijske sheme koja je optimalna, a njegova uporaba daje najmanju vjerojatnost da se pojavi pogreška u klasifikaciji.

Klasifikator koji umanjuje ukupni prosječni gubitak zove se Bayesov klasifikator. Kad mu se dodjeli nepoznati simbol opisan samo sa značajkama vektora, vjerojatnost da simbol pripada klasi \( c \) se izračunava za sve klase \( c=1...N \). Simbolu se tada dodjeljuje klasa s najvećom vjerojatnošću.

Da bi shema bila optimalna, funkcije gustoće vjerovatnosti simbola za svaku klasu bi trebale biti poznate i najčešće se polazi od pretpostavke da su sve klase jednako moguće, a funkcija gustoće da je ravnomjerno raspoređena. Parametri koji predstavljaju klasifikatore se upotprunjaju u fazi "treniranja" fazi unosa uzoraka i njihovih parametara u bazu podataka. Tijekom procesa unosa uspoređuju se svi uzorci i parametri te se izračunava zajednička značajka koja se dodjeljuje klasi. [2]

Neuronske mreže

Noviji oblik klasifikacije je prepoznavanja znakova i drugih uzoraka pomoću neuronskih mreža (19). Gledajući pozadinu mreže, ova se sastoji od više slojeva međusobno povezanih elemenata. Značajke vektora se unose u ulaznom sloju, a svaki element sloja izračunava ukupni zbroj unosa i pretvara ih u izlaznu značajku pomoću nelinearne funkcije.

Slika 19: Umijetna neuronska mreža
3.5.5.2. Strukturne metode

U području strukturnog prepoznavanja prevladavaju konstruktivne metode. Postoje i druga ali trenutno nam nisu bitne.

- Sintaktičke - konstuktivne metode

Mjerenjem sličnosti na osnovi veze između strukturnih komponenti može se formulirati koristeći gramatički koncept. Ta ideja proizlazi iz toga da svaka klasa ima svoju gramatiku koja definira sastav znaka. [2]

Naknadno procesuiranje
- Grupiranje

Rezultat prepoznavanja znakova sa dokumenta je zasebnost svakog znaka. Ti simboli najčešće ne sadrže dovoljno informacija stoga svakom znaku pridružujmo susjedni te pokušavamo sačiniti riječi ili niz brojeva. To pridruživanje nazivamo grupiranje. Grupiranje simbola u niz bazira se na poziciji simbola na unešenom dokumentu tj. znakovi koji se pronadu blizo jedan drugome sa malim razmakom se grupira u riječ. Poprilično je jednostavno grupirati znakove iz skupine fontova koja imaju specificirana pravila (istu visinu, podnožje znaka ...) no problem nastaje kod rukom pisanih tekstova pogotovo nakošenih.

- Detektiranje greške i ispravljanje

Sve do grupiranja se svaki znak tretirao odvojeno, a značenje svakog znaka nisu se koristile. Međutim, javljaju se problemi u naprednim optičkim prepoznavanju teksta, sustav koji se sastoje samo od jednog prepoznavanje znakova neće biti dovoljna. Čak i najbolji sustavi prepoznavanja neće dati 100 postotnu ispravnu identifikaciju svih znakova, ali neke od tih pogrešaka mogu se otkriti ili čak korigirati korištenjem u kontekstu.[9]
Postoje dva glavna pristupa od kojih prvi koristi metodu da se skupa ne pojavljuju dvije sekvence određenih znakova. To se može postići korištenjem pravila koja definiraju sintaksu riječi, npr. poslije točke najčešće dolazi veliko slovo. Također za različite jezike može se izračunati vjerovatnost da se dva ili više znakova pojave skupa u određenoj riječi te se iskoristiti za pronalaženje grešaka. Npr. u hrvatskom jeziku vjerovatnost da se "k" pojavi ispred "h" u riječi "nebo" daje kombinaciju koja se detektira kao greška.

Drugi pristup je korištenje rječnika, to se pokazalo kao najučinkovitija metoda za pronalaženje grešaka i njihovo ispravljanje. Kad se uzme riječ u kojoj je moguće da se nalazi greška, ta riječ se pretraži u rječniku. Ako riječ nije prisutna, označi se kao greška i prepravlja se u najsličniju riječ. Vjerojatnosti dobivene iz klasifikacije, mogu pomoći pri identifikaciji znaka koji je pogrešno klasificiran (20). Ako se riječ nalazi u rječniku to ne dokazuje da riječ ne sadrži grešku, neko samo da je neki od znakova pretvoren u slovo koje tvori novu riječ. Takve se greške se ne mogu zamijetiti ovom metodom. Mana ove metode je u tome da pretraga i usporedba oduzimaju podosta vremena.[2][9]

Slika 20: Kombinacije s malom vjerojatnošću
4. SLIKOVNE INVARIJANTE

Glavni problem u OCR-u je automatsko prepoznavanje znakova u slici bez obzira na njihovu poziciju, veličinu, orijentaciju itd. Da bi prepoznali varijacije istog znak, moraju se primijeniti značajke slike koje su nepromijenjive. Slikovne invarijacije su značajke koje imaju približno iste vrijednosti kao uzorci u bazi koji su pod utjecajem promjena kao na primjer, umanjenje, rotiranje, nagnutost, zamućenost ili šum.

Najčešće korištene značajke slike su invarijacije na osnovi izgleda oblika. Slikovno prepoznavanje zasnovano na ovim invarijacijama uključuje tri glavna problema: što oblik predstavlja, sličnosti mjera oblika i indeksiranje oblika. Među ovim problemima je najveći što oblik predstavlja. U literaturama se može pronaći različite metode i deskriptori oblika. Te metode se mogu razdvojiti u dvije kategorije: invarijacije na osnovi granica i invarijacije na osnovi prostora.


Osnova svih ovih metoda je definicija više matematičkih funkcija za predstavljanje slike i redukciju podataka. Najčešće uz ove treba još dodatne metode da se upotpune invarijacije kod značajki slike ali će mo se u ovom radu bazirati na njihovoj prednosti i manama.[4]

4.1. Invarijacije na osnovi granica

Kod zapažanja oblika u slici najčešće prvo što uočimo, a da toga nismo ni svjesni, su granice tj rubovi nekog oblika. Tako da se na tome i baziralo istraživanje u matematici da se definiraju te granica, što se na kraju pokazalu iznimno lako za izračunati. [8]
4.1.1. Lančani kod

Takozvani lančani kod je predstavio Freeman kao potrebu da predstavi linije i rubove oblika sa povezanim sekvencama sastavljenim od ravnih linija sa određenom dužinom i smjerom. Lančani kod ima dvije komponente: koordinatu početne točke i lanac kodova koji predstavljaju relativnu poziviju početnog piksela i onih koji slijede. Kod se generira koristeći promijenjivi smijer rubnih piksela i to na osnovi 4 i 8 smjerova povezivanja (21).

![Slika 21: Smjerovi rubnih piksela](image1)

Kod zatvorenih granica lančani kod ovisi o početnoj točci (22). Da bi postala nepromijenjena lančani kod se treba normalizirati pomoću slijedeće metode: lančani kod se treba tretirati kao sekvenc brojeva smjera kružno postavljenih. Tako se početna točka može promijeniti, a da se ne mijenja oblik.

![Slika 22: Lančani kod](image2)

Lančani kod: 21210066754444
Normalizacijski kod: 00667544442121

_Prva razlika:_ 70601670006717

Slika 22: Lančani kod
Lančani kod se može normalizirati radi rotacije koristeći takozvanu _prvu razliku_ koja se generira brojeći promijene smjera u suprotnom smjeru. Ona izračunava promijene smjera između završne točke i početne točke u lancu. [8]

4.1.2. Fourierovi deskriptori

![Diagram](image.png)

Slika 23: Osnovna slika simbola "A" i njen zatvoren oblik u Cartesianskom sustavu

Fourierov deskriptor se koristi za opisivanje zatvorenog planarnog oblika (23). Kad se uzme zatvoreni oblik u 2D Cartesianskom sustavu, granice se prate i sastavljaju u smjeru suprotnom od smjera kretanja kazaljki na satu zadržavajući raspodijelu _K_ točki. Svaka točka koordinate se može opisati u obliku (_x_0, _y_0), (_x_1, _y_1), (_x_2, _y_2),..., (_x_{K-1}, _x_K), a kordinate se prikazuju kao _x(k) = xk_ i _y(k) = yk_ ali samo pod uvijetom da je granica izražena kao sekvence kompleksnih brojeva.

\[ s(k) = x(k) + j y(k), k = 0, 1, 2, ..., K-1 \]

To znači da je os X stvarna dok je os Y imaginarna os sekvence kompleksnih brojeva. Koeficijent Diskretna Fourierova transformacija (DFT) kompleksne sekvence _z(u)_ je:

\[ z(u) = \frac{1}{K} \sum_{k=0}^{K-1} s(k) e^{-j 2\pi u k / K}, u = 0, 1, 2, ..., K-1 \]
Prema Fourieronoj transformaciji ukupni izgled oblika određuje frekvencija tj. učestalost komponenti. Što je frekvencija veća to je oblik detaljniji (24). Granice oblika možemo rekonstruirati koristeći koeficient $P$ u inverznoj Fourierovoj transformaciji:

$$s'(k) = \frac{1}{K} \sum_{u=0}^{P-1} a(u)e^{\frac{j2\pi uk}{K}}, \quad k = 0, 1, 2, ..., K-1$$

Iako samo koeficient $P$ se koristi za određivanje granica $s(k)$, k ipak obuhvaća prostor 0 do K-1. Drugim riječima, isti broj točaka je u približnim granicama ali ne koristi se isti broj točaka pri rekonstrukciji oblika. Primjer rekonstrukcije u gornjem prikazu pokazuje granice kvadra sa različitim $P$ koeficijentima koji se kreću od 2 do 62. Iz tog primjera vidimo da sa korištenjem većeg koeficijenta lakše postiže orginalni izgled oblika.

Fourierovi deskriptori imaju prednost pri reduciranju 2D oblika u 1D tj. funkciju. Sami deskriptori postavljaju temelj algoritmima i njihovoj klasifikaciji različitim metodama za određivanje granica oblika. Kao što je napomenuto prije, želimo
da slikovne invarijacije postanu neosjetljive ili uz minimalne promijene pri transformacijama tj. rotaciji, translaciji i skaliranju. Oni nisu direktno povezani sa slikovnim transformacijama ali promijene u parametrima se mogu povezati sa jednostavnijim operacijama na Fourierovim deskriptorima:

<table>
<thead>
<tr>
<th>Transformacija</th>
<th>Granica</th>
<th>Fourier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identitet</td>
<td>( s(k) )</td>
<td>( z(u) )</td>
</tr>
<tr>
<td>Rotacija</td>
<td>( s_r(k) = s(k)e^{j\theta} )</td>
<td>( z_r(u) = z(u)e^{j\theta} )</td>
</tr>
<tr>
<td>Translacija</td>
<td>( s_t(k) = s(k)\Delta_{xy} )</td>
<td>( z_t(u) = z(u) + \Delta_{xy}e^{j\theta} )</td>
</tr>
<tr>
<td>Skaliranje</td>
<td>( s_s(k) = \alpha s(k) )</td>
<td>( z_s(u) = \alpha z(u) )</td>
</tr>
<tr>
<td>Početna točka</td>
<td>( s_p(k) = s(k-k_0) )</td>
<td>( z_p(u) = z(u)e^{j2\pi u k_0/K} )</td>
</tr>
</tbody>
</table>

Tablica 2: Glavna svojstva Fourierovih deskriptora za neke transformacije

Simbol \( \Delta_{xy} = \Delta x + j\Delta y \). \( \delta(u) \) je startna funkcija koja ima vrijednost veću od nule kod orginala, dok je svugdje drugo vrijednost 0. U tablici /2/ vidimo sa magnitude \( z(u) \) su nepromijenjene nasprem slike jer:

\[
|z_r(u)| = |z(u)e^{j\theta}| = |z(u)| |e^{j\theta}| = |z(u)| \cdot 1 = |z(u)|
\]

Funkcija translacija slike sastoji se od dodavanja primjerenih zamjena svim koordinatama granica. Translacija nema efektana deskriptore osim za \( u=0 \) zbog startne funkcije \( \delta(u) \). Prva komponenta Fourierovih deskriptora ovisi samo o poziciji oblika i nje korisna pri opisu oblika i može se odbaciti.

Kod promijena pri skaliranju, invarijante se postižu formirajući omjer dvaju koeficijenata tako da možemo uklaoniti parametar \( \alpha \). Magnitude \( z(u) \) su nepromjenjive na promjenu početne točke:

\[
|z_p(u)| = |z(u)e^{-j2\pi u k_0/K}| = |z(u)| |e^{-j2\pi u k_0/K}| = |z(u)| \cdot 1 = |z(u)|
\]
Gornju analizu možemo koristiti da bi zadržali invarijaciju Fourierovih deskriptora na rotaciju, translataciju i skaliranje sve u isto vrijeme. U OCR softveru prihvaćamo prvih 10 elemenata $c(u-2)$ za usporedbu značajki vektora u slikovnom modulu.[8]

**Invarijacije na osnovi prostora**

Invarijacije na osnovi granica kao što je Lančani kod i Fourierovi deskriptori istražuju samo konture, a ne zahvaća unutrašnjost oblika. Stoga se mogu javljati problemi kod oblika s otvorenim granicama ili iz više dijelova tako da imaju ograničenun primjenu. Kod invarijacija na osnovi prostora uzimaju se svi pikseli u slici koji predstavljaju oblik. Radi toga što kombiniraju informacije cijelog prostora slike mogu obuhvatiti i veću količinu informacija a ne samo rubni predio oblika. Također može opisivati i oblike iz više dijelova.

Invarijacije na bazi momenata su najčešće korištene kod invarijacija slika na osnovi prostora koje se koriste u mnogim aplikacijama. Prvi tko je predstavio invarijacije na bazi momenata bio je Hu 1961 sa svojim nelinearnim kombinacijama standardnih momenata. Huove invarijacije imaju željena svojstva za opis oblika kod slika izloženim translaciji, skaliranju i rotaciji, ali se samo izračunavanje svodi na vrlo kompleksne funkcije, te složenost povećava pri rekonstrukciji oblika. Da bi se riješio taj problem uzima se Teagueov koncept ortogonalnih momenata za rekonstrukciju oblika. Da bi se riješio taj problem uzima se Teagueov koncept ortogonalnih momenata za rekonstrukciju oblika i to Zernikeovi momenti koji dopuštaju neovisne invarijacije kod oblika višeg stupnja. Također se razvili i Pseudo-Zernikovi momenti koji za razliku od standardnih Zernikeovih momenata nisu bili dovoljno otporan na smetnje u slici kao što su šum. [8]
Regularni i centralni momenti

Regularni ili tzv. geometrijski momenti se definiraju:

\[ m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) \, dx \, dy, \quad p, q = 0, 1, 2, \ldots \]

gdje je \( m_{pq} \) jednak (p + q)-ti moment slikovne funkcije f(x,y).

Centralni moment f(x,y) se definira:

\[ \mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \overline{x})^p (y - \overline{y})^q f(x, y) \, dx \, dy, \quad p, q = 0, 1, 2, \ldots \]

gdje su \( \overline{x} = m_{10} / m_{00} \) i \( \overline{y} = m_{01} / m_{00} \) centri slike.

Kod digitalne slike integrali se zamijenjuju zbrojem i \( m_{pq} \) postaje:

\[ m_{pq} = \sum_x \sum_y x^p y^q f(x, y), \quad p, q = 0, 1, 2, \ldots \]

Tad se centralni moment f(x,y) se mijenja u:

\[ \mu_{pq} = \sum_x \sum_y (x - \overline{x})^p (y - \overline{y})^q f(x, y), \quad p, q = 0, 1, 2, \ldots \]

Centralni momenti se izračunavaju koristeći centar slike koji je ekvivalent regularnim (geometrijskim) momentima slike čiji je centar pomaknut na težište same slike, tako centralni moment postaje nepromijenjen pri slikovnoj translaciji. Kod drugih transformacija tj. kod skaliranja nastaju promijene:

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  S_x & 0 \\
  0 & S_y
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]
da bi zadržali invarijaciju pri skaliranju \( f'(x', y') \) predstavlja sliku \( f(x, y) \) poslije skaliranja

\[
S_x = S_y = \alpha \quad f'(x', y') = f'(\alpha x, \alpha y) = f(x, y), \quad x' = \alpha x, y' = \alpha y,
\]
tad imamo:

\[
m'_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x'^p y'^q f'(x', y') \, dx' \, dy' \\
= \alpha^{p+q+2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) \, dx \, dy \\
= \alpha^{p+q+2} m_{pq}
\]

\[
\mu'_{pq} = \alpha^{p+q+2} \mu_{pq}, \quad \mu'_{00} = \alpha^2 \mu_{00}
\]

na kraju normaliziramo centralni moment:

\[
\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = (p+q+2)/2, \quad p+q = 2, 3, ...
\]

\( \eta_{pq} \) je invarijantan na promjene skaliranja zbog:

\[
\eta'_{pq} = \frac{\mu'_{pq}}{\mu_{00}^{\gamma}} = \frac{\alpha^{p+q+2} \mu_{pq}}{\alpha^{2\gamma} \mu_{00}^{\gamma}} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} = \eta_{pq}
\]

kao zaključak imamo to da je normalizirani centralni moment nepromijenjiv pri translaciji i skaliranju.[8]
Huovi sedam momenata

Na osnovi normaliziranih centralnih momenata Hu uvodi sedam nelinearnih funkcija koje su nepromijenjene pri translataciji, skaliranju i rotaciji. Tih sedam momenata se definira kao slijedeće funkcije:

\[
\begin{align*}
\phi_1 &= \eta_{20} + \eta_{62} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \\
\phi_6 &= (\eta_{20} - \eta_{02})[\eta_{30} + \eta_{12}]^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})^2 \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]
\end{align*}
\]

Ovi momenti su invarijantni na slikovne transformacije ali ne i na promijene kontrasta. Huovi momenti su jedni od najkorištenijih momenata u softverima za prepoznavanje znakova i drugih uzoraka. Kvaliteta njihove izvedbe se procijenjuje i kod drugih deformacija poput zamućenja, prostorne degradacije, nasumičnog šuma, nagiba i perspektivne transformacije.[10]

Huovi momenti uzimaju svaki piksel u obzir za razliku od invarijacija na osnovi granica, stoga se i povećava vrijeme obrađivanja podataka. Upravo radi toga se konstantno radi na njegovom usavršavanju i razvijanju, najviše da se postigne što veća točnost pri manjim rezolucijama.[8]
Zernikeovi momenti

Zernike je predstavio skupinu kompleksnih polinoma \( \{V_{nm}(x, y)\} \) koji formiraju kompletan ortogonalni skup preko jediničnog prostora u polarnim koordinatama.

\[
V_{nm}(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho)e^{im\theta}
\]

\( n \) je pozitivan broj ili nula, \( m \) je također pozitivan broj ograničen sa \( n - |m| \) i sa \( |m| \leq n \); \( \rho \) je duljina vektora sa izvornog piksela \((x, y)\); \( \theta \) je kut između vektora \( \rho \) i x osi u suprotnom smjeru; dok je \( R_{nm}(\rho) \) radial polinoma definiran kao:

\[
R_{nm}(\rho) = \sum_{s=0}^{(n-|m|)/2} (-1)^s \frac{(n-s)!}{s!(\frac{n-|m|}{2} - s)!} \frac{(n-|m|)}{2} \frac{\rho^{n-2s}}{s!(\frac{n-|m|}{2} - s)!}
\]

Zernikeov moment reda \( n \) s ponavljanjem \( m \) u funkciji \( f(x,y) \) se definira kao:

\[
A_{nm} = \frac{n+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x,y)V_{nm}^*(x,y)dx\,dy
\]

gdje je:

\[
V_{nm}^*(x,y) = V_{n,-m}(x,y)
\]

Da bi izračunali Zernikeov moment za digitalne slike moramo samo promijeniti integral sa zbrojem:

\[
A_{nm} = \frac{n+1}{\pi} \sum_{x} \sum_{y} f(x,y)V_{nm}^*(x,y), \quad x^2+y^2 \leq 1
\]

Definirane značajke Zernikeova momenta same po sebi su samo nepromijenjive na rotaciju, da bi se postigla invarijantnost na skaliranje i translaciju slika treba prvo biti normalizirana koristeći regularni Zernikeov moment. Za postizanje nepromijenjivosti na translaciju moramo orginalnu sliku \( f(x,y) \) translatirati \( f(x+x', y) \)
+ y) pri čemu je:

\[ \overline{x} = \frac{m_{10}}{m_{00}} \quad \overline{y} = \frac{m_{01}}{m_{00}} \]

Drugim riječima centar orginalne slike je pomaknut na križanja koordinata prije primjene izračuna Zernikeova momenta.

Nepromjenjivost kod skaliranja se postiže uvečavanjem ili reduciranjem oblika tako da slikovni nulti regularni moment \( m'00 \) bude jednak već zadanoj vrijednosti \( \beta \). Za binarnu sliku \( m'00 \) ima vrijednost ukupnog broja piksela u slici oblika. Prema jednadžbi za skaliranu sliku \( f(\alpha x, \alpha y) \) regularni moment se izračunava pomoću \( m'_{pq} = \alpha^{p+q} m_{pq} \), \( m_{pq} \).

Ako pretpostavimo da znamo sve Zernikeove momente \( A_{nm} \) od funkcije \( f(x, y) \) do reda \( N \) prema ortogonalnim svojstvima sa Zernikeovim momentima trebali bi moći rekonstruirati sliku pomoću funkcije [8]:

\[ f'(x, y) = \sum_{n=0}^{N} \sum_{m} A_{nm} V_{nm}(x, y) \]
4. EKSPERIMENTALNI RAD

4.1. Uvod u eksperimentalni rad

Slikovne invarijacije na osnovi granica i prostora koje su ranije opisane pokazuju nam svojstva pod utjecajem različitih transformacija u koje spadaju promijene pri translaciji, skaliranju i rotaciji. Da bi smo mogli napraviti kvalitetnu procijenu izvedbe različitih slikovnih invarijacija i istražili njihove prednosti i mane moramo testirati neke od njih kroz određen set slikovnih primjera. U ovom poglavlju ćemo prezentirati i raspraviti primjenjene algoritme. Zbog prevelikog raspona izračuna kod invarijacija na osnovi prostora na slikama visokih rezolucija, držat ćemo se istraživanja slika na nižim rezolucijama i samo osjetljivosti algoritama na njih. Da ne bi širili temu izbijegavati ćemo susret sa šumom u slici tj. bazirati ćemo se na slikama bez utjecaja smetnji u njima.

U poglavlju će mo istražiti svojstva Fourierovih deskriptora i Huovih sedam momenata koji će biti implementirani u OCR softver u kojima su korištene MATLAB skripte. S ovim OCR softverom korisnici mogu stvoriti vlastitu bazu slikovnih značajki baziranim na Fourierovim i Huovim invarijacijama i spremiti ih bazu u obliku MATLAB datoteka koje se kasnije mogu pozvati za usporedbu.

4.2. Slikovni primjeri

Za slikovne primjere u eksperimentu koristit ćemo 8 slova hrvatske abecede u standardnom crno bijelom ispisu od koih će jedno biti iz više dijelova da prikažemo prednost prostorne nad graničnom invarijacijom. Također će mo s njim pokazati kako Fourierovi deskriptori dvodijelne obrise stope u granicu iz jednog dijela te time izgubi dio podataka (25) te tako mogu voditi do netočnih rezultata.
Koliko god pokušavali maksimalno iskoristiti mogućnosti slikovnih invarijanti osjetljivih na smanjenje rezolucije, uvijek je bitan korak predprocesuiranje da se uklone nedostatci na slici. Tu pomaže segmentacija, uklanjanje zacrnjenja i stanjivanje linija oblika.

Rezolucije koje će mo koristiti u eksperimentu kod slikovnih primjera su 16x16, 32x32, 64x64, 128x128, 256x256 i 512x512. Za primjer će mo uzeti veliko slovo "M" (26).
4.3. Opis primjene

Za invarijacije na osnovi granica koristimo algoritme Fourierovih deskriptora, za invarijaciju na osnovi prostora uzet ćemo regularni moment, normalizacijski moment, Huovih sedam momenata i Zernikeove momente. Korištenjem ovih algoritama opisat će mo u slijedećim poglavljima.

4.3.1. Fourierovi deskriptori

Slika 27: Diagram izračunavanja Fourierovih deskriptora

Diagram nam pokazuje proces računanja Fourierovih deskriptora (27). Na početku svi sivi leveli se pretvaraju u binarnu sliku pomoću uklanjanja zacrnjenja prateći slikovni histogram. Kod binarizacije slike bijeli pikseli se predstavljaju vrijednošću "1", te crni vrijednošću "0".

Kod praćenja oblika po granici vraćamo X i Y koordinate za sve piksele uzduž granice slova. Proces praćenja počinje na najgornjem lijevom pikselu slike i prati u smjeru suprotnom od kazaljke na satu dok ne zatvori obris. Sve koordinate se sačuvaju u 2xn matrici, gdje je n ukupni broj graničnih piksela. Prvi red matrice čuva X koordinate te drugi red Y kordinate.

Kod izračunavanje transformacije sačinjene od X i Y koordinata graničnih piksela bitno je rastaviti sliku do najmanjeg broja te ju sastaviti ovisno o našem zahtjevu. U eksperimentu ćemo broj odrediti na 64 tako da će sastavljena slika sadržavati piksele koordinata sačuvane u 2x64 matrici.
4.3.2. Huovih sedam momenata

Gornji prikaz pokazuje proces izračunavanja Huovih sedam momenata (28). Isto kao kod Fourierovih deskriptora slika se prvo binarizira, zatim se izračunava regularni moment sa \( m = \text{moment} (\text{fig}, p, q) \). \( \text{fig} \) je unos binarne slike, a \( p \) i \( q \) su preddefinirani momenti reda. S ovim parametrima možemo dalje izračunavati regularni moment. [14]

\[
m_{pq} = \sum_{x} \sum_{y} x^p y^q f(x, y)
\]

\( m_{pq} \) prolazi kroz sve piksele u slici. Za binarnu sliku \( m_{00} \) je ukupni broj bijelih piksela u slici. Centriranje binarne slike se računa:

\[
\bar{x} = m_{10} / m_{00} \quad \bar{y} = m_{01} / m_{00}
\]

Slično regularnom momentu, centralni moment se izračunava funkcijom:

\[
\mu = \text{central\_moment} (\text{fig}, p, q)
\]

a to se izračunava prema definiciji:

\[
\mu_{pq} = \sum_{x} \sum_{y} (x - \bar{x})^p (y - \bar{y})^q f(x, y)
\]
4.3.3. Zernikeovi momenti

Gledano nasprem Huovih momenata, Zernikeovi momenti su složeniji. Glavni razlog za to je normalizacijski procesi jer Zernikeovi momenti (29) su samo invarijantni na slikovnu rotaciju. Da bi se postigla invariјacija na translaciju i skaliranje mora se proći normalizacijski postupak za oba slučaja.

Normalizacija kod translacije se postiže pomicanjem slikovnog centra na težište slike znaka (30)(31).

Slika 30: Usporedba orginalne i normalizirane translirane slike

Slika 31: Usporedba originalne i normalizirane skalirane slike
Normalizacija skaliranjem se postiže postavljanjem slike sa regularnog momenta nultog reda $m_00$ na predodređene vrijednosti. Zbog toga što je $m_00$ ukupni broj bijelih piksela, za binarnu sliku koristimo interpolaciju da bi $m_00$ dobila predodređene vrijednosti.

Za razliku od regularnog momenta koji učitava zbrojeviksela unutar kvadrata, Zernikeovi polinomi zauzimaju jediničnu površinu u obliku kruga (32) $x^2+y^2 \leq 1$. Kod izračunavanja Zernikeova momenta za digitalnu sliku, kad se podesi centar dvaju slike, mapira se prostor ispod kruga. Sve što ostane vani se izbaci iz procesa izračunavanja. To se radi da bi se smanjilo vrijeme izračunavanja. Kao primjer može se dati prostorna rezolucija binarne slike 64x64 što kod Zernikeova uklanjanja pomoću kruga iznosi samo vrijednost radijusa od 32 piksela.

Slika 32: Prikaz redukcije prostora mapiranja za Zernikeove momente

Zernikeovi momenti samo po sebi nisu invariantni na translaciju i skaliranje već slika treba biti predprocesirana da bi se dobili pogodnija svojstva za invarijaciju.

4.4. OCR engine

Testiranje izvedbe Fourierovih deskriptora i Huovih sedam momenata na slikovnim primjerima sa različitim transformacijama i prostornim rezolucijama, izvršava se ubacivanjem MATLAB skripte u "OCR engine".
"OCR engine" (33) uključuje dvije osnovne funkcije: izvlačenje značajki i slikovno prepoznavanje (34).

Slike se učitavaju u OCR engine preko "Image" menija te se odabire "Open" izbor u meniju. OCR engine podržava rad sa "TIFF" i "JPEG" slikovnim formatima. Izvlačenje značajki je u biti proces treniranja (učenja) i popunjavanja baze modela. Nakon što je slika učitana dodjeljujemo joj slovo iz abecedne kode koje slika predstavlja. Pritiskom na tipku “Get Fourier Descriptors” ili “Get Hu’s Invariants” počinje izračunavanje deskriptora ili momenata. Dobiveni podaci se pohranjuju kao MATLAB datoteka ".mat" u bazu modela kad se pritisne "Data" izbornik i odabere "Save".
Pri koraku slikovnog prepoznavanja, učitavamo prvo bazu modela odabirući "Data" meni (35). Tad možemo otvoriti sliku i implementirati prepoznavanje. Pritiskom na tipku za prepoznavanje, OCR engine može izvući značajke trenutne slike i izračunati Euklidianovu udaljenost.

Slika 35: Prikaz klasifikacije slike "A" 128x128 rotirane za -45°
5. REZULTATI EKSPERIMENTA

Eksperimentalni rezultati se baziraju na dva procesa i to na izvlačenju značajki i prepoznavanju slike. Rezultati izvlačenja značajki uklučuju izračun Fourierovih deskriptora i Huovih sedam momenata iz učitane slike sa različitim prostornim rezolucijama. Kod prepoznavanja slike, stupanj prepoznavanja je postignut koristeći OCR engine za prepoznavanje slika 8 velikih slova abecede koja su bila izložena različitim rezolucijama.

Za Zernikeove momente, eksperiment se fokusirao na istraživanju mogućnosti rekonstrukcije slike. Više primjeraka rekonstruiranih slika baziranim na Zernikeovom momentu je analizirano /3/.

5.1. Rezultati Fourierivih deskriptora

Tabela 3: Prvih 10 Fourierovih deskriptora za slike 8 velikih slova abecede sa prostornom rezolucijom 512x512

<table>
<thead>
<tr>
<th>Fourier</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>0.6407</td>
<td>0.1362</td>
<td>0.1197</td>
<td>0.0137</td>
<td>0.0471</td>
<td>0.0234</td>
<td>0.0255</td>
<td>0.0224</td>
<td>0.0112</td>
<td>0.0170</td>
</tr>
<tr>
<td>A</td>
<td>0.5777</td>
<td>0.3097</td>
<td>0.1774</td>
<td>0.0810</td>
<td>0.0324</td>
<td>0.1030</td>
<td>0.0741</td>
<td>0.0093</td>
<td>0.0084</td>
<td>0.0362</td>
</tr>
<tr>
<td>R</td>
<td>0.1020</td>
<td>0.0169</td>
<td>0.0098</td>
<td>0.0032</td>
<td>0.0165</td>
<td>0.0079</td>
<td>0.0031</td>
<td>0.0094</td>
<td>0.0085</td>
<td>0.0001</td>
</tr>
<tr>
<td>I</td>
<td>0.4699</td>
<td>0.1580</td>
<td>0.1354</td>
<td>0.1018</td>
<td>0.0898</td>
<td>0.0438</td>
<td>0.0201</td>
<td>0.0340</td>
<td>0.0208</td>
<td>0.0205</td>
</tr>
<tr>
<td>O</td>
<td>0.3296</td>
<td>0.0803</td>
<td>0.0379</td>
<td>0.0839</td>
<td>0.0319</td>
<td>0.0149</td>
<td>0.0260</td>
<td>0.0067</td>
<td>0.0043</td>
<td>0.0080</td>
</tr>
<tr>
<td>K</td>
<td>0.9483</td>
<td>0.4274</td>
<td>0.1452</td>
<td>0.0498</td>
<td>0.0794</td>
<td>0.0115</td>
<td>0.0275</td>
<td>0.0332</td>
<td>0.0105</td>
<td>0.0221</td>
</tr>
<tr>
<td>S</td>
<td>0.0381</td>
<td>0.1239</td>
<td>0.0602</td>
<td>0.0468</td>
<td>0.0384</td>
<td>0.0082</td>
<td>0.0080</td>
<td>0.0108</td>
<td>0.0089</td>
<td>0.0078</td>
</tr>
<tr>
<td>C</td>
<td>0.1440</td>
<td>0.4992</td>
<td>0.0395</td>
<td>0.0757</td>
<td>0.0144</td>
<td>0.0277</td>
<td>0.0335</td>
<td>0.0508</td>
<td>0.0117</td>
<td>0.0196</td>
</tr>
</tbody>
</table>

Slika 36: Rasprostranjenost F.D. za sliku slova "O" sa rezolucijom 512x512
Iz primjera vidimo da su najznačajnije vrijednosti na krivulji dobivene iz prvih 10 Fourierovih deskriptora (36). Također je uzeto prvih deset elemenata da bi sastavili značajke vektora, dok su daljne vrijednosti jako blizu nuli pa se ne uzimaju.

5.2. Nepromjenjivost pri različitim prostornim rezolucijama

Proučavanjem invarijacija svojstava Fourierovih deskriptora na različite prostorne rezolucije sakupljeni su deskriptori svih primjera slika sa rezolucijama od 512x512 do 32x32. /4//5/ (37) (38)

<table>
<thead>
<tr>
<th>O</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x512</td>
<td>0.3296</td>
<td>0.0803</td>
<td>0.0379</td>
<td>0.0839</td>
<td>0.0319</td>
<td>0.0149</td>
<td>0.0260</td>
<td>0.0067</td>
<td>0.0043</td>
<td>0.0080</td>
</tr>
<tr>
<td>256x256</td>
<td>0.3371</td>
<td>0.0797</td>
<td>0.0347</td>
<td>0.0854</td>
<td>0.0311</td>
<td>0.0142</td>
<td>0.0254</td>
<td>0.0067</td>
<td>0.0044</td>
<td>0.0080</td>
</tr>
<tr>
<td>128x128</td>
<td>0.3521</td>
<td>0.0773</td>
<td>0.0293</td>
<td>0.0889</td>
<td>0.0293</td>
<td>0.0108</td>
<td>0.0282</td>
<td>0.0084</td>
<td>0.0057</td>
<td>0.0059</td>
</tr>
<tr>
<td>64x64</td>
<td>0.3417</td>
<td>0.0860</td>
<td>0.0324</td>
<td>0.0893</td>
<td>0.0361</td>
<td>0.0193</td>
<td>0.0307</td>
<td>0.0125</td>
<td>0.0120</td>
<td>0.0028</td>
</tr>
<tr>
<td>32x32</td>
<td>0.5781</td>
<td>0.1667</td>
<td>0.0345</td>
<td>0.1149</td>
<td>0.0200</td>
<td>0.0363</td>
<td>0.0453</td>
<td>0.0358</td>
<td>0.0086</td>
<td>0.0177</td>
</tr>
<tr>
<td>MEAN</td>
<td>0.3877</td>
<td>0.0980</td>
<td>0.0338</td>
<td>0.0925</td>
<td>0.0297</td>
<td>0.0191</td>
<td>0.0311</td>
<td>0.0140</td>
<td>0.0070</td>
<td>0.0085</td>
</tr>
<tr>
<td>STD DEV</td>
<td>0.0955</td>
<td>0.0345</td>
<td>0.0028</td>
<td>0.0114</td>
<td>0.0054</td>
<td>0.0090</td>
<td>0.0073</td>
<td>0.0111</td>
<td>0.0029</td>
<td>0.0050</td>
</tr>
</tbody>
</table>

Tabela 4: Prvih 10 FD slike slova "O" sa prostornom rezolucijom od 512x512 do 32x32

Slika 37: Graf prvih 10 FD slike slova "O" sa rezolucijom od 512x512 do 32x32

<table>
<thead>
<tr>
<th>C</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x512</td>
<td>0.1440</td>
<td>0.4992</td>
<td>0.0395</td>
<td>0.0757</td>
<td>0.0144</td>
<td>0.0277</td>
<td>0.0335</td>
<td>0.0508</td>
<td>0.0117</td>
<td>0.0196</td>
</tr>
<tr>
<td>256x256</td>
<td>0.1400</td>
<td>0.5012</td>
<td>0.0397</td>
<td>0.0737</td>
<td>0.0143</td>
<td>0.0276</td>
<td>0.0327</td>
<td>0.0503</td>
<td>0.0125</td>
<td>0.0180</td>
</tr>
<tr>
<td>128x128</td>
<td>0.1367</td>
<td>0.5023</td>
<td>0.0363</td>
<td>0.0723</td>
<td>0.0117</td>
<td>0.0260</td>
<td>0.0318</td>
<td>0.0518</td>
<td>0.0097</td>
<td>0.0188</td>
</tr>
<tr>
<td>64x64</td>
<td>0.1234</td>
<td>0.5062</td>
<td>0.0380</td>
<td>0.0767</td>
<td>0.0124</td>
<td>0.0290</td>
<td>0.0221</td>
<td>0.0572</td>
<td>0.0150</td>
<td>0.0202</td>
</tr>
<tr>
<td>32x32</td>
<td>0.1362</td>
<td>0.5020</td>
<td>0.0422</td>
<td>0.0843</td>
<td>0.0008</td>
<td>0.0712</td>
<td>0.0385</td>
<td>0.0434</td>
<td>0.0242</td>
<td>0.0070</td>
</tr>
<tr>
<td>MEAN</td>
<td>0.1361</td>
<td>0.5022</td>
<td>0.0392</td>
<td>0.0765</td>
<td>0.0107</td>
<td>0.0363</td>
<td>0.0317</td>
<td>0.0507</td>
<td>0.0146</td>
<td>0.0167</td>
</tr>
<tr>
<td>STD DEV</td>
<td>0.0069</td>
<td>0.0023</td>
<td>0.0020</td>
<td>0.0042</td>
<td>0.0051</td>
<td>0.0175</td>
<td>0.0053</td>
<td>0.0044</td>
<td>0.0051</td>
<td>0.0049</td>
</tr>
</tbody>
</table>

Tabela 5: Prvih 10 FD slike slova "C" sa prostornom rezolucijom od 512x512 do 32x32
5.3. Rezultati Huovih sedam momenata

Tabela 6: Huovih sedam momenata na slikama 8 slova abecede pri rezoluciji 512x512

<table>
<thead>
<tr>
<th>HU</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>0.3669</td>
<td>0.0076</td>
<td>0.0347</td>
<td>0.0018</td>
<td>0.0000</td>
<td>0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>A</td>
<td>0.2838</td>
<td>0.0036</td>
<td>0.0006</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>R</td>
<td>0.5330</td>
<td>0.0173</td>
<td>0.0239</td>
<td>0.0280</td>
<td>-0.0004</td>
<td>-0.0025</td>
<td>0.0006</td>
</tr>
<tr>
<td>I</td>
<td>0.3330</td>
<td>0.0001</td>
<td>0.0033</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>O</td>
<td>0.3874</td>
<td>0.0217</td>
<td>0.0006</td>
<td>0.0022</td>
<td>0.0000</td>
<td>-0.0003</td>
<td>0.0000</td>
</tr>
<tr>
<td>K</td>
<td>0.3698</td>
<td>0.0374</td>
<td>0.0106</td>
<td>0.0008</td>
<td>0.0000</td>
<td>-0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>S</td>
<td>0.4868</td>
<td>0.0031</td>
<td>0.0039</td>
<td>0.0052</td>
<td>0.0000</td>
<td>-0.0001</td>
<td>0.0000</td>
</tr>
<tr>
<td>C</td>
<td>0.3354</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Iz tabele 6/ je vidljivo da su samo prva tri momenta bitna dok ostala četiri teže nuli. Pošto prva tri daju jako dobre rezultate može se zaključiti da se Huovih sedam momenata može i dalje koristiti kao algoritam u softverima.

5.3.1. Nepromjenjivost pri različitim prostornim rezolucijama

Kao i kod Fourierovih deskriptora i kod Huovih momenata proučavaju se izračuni i svojstva pri različitim prostornim rezolucijama od 512x512 do 32x32./7/ Uzet će se slike određenih slova i prikazati će ih se u tablici kroz slike dijagrama(39).
Tabela 7: Huovih sedam momenata na slikama slova "R" sa prostornom rezolucijom od 512x512 do 32x32

<table>
<thead>
<tr>
<th>Resolucija</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
</tr>
</thead>
<tbody>
<tr>
<td>512x512</td>
<td>0.5330</td>
<td>0.0173</td>
<td>0.0239</td>
<td>0.0280</td>
<td>-0.0004</td>
<td>-0.0025</td>
<td>0.00006</td>
</tr>
<tr>
<td>256x256</td>
<td>0.5319</td>
<td>0.0171</td>
<td>0.0234</td>
<td>0.0281</td>
<td>-0.0004</td>
<td>-0.0026</td>
<td>0.00006</td>
</tr>
<tr>
<td>128x128</td>
<td>0.5324</td>
<td>0.0176</td>
<td>0.0242</td>
<td>0.0281</td>
<td>-0.0005</td>
<td>-0.0027</td>
<td>0.00006</td>
</tr>
<tr>
<td>64x64</td>
<td>0.5358</td>
<td>0.0166</td>
<td>0.0223</td>
<td>0.0294</td>
<td>-0.0006</td>
<td>-0.0030</td>
<td>0.00005</td>
</tr>
<tr>
<td>32x32</td>
<td>0.5373</td>
<td>0.0236</td>
<td>0.0274</td>
<td>0.0254</td>
<td>-0.0005</td>
<td>-0.0032</td>
<td>0.00005</td>
</tr>
<tr>
<td>16x16</td>
<td>0.5862</td>
<td>0.0592</td>
<td>0.0362</td>
<td>0.0224</td>
<td>-0.0002</td>
<td>-0.0031</td>
<td>0.00006</td>
</tr>
<tr>
<td>MEAN</td>
<td>0.5427</td>
<td>0.0252</td>
<td>0.0262</td>
<td>0.0269</td>
<td>-0.0004</td>
<td>-0.0028</td>
<td>0.00005</td>
</tr>
<tr>
<td>STD DEV</td>
<td>0.0195</td>
<td>0.0154</td>
<td>0.0047</td>
<td>0.0023</td>
<td>0.0001</td>
<td>0.0003</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Slika 39: Huovi momenti na slici "R" pri različitim rezolucijama od 512x512 do 32x32

5.4. Rezultati Zernikeova momenta

U usporedbi sa Huovim momentima, prednost Zernikeovih momenata je da se mogu lakše izračunavati na višim redovima.
Iz slika (40) (41) se vidi da Zernikeovi momenti mogu već od desetog reda hvatati oblike pri niskim rezolucijama 64x64. Kad se red poveća na dvadeset, rekonstrukciju slike već može prepoznati i ljudsko oko bez greške. Što idemo prema većem redu rekonstrukcija je kvalitetnija i uključuje više detalja orginalne slike. Na 40-tom redu rekonstrukcije su skoro originalne i uključuju većinu glavnih svojstava znaka.

Slika 40: Raspodjela magnituda Zernikeovih momenata do reda 30 za sliku slova "B" pri rezoluciji 64x64

Slika 41: Rezultati rekonstrukcije Zernikeovim momentima sa redom 5,10, 20, 30, 40
6. ZAKLJUČAK

Kad se pogleda odakle je OCR krenuo i gdje je stigao, pogotovo zadnjih deset godina, te streloviti napredak u novim granama prepoznavanja na novim računalnim platformama, tabletima, fotoaparatima itd. možemo samo očekivati još bolja usavršavanja dosadašnjih i razvijanje novih algoritama za optičko prepoznavanje.

Primjena je sve šira, tehnologija i napredovanje sve brže da se nećemo morati brinuti da niskorezolucijske slike stvaraju problem. Kao što smo vidjeli već do sad razvijeni algoritmi uspiješno savladavaju sve prepreke čitljivosti do te mjere da se mogu mjeriti ljudskim okom. Mana uvijek ima ali ima i onih koji će potražiti pravo rješenje.

Algoritmi poput Fourierovih deskriptora i Huovih sedam momenata se oba mogu koristiti u svim današnjim OCR softvarima jer zadovoljavaju invarijaciju pogotovo velikih pisanih slova i pri težim transformacijama poput skaliranja, translatacije i rotacije. Vidjeli smo da nisu otporni na zacrnjenja ali sa kvalitetnim predprocesuiranjem bez problema se riješi i to. Također smo vidjeli relativno brzo opadanje prepoznavanja pri nižim rezolucijama, pogotovo na 64x64 gdje naglo opada na 53% i niže. Zaključak nas dovodi do toga da višim stupnjem izračunavanja dolazi do usporavanja obrade, a samim time i do sporijeg prepoznavanja. No nove generacije tehnologija brzih višejezgornih procesora bez problema provode prepoznavanje da niti ne zamijetimo usporavanja.

Vidjeli smo da su jedno od rješenja i algoritmi poput Zernikeovih momenata koji reduciraju nepotrebne informacije u predprocesu te zbog toga lakše savladaju prepoznavanje iako su samo po sebi samo invariantni na rotaciju no kao i inače, dobra predobrada potpomogne daljnjem procesu.

Kao što sam rekao na početku, mislim da budućnost donosi još brži razvoj jer već sad sigurno postoje algoritmi koji i više nego dobro odradjuju prepoznavanje. Kriju se u algoritmima novih softvera. Na nama je samo da čekamo i divimo se.
Popis slika i tablica

Slika 1: Podijela prepoznavanja znakova prema obradi (1)
Slika 2: Patent G.Tauschek – Uređaj za čitanje (2)
Slika 3: Izgled OCR-A i OCR-B fonta (3)
Slika 4: Beyo OCR softver – direktan prijenos slike u tekst (4)
Slika 5: Raspodijela pisma u svijetu (5)
Slika 6: Površina za unos elektronskih potpisa (6)
Slika 7: Degradacija slike sa 512x512 na 16x16 (7)
Slika 8: Prikaz razvučenih niskorezolucijskih slika na orginalnu veličinu (8)
Slika 9: Prikaz konvencionalnog pristupa OCR-u (9)
Slika 10: Shematski prikaz diagrama za pretraživanje dokumenata i rječnika (10)
Slika 11: Pripreme poravnavanja slike u predobradi (11)
Slika 12: Dvodimenzionalna geometrijska transformacija (12)
Slika 13: Komponente OCR sistema (13)
Slika 14: "trasholding" zacrnjenje pri skeniranju (14)
Slika 15: Degradacija simbola (15)
Slika 16: Proces normalizacije simbola (16)
Slika 17: Zoniranje (17)
Slika 18: Eliptični Fourierov deskriptor (18)
Slika 19: Umjetna neuronska mreža (19)
Slika 20: Kombinacije s malom vjerojatnošću (20)
Slika 21: Smjerovi rubnih piksela (21)
Slika 22: Lančani kod (22)
Slika 23: Osnovna slika simbola "A" i njen zatvoren oblik u Cartesianskom sustavu (23)
Slika 24: Primjer prikaza Fourierovih deskriptora (24)
Slika 25: Obrada slike s različitim konturama (25)
Slika 26: Set slikovnih primjera za prostornu rezoluciju od 512x512 do 16x16 (26)
Slika 27: Dijagram izračunavanja Fourierovih deskriptora (27)
Slika 28: Dijagram izračunavanja Huovih momenata (28)
Slika 29: Dijagram izračunavanja Zernikeova momenata (29)
Slika 30: Usporedba orginalne i normalizirane translatirane slike (30)
Slika 31: Usporedba orginalne i normalizirane skalirane slike (31)
Slika 32: Prikaz redukcije prostora mapiranja za Zernikeove momente (32)
Slika 33: OCR engine (33)
Slika 34: Radni proces OCR engine-a (34)
Slika 35: Prikaz klasifikacije slike "A" 128x128 rotirane za -45° (35)
Slika 36: Rasprostranjenost F.D. za sliku slova "O" sa rezolucijom 512x512 (36)
Slika 37: Graf prvih 10 FD slike slova "O" sa rezolucijom od 512x512 do 32x32 (37)
Slika 38: Graf prvih 10 FD slike slova "C" sa rezolucijom od 512x512 do 32x32 (38)
Slika 39: Huovi momenti na slici "R" pri različitim rezolucijama od 512x512 do 32x32 (39)
Slika 40: Raspodijela magnituda Zernikeovih momenata do reda 30 za sliku slova "B" pri rezoluciji 64x64 (40)
Slika 41: Rezultati rekonstrukcije Zernikeovim momentima sa redom 5,10, 20, 30, 40 (41)

Tabela 1: Procjena tehnika za izvlačenje značajki /1/
Tablica 2: Glavna svojstva Fourierovih deskriptora za neke transformacije /2/
Tabela 3: Prvih 10 Fourierovih deskriptora za slike 8 velikih slova abecede sa prostornom rezolucijom 512x512 /3/
Tabela 4: Prvih 10 FD slike slova "O" sa prostornom rezolucijom od 512x512 do 32x32 /4/
Tabela 5: Prvih 10 FD slike slova "C" sa prostornom rezolucijom od 512x512 do 32x32 /5/
Tabela 6: Huovih sedam momenata na slikama 8 slova abecede pri rezoluciji 512x512 /6/
Tabela 7: Huovih sedam momenata na slikama slova "R" sa prostornom rezolucijom od 512x512 do 32x32 /7/
LITERATURA


26. ?lng=hr&vie=ctl&gr1=strSvt&gr2=&id=2011040115341553 Srpanj 2012 [26]


