SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 858

PROGRAMSKI SUSTAV ZA RASPOZNAVANJE TISKANOG TEKSTA

Mladen Jurković

Zagreb, lipanj 2009.

Mladen Jurković, 0036428019

Sadržaj

Uvod

Prvi put kad se čovjek stvarno počeo razlikovati od životinja bio je dan kada je naučio govoriti. Govor je ono što nas razlikuje od životinja. Mogućnost razmjene informacija s drugim ljudima. Sljedeća stepenica u intelektualnom razvoju čovjeka bilo je pismo, mogućnost zapisa znanja i ideja. Čovjek nauči pisati vrlo rano u životu, oko svoje sedme godine. Nama to ne predstavlja problem. Ako smo naučili jedno pismo, na primjer latinicu, znati ćemo ju pročitati, bez obzira koja osoba je napisala neki tekst koristeći to pismo.

Za računala to i nije tako jednostavan zadatak. Da bi računalo moglo „pročitati“ neki tekst koji snimi kamerom ili skenerom, mora imati ugrađeni sustav za raspoznavanje znakova. Danas postoje izrazito sposobni sustavi za raspoznavanje tiskanog teksta, a napredak u tom području raspoznavanja znakova može se očekivati u raspoznavanju rukom pisanog teksta. Budući da je sve počelo od raspoznavanja tiskanog teksta, zbog razumijevanja ove teme i mogućnosti napredovanja u ovom području, ovaj rad se bavi izgradnjom programskog sustava za raspoznavanje tiskanog teksta.

1. Optičko raspoznavanje znakova

Optičko raspoznavanje znakova je proces prepoznavanja znakova, slova ili brojeva iz slike na kojoj se ti znakovi nalaze. U ovom poglavlju su opisana povijest optičkog raspoznavanja uzoraka, njezina poveznica sa srodnim disciplinama i opis nekih metoda za izradu sustava za optičko raspoznavanje znakova.

1.1. Povijest optičkog raspoznavanja znakova

Razvoj i istraživanje optičkog raspoznavanje znakova (engl. Optical character recognition - OCR) seže u daleke 1950-te godine kada su znanstvenici po prvi puta pokušali spremiti sliku znakova i teksta. Optičko raspoznavanje znakova je u počecima bio spor proces jer prvi skeneri nisu mogli uhvatiti više od jedne linije teksta s pokretne trake. Razvojem rotacijskih i „flatbed“ skenera, snimanje teksta je ubrzano i povećano na cijelu stranicu. Brži transport dokumenata i veća brzina skeniranja i digitalne pretvorbe omogućili su brži razvoj optičkog raspoznavanja znakova. Razvoj OCR aplikacija potpomoglo je i njihovo korištenje unutar raznih područja, korištene su u bankama, bolnicama, poštama i raznim drugim industrijama.

Razvoj sklopovlja pratila su i istraživanja optičkog raspoznavanja znakova unutar akademskih, ali i industrijskih sektora. U početku su OCR sustavi radili mnogo pogrešaka, pa se pribjeglo stvaranju novih standardiziranih fontova. Oblikovani su novi fontovi OCRA i OCRB u 1970-ima od strane American National Standards Institute (ANSI) i the European Computer Manufactures Association (ECMA). Ti fontovi su prihvaćeni od međunarodne organizacije International Organization for Standarization (ISO) što je uzrokovalo njihovo veće korištenje u poslovnim sustavima i time omogućilo veću razinu raspoznavanja OCR sustava. Danas postoje mnogi OCR sustavi koji s velikom preciznošću prepoznaju tiskani tekst. [Cheriet, 2007]

1.2. Računalni vid i raspoznavanje uzoraka

Optičko raspoznavanje uzoraka je područje istraživanja unutar znanstvenih disciplina računalnog vida i raspoznavanja uzoraka. Slijedi kratki pregled tih znanstvenih disciplina.

Računalni vid je znanstvena disciplina koja se bavi transformacijom podataka iz mirne ili pokretne kamere u odluku ili novu reprezentaciju podataka. Za prikaz slike u boji u računalu koriste se različiti sustavi za prikaz boja, RGB, HSL, HSV, CMYK. RGB sustav za prikaz boja je sustav kod kojeg je svaka točka slike prikazana s tri komponente boja, crvenom, zelenom i plavom. RGB slika je pohranjena kao dvodimenzionalna matrica dimenzija širina x visina u koju su pohranjeni trodimenzionalni vektori . Prosječno velika slika širine 640 i visine 480 pixela (pixel = element slike) zauzima više od 1.000.000 podataka. Obrade ovolike količine podataka u današnjim računalima ne predstavlja problem, ali često se u današnjim sustavim zahtijeva obrada velikog broja slika u malom vremenu. Ovaj problem se rješava pojednostavljenjem prikaza slike koja se obrađuje. Pretprocesiranje slike i izlučivanje značajki su faze sustava za optičko raspoznavanje teksta unutar kojih se ovaj problem uspješno rješava.

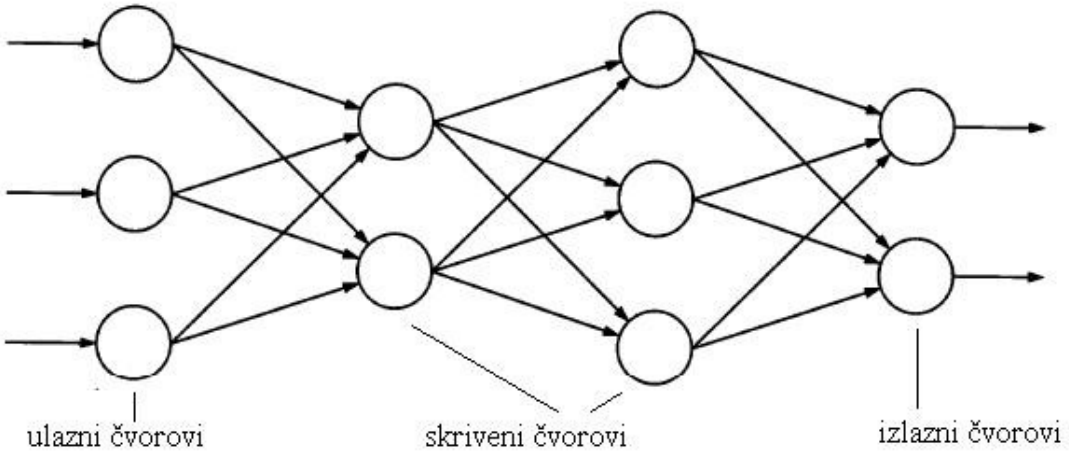
Druga disciplina unutar koje se optičko raspoznavanje znakova svrstava je raspoznavanje uzoraka. Raspoznavanje uzoraka je znanstvena disciplina koja razvrstava, klasificira promatrane objekte u kategorije ili klase. Objekt može biti slika, zvuk ili općenito neka veličina koja se može mjeriti. Generički naziv za objekte je uzorak. Tri su osnovna tipa raspoznavanja uzoraka. To su raspoznavanje uzoraka temeljeno na primjerima, raspoznavanje uzorak temeljeno na strukturi složenih uzoraka i grupiranje.[Tou, 1974] Raspoznavanje uzoraka temeljeno na primjerima još se naziva i učenje s učiteljem. Učenje s učiteljem podijeljeno je u dvije faze, fazu učenja i fazu odlučivanja. U fazi učenja učitelj označuje skup uzoraka za učenje i traži način kako uz najmanju pogrešku klasificirati uzorke u razrede. Ovaj tip, učenje s učiteljem korišten je u izradi ostvarenog programskog sustava.

1.3. Metode kod oblikovanja sustava za optičko raspoznavanje znakova

Postoje mnoge metode koje se koriste kod oblikovanja sustava za optičko raspoznavanje znakova, neke od njih su umjetne neuronske mreže i raspoznavanje uzoraka temeljeno na strukturi složenih uzoraka. Osnovna ideja prvo spomenute metode opisana je u sljedećem potpoglavlju.

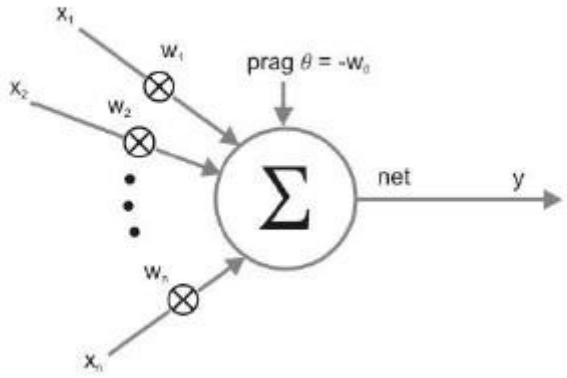
1.3.1. Umjetna neuronska mreža

Umjetna neuronska mreža je računalni sustav čija je ideja rada posuđena iz prirode. Osnovna ideja umjetnih neuronskih mreža jest oponašanje biološke neuronske mreže kako bi se riješio zadani problem. Model jednostavne umjetne neuronske mreže prikazan je na slici 1.



Slika 1 Model umjetne neuronske mreže

Umjetna neuronska mreža je mreža međusobno povezanih čvorova. Postoje tri vrste čvorova, unutarnji, vanjski i skriveni. U svakom čvoru obavlja se neka jednostavna računska operacija, a svaka veza između čvorova prenosi signal. Svaki čvor umjetne neuronske mreže predstavlja jedan neuron, model takvog umjetnog neurona prikazan je na slici 2.



Slika 2 Model umjetnog neurona

U modelu umjetnog neurona signali su opisani numeričkim iznosom i na ulazu u neuron množe se težinskim faktorom, tako dobiveni signali se nakon toga sumiraju i ako je dobiveni iznos iznad definiranog praga neuron daje izlazni signal.

Ulazni signali su označeni s $x_1, x_2, \ldots, x_n$, težinski faktori s $w_1, w_2, \ldots, w_n$, a prag s w_0.

Određivanje težinskih faktora događa se u procesu učenja, kod kojega postoje dva tipa, učenje s učiteljem i učenje bez učitelja. Nakon što se jednom od ovih metoda izgradi mreža čvorova i veza sustav je spreman za klasifikaciju uzoraka.

Klasifikacija uzoraka u umjetnim neuronskim mrežama je vrlo jednostavna. Na ulaz se dovede nepoznati uzorak, i ako i-ti izlazni čvor ima najveću vrijednost među izlaznim čvorovima, zaključuje se da uzorak pripada razredu ω_i. [Mehrotra, 1996]

2. Sustav za raspoznavanje tiskanog teksta

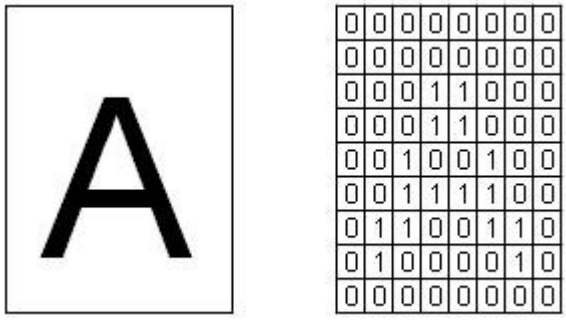
Sustav za raspoznavanje uzoraka je sustav koji neki promatrani objekt razvrstava u jedan od razreda. Sustav za raspoznavanje tiskanog teksta je sustav za raspoznavanje uzoraka kod kojega su uzorci slova na slici koja se obrađuje. Slova se razvrstavaju u jedan od 26 razreda (pretpostavljena je engleska abeceda koja sadrži 26 slova). Ovaj sustav sastoji se od nekoliko dijelova. Model sustava je prikazan na slici 3.

Slika 3 Model sustava za raspoznavanje tiskanog teksta

Dijelovi sustava za raspoznavanje tiskanog teksta su kamera, pretprocesiranje slike, izlučivanje značajki, odlučivanje i evaluacija sustava.[Cheriet, 2007] Svaki od dijelova sustava opisan je u sljedećim poglavljima.

2.1. Pretprocesiranje

Pretprocesiranje je proces u kojem se iz ulaznog podatka stvara novi podatak pogodan za obradu. Ulazni podatak u sustavu za raspoznavanje tiskanog teksta je slika. Ulazna slika se prvo pretvara iz slike u boji u sliku s nijansama sive boje. Prije je opisano koliko jedna slika sadrži podataka. Ovaj korak pretvaranja slike u boji u sivu sliku smanjuje količinu podataka za obradu na trećinu. Svaka točka slike više nije prikazana kao kombinacija sve tri komponente boja, nego s nijansom sive boje u toj točki. Ovako dobivena slika se procesom sedimentacije pretvara u crno-bijelu sliku. Sedimentacija je proces u kojem se boja svake točke, čija je nijansa sive boje veća od proizvoljno odabranog praga, pretvara u crnu ili ako je ispod praga u bijelu boju. Crni bit se zapisuje kao 1, a bijeli kao 0, i zato se ovako dobivena crno-bijela slika naziva binarna slika. Ovakva binarna slika je pogodna za daljnju obradu.



Slika 4 Binarna slika i njezin zapis u računalu

Slika 4 prikazuje takvu jednu binarnu sliku, svaki bit prikazan je crnom ili bijelom bojom, a zapis takve slike je dvodimenzionalna tablica s binarnim elementima.

2.2. Izlučivanje značajki

Izlučivanje značajki je bitan dio svakog sustava za raspoznavanje uzoraka. Problem koji se pojavljuje u izradi sustava za raspoznavanje uzoraka jest kako prikazati neki uzorak, koje značajke izabrati kod prikaza, da li postoje neke univerzalne značajke za sve tipove uzoraka.

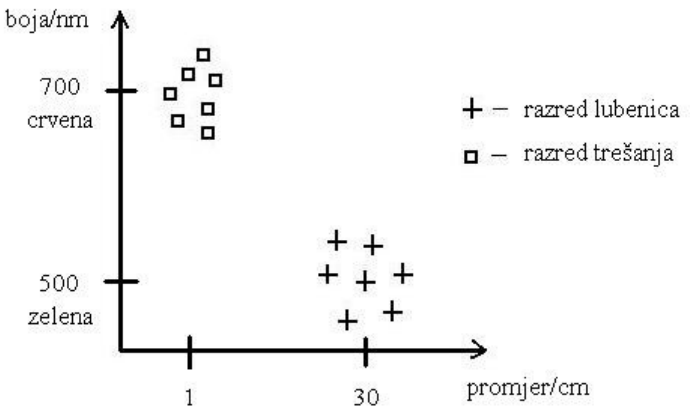
Izlučivanje značajki je faza u sustavu za raspoznavanje uzoraka u kojem se svaki uzorak prikazuje svojim vektorom značajki i tako priprema za jednostavniju i bržu odluku o pripadnosti tog uzorka nekom razredu.

2.2.1. Vektor značajki

Svaki uzorak želimo prikazati što jednostavnije, a opet dovoljno složeno kako bi se moglo prepoznati kojem razredu taj uzorak pripada. Zato se svaki uzorak prikazuje njegovim vektorom značajki. Značajke koje se odabiru su diskriminacijske, to su one koje odjeljuju taj uzorak od uzoraka iz nekog drugog razreda.

Vektor značajki $x = \{x_1, x_2, ..., x_n\}$, gdje je n dimenzionalnost prostora, a x_i su komponente tog vektora

Slijedi primjer uzoraka predstavljenih vektorom značajki. Pretpostavimo da imamo dva razreda uzoraka, razred lubenica i razred trešanja. Odabrali smo kao diskriminacijske značajke promjer i boju voća. Distribucija uzoraka u tom dvodimenzionalnom prostoru je prikazana na slici 5.



Slika 5 Prikaz vektora značajki uzoraka

Neki uzorak iz razreda trešanja ima kao vektor značajki vektor $x = [1,700]^T$, a neki uzorak iz razreda lubenica ima za vektor $x = [30,500]^T$. Na ovom jednostavnom primjeru pokazano je kako prikazati neki uzorak njegovim vektorom značajki.

2.3. Odlučivanje

Zadatak faze odlučivanja unutar sustava za raspoznavanje tiskanog teksta jest izgraditi klasifikator koji će razvrstati nepoznate uzorke u odgovarajuće razrede.

Sustav mora nepoznati uzorak prikazan svojim vektorom značajki x svrstati u jedan od M razreda koji se označavaju s $\omega_1, \omega_2, ..., \omega_M$. Neki od mogućih pristupa rješavanju ovog problema su Bayesova teorija odlučivanja, primjena linearnog klasifikatora i primjena nelinearnog klasifikatora. Osnovna ideja rada prvog pristupa objašnjena je u sljedećem poglavlju.

2.3.1. Bayesova teorija odlučivanja

Bayesova teorija odlučivanja je statistički pristup i u odlučivanju koriste vjerojatnosti $P(\omega_i), p(x|\omega_i), P(\omega_i|x)$. $P(\omega_i)$ je vjerojatnost pojave nekog uzorka iz razreda ω_i. Ova vjerojatnost još se naziva i *a priori* vjerojatnost. $p(x|\omega_i)$ je uvjetna gustoća razdiobe vjerojatnosti, još se naziva i vjerodostojnost. I konačno vjerojatnost $P(\omega_i|x)$ jest vjerojatnost da nepoznati uzorak pripada razredu ω_i. Vjerojatnost $P(\omega_i|x)$ računa se prema Bayesovoj formuli, uz pretpostavku da su ostale vjerojatnosti zadane.

$$P(\omega_i|\vec{x}) = \frac{p(x|\omega_i) \cdot P(\omega_i)}{p(x)}$$

gdje je $p(\vec{x}) = \sum_{i=1}^{M} p(\vec{x}|\omega_i) \cdot P(\omega_i)$

Formula 1 Bayesova formula uvjetne vjerojatnosti

Bayesovo pravilo odlučivanje za primjer od M = 2 razreda glasi ovako:

Ako $P(\omega_1|x) > P(\omega_2|x)$, onda x pripada razredu ω_1

Ako $P(\omega_2|x) > P(\omega_1|x)$, onda x pripada razredu ω_2

Ako je vjerojatnost da je uzorak x iz razreda ω_1 veća od vjerojatnosti da je uzorak x iz razreda ω_2, onda odlučujemo da uzorak pripada razredu ω_1, i obratno.[Theodoridis, 2003] Ovo pravilo odlučivanja je logično i najjednostavnije od pravila odlučivanja.

2.4. Evaluacija sustava

Evaluacija sustava je posljednji dio u sustavu za raspoznavanje tiskanog teksta. U ovoj fazi sustav je već odredio optimalni klasifikator za vektore značajki uzoraka za učenje. Cilj ove faze, evaluacije sustava jest odrediti koliko sustav griješi ili koliko je točan pri raspoznavanju nepoznatih uzoraka . Jedan od mogućih pristupa ovom problemu jest pristup izračuna greške. Cilj pristupa izračuna greške jest procijeniti vjerojatnost pogreške klasifikacije sustava. [Theodoridis, 2003] Kod ovog pristupa izračuna greške računa na koliko uzoraka za testiranje sustav treba biti testiran ako se želi dokazati određen postotak greške. Ako je pokazano da sustav ima 100% uspješnost, a testiran je na malom broju uzoraka, onda ova uspješnost nema temelj i ne može se koristiti ili očekivati kod uporabe takvog sustava.

3. Ostvareni programski sustav za raspoznavanje tiskanog teksta

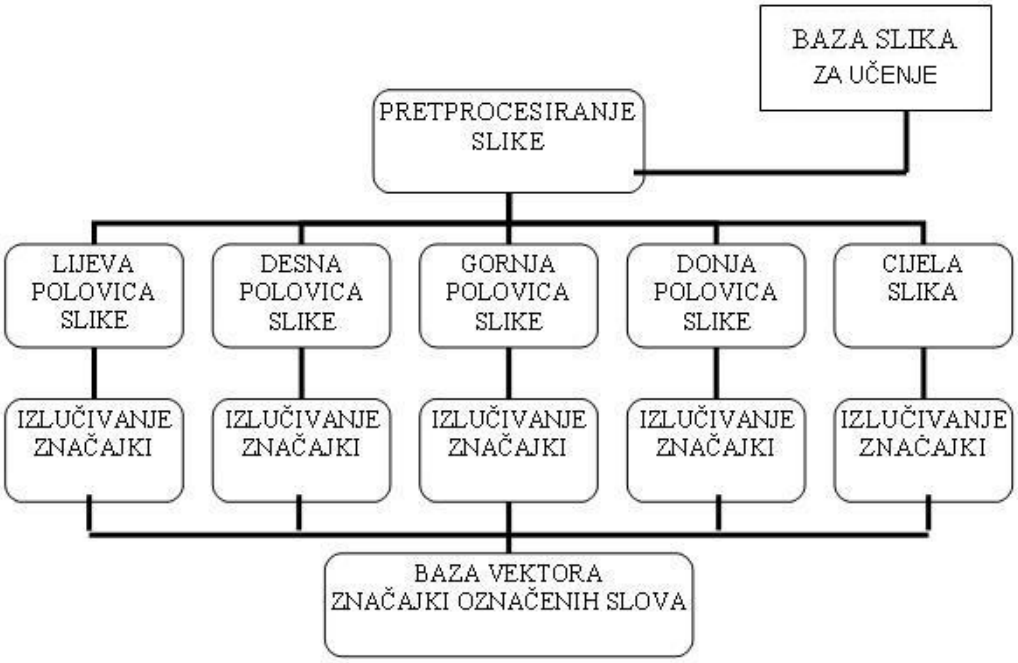
Ostvareni programski sustav je sustav za raspoznavanje temeljen na primjerima. To je sustav koji na temelju uzoraka za učenje određuje kojem razredu pripada nepoznati promatrani uzorak. Na temelju slike na kojoj se nalazi tekst, sustav prikazuje slovo koje se trenutno obrađuje, njegovu pripadnost nekom od razreda i na kraju cijeli prepoznati tekst zapisuje u tekstualnu datoteku kako bi ga korisnik mogao koristiti.

Raspoznavanje teksta je podijeljeno u dvije faze, fazu učenja i fazu odlučivanja. Učenje je dovoljno jednom pokrenuti s određenim skupom slika za učenje i sustav je pripremljen za raspoznavanje nepoznatih uzoraka. U sljedećim poglavljima opisani su faze učenja i odlučivanja.

3.1. Učenje u ostvarenom programskom sustavu

Učenje u ostvarenom programskom sustavu je ostvareno na oko 500 uzoraka, svaki razred uzoraka, svako slovo naučeno je s 19 uzoraka, odnosno s 19 različitih fontova. Fontovi koji su korišteni za učenje su redom: Times New Roman, Segoe Condensed, Arial, Lucida Sans Unicode, Garamond, Comic Sans MS, Microsoft Sans Serif, Courier New, Centhury Gothic, Gautami, Arial Narrow, Georgia, Latha, Arial Black, Bookman Old Style, Book Antiqua, System, Trebuchet MS i Verdana. Izabrani fontovi birani su slučajnim odabirom, jedino su Arial i Times New Roman izabrani jer su među najzastupljenijima u korištenju kod pisanja nekog teksta. Prepoznavanje ovih istih fontova prilično je uspješno, ali o tome će biti više rečeno kasnije u radu.

Učenje u ostvarenom programskom sustavu podijeljeno je u nekoliko faza, pretprocesiranje ulazne slike, izdvajanje označenog slova, stvaranje pet slika za obradu, izlučivanje značajki iz njih i na kraju zapis u bazu vektora značajki naučenih slova. Model sustava prikazan je na slici 6.



Slika 6 Model faze učenja

Slijede opisi svake od faza unutar učenja.

3.1.1. Izdvajanje označenog slova

Izdvajanje označenog slova temelji se na raspoznavanju kontura na slici. Kontura je niz točaka koje odjeljuju tamne od svijetlih područja na slici. Postoje dvije vrste, unutarnje i vanjske konture. Na slici 7 crvenom bojom su prikazane vanjske konture, a plavom unutarnje.

Slika 7 Prikaza vanjskih i unutarnjih kontura

Nakon što se ulazna slika postupkom pretprocesiranja pretvori u binarnu sliku, na slici se traže vanjske konture. Ulazne slike su prije pripremljene, tako je svaka vanjska kontura jedno slovo. Prostor unutar tako prepoznate slike izdvaja se iz slike i zapisuje u novu sliku, sliku

cijelog slova. Prije je spomenuto da se stvaraju pet slika za obradu. Cijela slika slova je jedna od slika, a sljedeće su slike svake od polovice slova, lijeva, desna, gornja i donja polovica slova. Odgovor na pitanje zašto se od jednog slova stvaraju pet slika leži u tome da sustav teži prema stopostotnoj preciznosti prepoznavanja nepoznatog slova, a samo na temelju sličnosti cjeline nepoznatog slova s nekim od poznatih slova sustav će griješiti. Naravno, ostvareni programski sustav nije nepogrešiv, ali kombinacijom ovih pet slika greška sustava se smanjila. Ove pripremljene slike pogodne su za izlučivanje značajki, što je sljedeći korak sustava.

3.1.2. Momenti

Prije nego se krene na fazu izlučivanja značajki par riječi o momentima. Momenti se koriste za uspoređivanje kontura. Momenti su definirani prema sljedećoj formuli

$$m_{p,q} = \sum_{i=1}^{n} I(x,y) x^p y^q \ ,$$

gdje su p,q potencije s kojom je svaka komponenta uzeta u ukupnoj sumi, a $I(x,y)$ predstavlja informaciju o točki (x,y), ako je slika binarna, onda je vrijednost $I(x,y) = \in \{0,1\}$

Iz ovako definiranih momenata računaju se središnji momenti prema formuli

$$\mu_{p,q} = \sum_{i=1}^{n} I(x,y)(x - x_{avg})^p (y - y_{avg})^q \ , \text{ gdje su}$$

$$x_{avg} = m_{10} / m_{00} \ \text{ i } \ y_{avg} = m_{01} / m_{00}$$

U daljnjem proračunu računaju se normalizirani momenti iz središnjih prema formuli

$$\eta_{p,q} = \frac{\mu_{p,q}}{m_{00}^{(p+q)/2+1}}$$

Momenti koji su korišteni u izradi ostvarenog programskog sustava za raspoznavanje tiskanog teksta su Hu-ovi momenti. Oni se računaju prema sljedećim formulama

$$h_1 = \eta_{20} + \eta_{02}$$

$$h_2 = (\eta_{02} - \eta_{20})^2 + 4\eta_{11}^2$$

$$h_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$h_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

Ovi momenti su korišteni jer su neovisni na translaciju, rotaciju i skaliranje konture objekta. Ova četiri Hu-ova momenta čine posljednje četiri komponente vektora značajki slova. [Bradski, 2008]

3.1.3. Izlučivanje značajki u ostvarenom programskom sustavu

Problem koji se pojavio pri izradi ovog sustava bio je kako brzo prepoznati slovo. Taj problem rješava izlučivanje značajki iz uzoraka, jer se tako sam zapis uzorka pojednostavljuje, čime se brzina prepoznavanja tako zapisanog uzorka povećava. Već je prije spomenuto da se svaki uzorak zapisuje svojim vektorom značajki. Tijekom izrade ovog sustava odlučeno je da će taj vektor biti iz prostora s pet dimenzija.

Kod prepoznavanja teksta koriste se još i dvije dodatne značajke, ali samo kako bi se odijelilo slovo od ne-slova. To su širina i visina koje ako su za zadani promatrani uzorak unutar očekivanih veličina zaključuje se da je taj uzorak slovo, pa se on dalje promatra i svrstava unutar jednog od 26 razreda.

Nakon što sustav pronađe kontura promatranog uzorka, odlučuje da li je to moguće slovo, i zatim stvara vektor značajki tog uzorka. Vektor značajki je 5-dimenzionalan, gdje je prvi podatak broj unutarnjih kontura promatranog uzorka, a sljedeći podaci su Hu-ovi momenti. Hu-ovi momenti opisani su u prethodnom potpoglavlju. . Prvi podatak u vektoru značajki ima najveću težinu kod prepoznavanja, ostali su jednake vrijednosti. Time se postiglo da nikad neće uzorak koji ima jednu unutarnju konturu biti zamijenjen za uzorak koji ih nema. Unutarnje i vanjske konture su prikazane na slici 7.

Tablica 1 Vektori značajki nekih slova

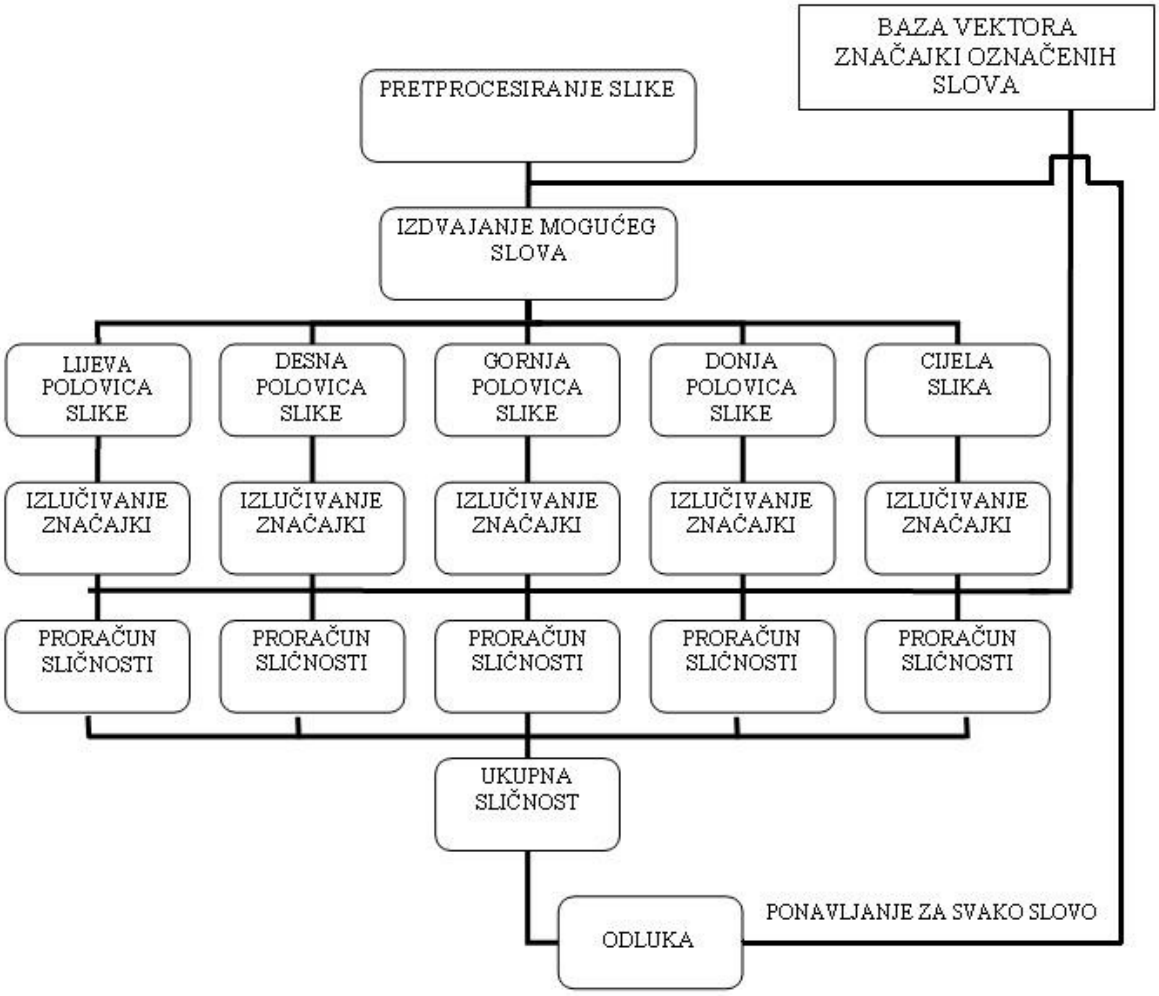
Slovo	Vektor značajki, $x = \{x_1, x_2, x_3, x_4, x_5\}$				
	x_1	x_2	x_3	x_4	x_5
A	1	0.258339	0.000210804	0.014233	0.000783863
B	2	0.166732	0.000348205	5.96992e-005	2.56154e-008
…	…	…	…	…	…
Z	0	0.440966	0.0246968	0.00116988	0.000194616

Ovo je prikaz vektora značajki za cijelu sliku slova, budući da se u ostvarenom programskom sustavu koriste i slike svake od polovica slova, svaka od polovica ima zapis vektora značajki

u svojoj tablici. Pri prikazu vektora u tablici su korištene tipične vrijednosti vektora značajki za slovo A, B i Z, ali ustvari cjelokupna baza vektora značajki ima zapis vektora za svaki od naučenih uzoraka. Za učenje ovog sustava korišteno je 19 različitih fontova, tako da je svako slovo prikazano s 19 različitih vektora značajki.

3.2. Odlučivanje u ostvarenom programskom sustavu

Odlučivanje u ostvarenom programskom sustavu podijeljeno je u nekoliko faza kao i učenje. Prva faza je pretprocesiranje slike, pa slijede faza izdvajanja mogućeg slova, stvaranje pet slika za obradu, izlučivanje značajki, računanje sličnosti, računanje ukupne sličnosti i konačno odluka.



Slika 8 Model faze odlučivanja

Ono što se razlikuje od procesa učenja faze su računanja sličnosti svake od pet slika promatranog nepoznatog uzorka s naučenim uzorcima, računanja ukupne sličnosti i odluke. U sljedećim potpoglavljima opisane su te faze.

3.2.1. Udaljenost i sličnost vektora

Sličnost vektora je u ostvarenom sustavu definirana je preko njihove udaljenosti. Što je udaljenost manja vektori su sličniji. Udaljenost između dva N-dimenzionalna vektora računa se prema formuli

$$d(x_A, x_B) = \|x_A - x_B\|, \text{ gdje je } \| \ \| \text{ norma vektora}$$

U ovom sustavu nije korišten ovaj tip udaljenosti jer nije bio pogodan zbog oblika vektora značajki kod kojega su članovi različitih redova veličina, zato je ova formula modificirana na

$$d(\vec{x}_A, \vec{x}_B) = \sum_{i=1}^{n} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right|, \text{ gdje je } n = 5, \text{ dimenzionalnost vektora značajki}$$

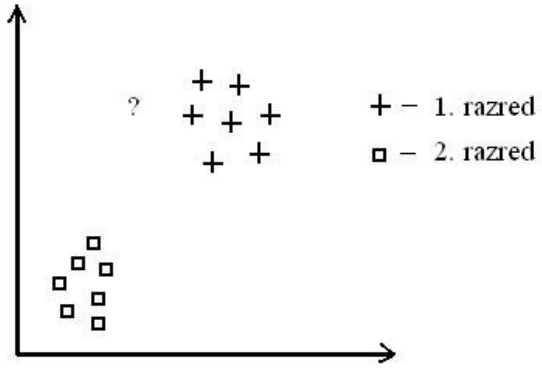
$$m_i^A = sign(x_i^A) \cdot \log \left| x_i^A \right|, \text{ a } x_i^A \text{ je i-ti član vektora značajki uzorka A}$$

Logaritmiranjem članova vektora značajki se postigla normalizacija, čime je svaka značajka dobila jednaku vrijednost u proračunu sličnosti. Ta sličnost između vektora se računa za svaku od polovica slika, i onda se konačna sličnost računa kao kombinacija tih sličnosti.

3.2.2. Metoda najbližeg susjeda

Metoda najbližeg susjeda, još se naziva 1-NN metoda, je metoda kojom se na temelju jednog najbližeg susjeda odlučuje kojem razredu pripada nepoznati promatrani uzorak. Slijedi primjer 1-NN metode.

Recimo da imamo uzorke koji mogu pripadati jednom od 2 razreda i da su ti uzorci prikazani vektorom značajki od dvije komponente. Distribuirani su u dvodimenzionalnom prostoru kao na slici 9.



Slika 9 Metoda najbližeg susjeda

Naš promatrani nepoznati uzorak prikazan je upitnikom (?) na slici 9. Računa se udaljenost od nepoznatog uzorka do svakog uzorka iz oba razreda. Gleda se kojem razredu pripada najbliži uzorak i odlučuje se da promatrani uzorak pripada prvom razredu, budući da je mu najbliži uzorak iz prvog razreda.1-NN metoda je specijalizirani slučaj k-NN metode. K-NN metoda je metoda kod koje se promatraju k najbližih susjeda i na temelju najviše predstavnika nekog razreda odlučuje da promatrani uzorak pripada tom razredu.

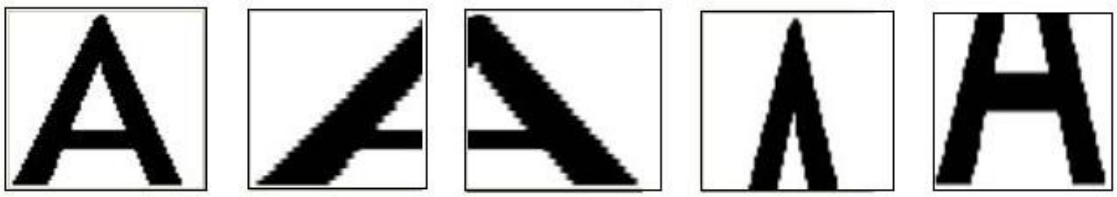
3.2.3. Računanje ukupne sličnosti i odluka o pripadnosti razredu

Nakon što su izračunate sličnosti nepoznatog uzorka, njegovih pet slika sa svakim slovom računa se ukupna sličnost. Ukupna sličnost nepoznatog uzorka $x_?$ s uzorcima iz svakog od razreda računa se prema sljedećoj formuli

$$ukupna(x_?, x_i) = lijeva(x_?, x_i) \cdot desna(x_?, x_i) \cdot gornja(x_?, x_i) \cdot donja(x_?, x_i) \cdot cijela(x_?, x_i)^f,$$

gdje je $x_?$ nepoznati promatrani uzorak, a x_i uzorak iz razreda $i \in \{A, B, ..., Z\}$ i $f = 1.5$

Broj f je potencija važnosti sličnosti promatranog uzorka s cijelim slovom, pokazalo se kroz mjerenja da raspoznavanje daje dobre rezultate uz vrijednost $f = 1.5$.



Slika 10 Prikaz cijelog i svih polovica slova

Svaka od pojedinih sličnosti računata je prema formuli

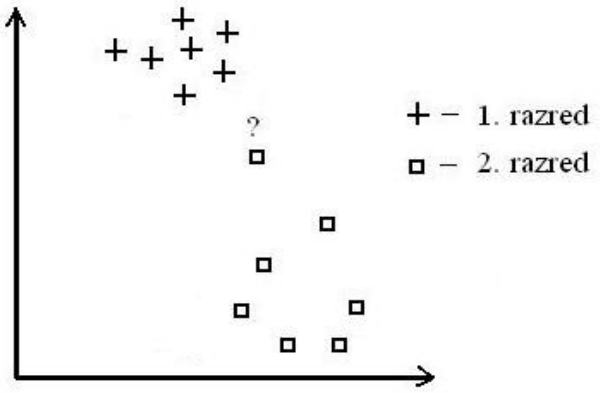
$slicnost(x_?, x_i) = \min(d(x_?, x_i)), j \in \{1,2,...,broj\_uzoraka_i\}$, a $broj\_uzoraka_i$ je broj uzoraka u razredu i, u ostvarenom sustavu taj je broj za svaki od razreda 19

Samo je minimalna sličnost svake od slika nekog slova sa svakom od slika nepoznatog uzorka korištena u proračunu ukupne sličnosti.

Kada se izračunaju sve ukupne sličnosti pripadnost uzorka nekom razredu određuje se 1-NN metodom, znači uzorak pripada onom razredu čijem razredu je najsličniji. Kod mjerenja pogreške ovako ostvarenog sustava ustvrđeno je da bi se mogla smanjiti. Zbog toga je dodana i mjera prosječne sličnosti. Ukupna prosječna sličnost računata je prema istoj formuli, a prosječne sličnosti svake od slika su računate prema sljedećoj formuli

$$prosjecna\_slicnost(x_?, x_i) = \frac{1}{broj\_uzoraka_i} \cdot \sum_{j=1}^{broj\_uzoraka_i} d(x_?, x_j)$$

Problem koji se htio riješiti dodavanjem prosječne sličnosti raštrkanost uzoraka nekog razreda. Ova je pojava je objašnjena na sljedećem primjeru. Neka uzorci pripadaju samo jednom od dva moguća razreda, i neka su prikazani s vektorom značajki od dvije komponente kao na slici 11.



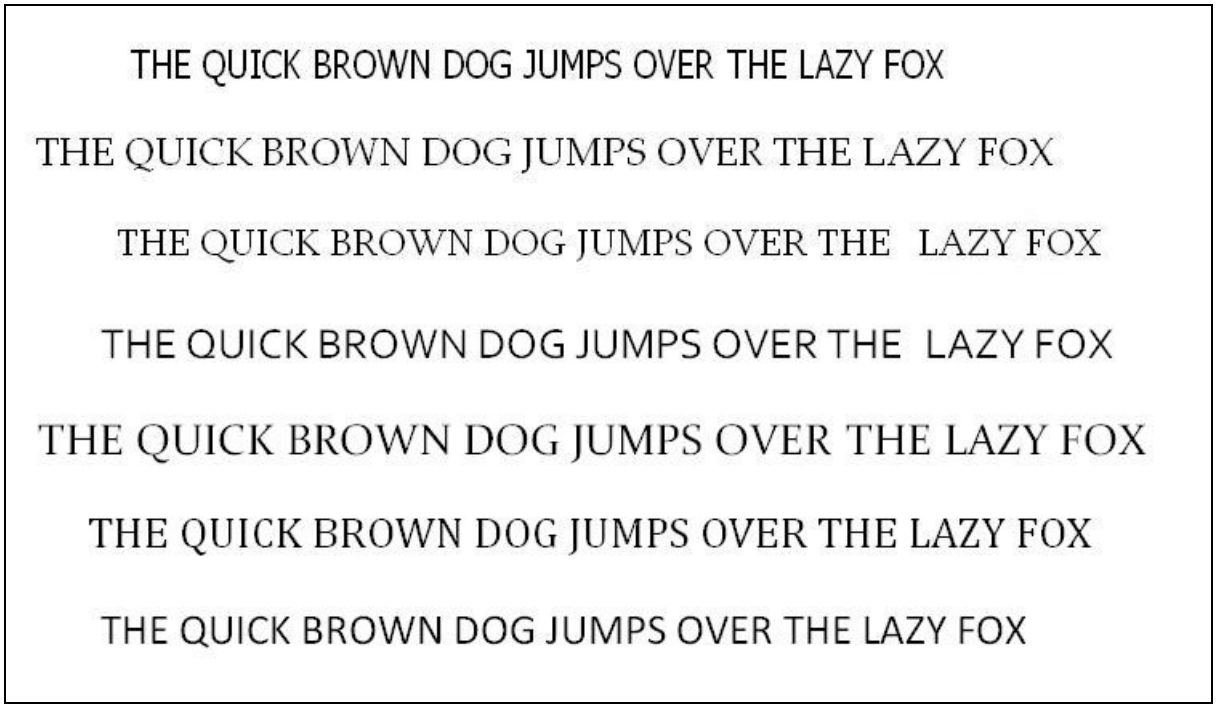
Slika 11 Raštrkanost uzoraka jednog razreda

Zbog jednog slučajno zalutalog uzorka iz skupa 2, nepoznati uzorak će biti svrstan u razred 2, makar je u prosjeku taj nepoznati uzorak bliže razredu 1. Dodavanjem prosječne sličnosti ova pojava je spriječena.

Nakon što su izračunate obje sličnosti, ukupna i prosječna sličnost se uspoređuju i odlučuje se kojem razredu pripada nepoznati uzorak.

3.3. Testiranje i uspješnost programskog sustava

Ostvareni programski sustav je za učenje koristio slike slova 19 različitih fontova koji su nabrojani kod faze učenja. Sustav je testiran na slikama cijelog teksta pisanog u sljedećim fontovima: Calibri, Cambria, Constantia, Corbel, Nina, Palatino i Sylfaen. Sustav je testiran na abecedama svakog od fontova (slika 12) i testiran je na tekstu od otprilike 400 znakova(slika 14). Prepoznavanje se otežalo jer se nije zahtijevalo od sustava da prepozna pojedinačno slovo nego da ga izdvoji iz teksta. Pojave poput sljepljivanja slova (slika 13a) su onemogućavale ovo prepoznavanje.



Slika 12 Primjerak slike za testiranje

. Na slici 12 sustav je testiran i dao je sljedeće rezultate:

THE QUICK BROMN DOG JUMPS OVER THE AZY FON

THE QUECK BMOWN DOG JUMFS OVES THE LAY FOX

THE QUICK BROKN DOC JUMPS OVER THE LAY FOZ

TNE QUICK BROWN DOG JUMPS OVER TNE LAZY FOX

TME QUICK BRONS SOG JUMPS OVER TME AY FOX

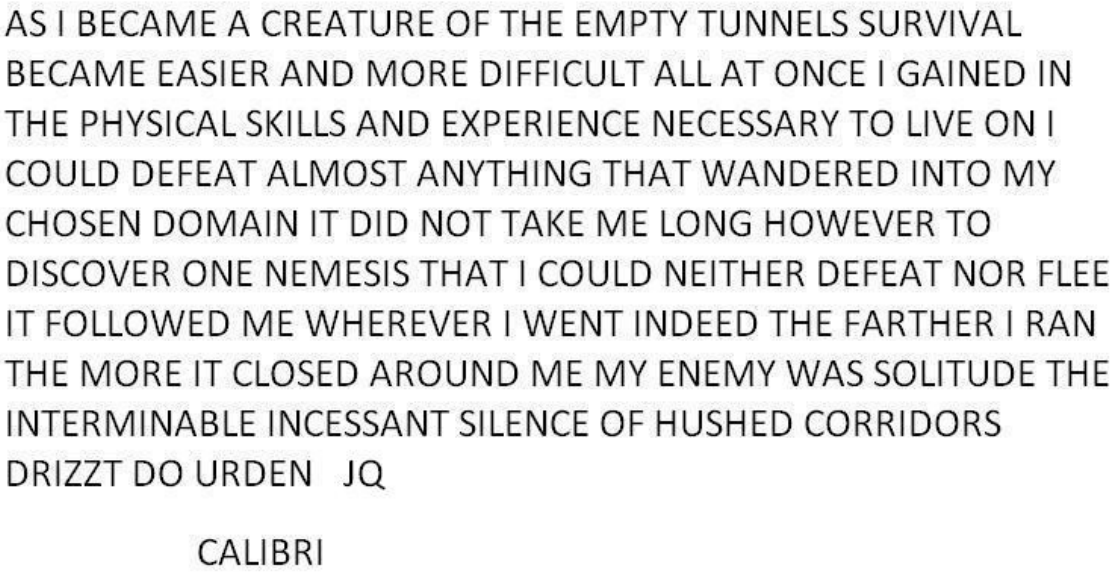
TGE QUICE BROSM DOG JXMPS OVER TME LAZY FON

TNE QUICK BROMN DOG JUMPS OVER TNE AZV FOX

Iz rezultata daje se zaključiti da sustav vrlo dobro prepoznaje slova, osim što zna zamijeniti neka slična slova kao što su M i W, H i N, V i Y. Još jedan od tipova greške koji se pojavljuje kada naiđe na slovo kojem nisu zatvorene konture kao na slici 13b, budući da je jedna od značajki u opisu uzoraka i broj unutarnjih kontura, a slovo ih nema, kod usporedbe neće biti slično odgovarajućem slovu.

Slika 13 a) Sljepljivanje slova b) Nezatvorenost kontura c) Razlomljeno slovo

Nesposobnost prepoznavanja slova pojavljuje se i kod pojave sljepljivanja koja je prikazana na slici 13a. Kada su slova međusobno slijepljena sustav neće moći prepoznati koja su to slova. Ovaj problem bi se mogao riješiti prije prepoznavanja tako što bi se mjerila prosječna širina slova, i pretpostavljalo ako je slovo više puta šire od prosječne širine da su to onda više slova u jednom. Naravno, ovdje postoji problem slova W koje je samo po sebi puno šire od ostalih i većinom se sastoji od dva slova V. Ovdje je sustav testiran na malom broju znakova i ne može se izračunati njegova uspješnost prepoznavanja, tako da je testiran i na tekstu (slika 14) od otprilike 400 znakova. Ovaj tekst(slika 14) je napisan na svakom od prethodno spomenutih fontova i uspješnost sustava je prikazana tablicom 2. Problemi kod prepoznavanja koji su se pojavili su prethodno spomenuto sljepljivanje slova, razlomljena slova (slika 13c) i krivo postavljanje slova ili novog reda u tekst. Uspješnost je računana na slovima koja nisu bila slijepljena, odnosno na onim slovima koje je sustav mogao točno prepoznati. Ovako izračunata uspješnost daje pogrešku od 5.66% što je zadovoljavajuće za sustav za prepoznavanje tiskanog teksta. Uspješnost bi se mogla poboljšati dodavanjem rječnika određenog jezika, čime bi se većina pogrešaka eliminirala.



Slika 14 Tekst za testiranje

Tablica 2 Prepoznavanje slova nepoznatih fontova

Ukupno/krivo	Calibri	Cambria	Constantia	Corbel	Nina	Palatino	Sylfaen
A	27/0	29/1	22/7	28/0	26/26	29/0	25/0
B	4/0	4/0	3/0	4/0	3/0	3/0	3/0
C	13/0	17/0	15/0	17/0	15/0	122/0	16/0
D	25/0	25/0	25/0	25/0	25/0	25/0	25/0
E	66/0	66/0	66/0	67/0	60/0	57/8	67/0
F	14/0	14/0	14/0	14/0	14/0	14/0	15/0
G	3/0	3/0	3/0	3/0	3/0	3/0	3/0
H	16/0	16/12	15/0	16/10	16/0	16/0	16/0
I	35/0	35/0	34/5	34/0	27/0	34/5	34/0
J	1/0	1/0	1/0	1/0	1/0	1/0	1/0
K	2/0	2/0	1/0	2/0	2/0	2/0	2/0
L	17/0	19/0	19/0	20/0	18/0	20/0	20/0
M	15/0	16/14	14/0	15/0	15/1	15/0	15/0
N	34/0	34/0	36/2	34/0	36/0	35/3	35/0

O	28/0	28/0	29/0	29/0	28/0	29/0	28/0
P	3/0	3/0	3/0	3/0	3/0	4/0	3/0
Q	1/0	1/0	1/0	1/0	1/0	1/0	1/0
R	26/0	25/0	23/1	36/0	25/0	24/5	23/1
S	20/0	22/0	21/0	22/1	12/0	22/0	190
T	31/0	32/0	34/3	33/2	25/0	32/0	3/0
U	10/0	10/0	10/0	10/0	10/5	10/0	10/0
V	6/0	6/0	6/0	6/0	6/0	2/1	6/0
W	6/4	6/0	6/6	6/3	6/5	6/6	6/6
X	1/1	1/1	1/0	1/0	1/1	1/1	1/0
Y	4/4	6/5	6/0	6/0	4/2	6/1	8/1
Z	2/0	2/0	2/0	2/0	0/0	1/0	2/0
Spojeno	7	2	9	0	21	12	5
Kriva pozicija	4/1	0/0	0/0	0/0	0/0	1/0	2/0
Ukupno	10/414	33/423	24/410	16/427	40/382	30/404	8/417

U sljedećem poglavlju slijede upute za instalaciju programske potpore.

3.3.1. Instalacija programske potpore

Razvijeni programski sustav napisan je u jeziku C++, stoga je potrebno osigurati razvojno okruženje koje podržava prevođenje tog programskog jezika. Testiranje je provedeno na Microsoftovom Visual Studiju 2008, te Visual Studiju C++ Expressu.

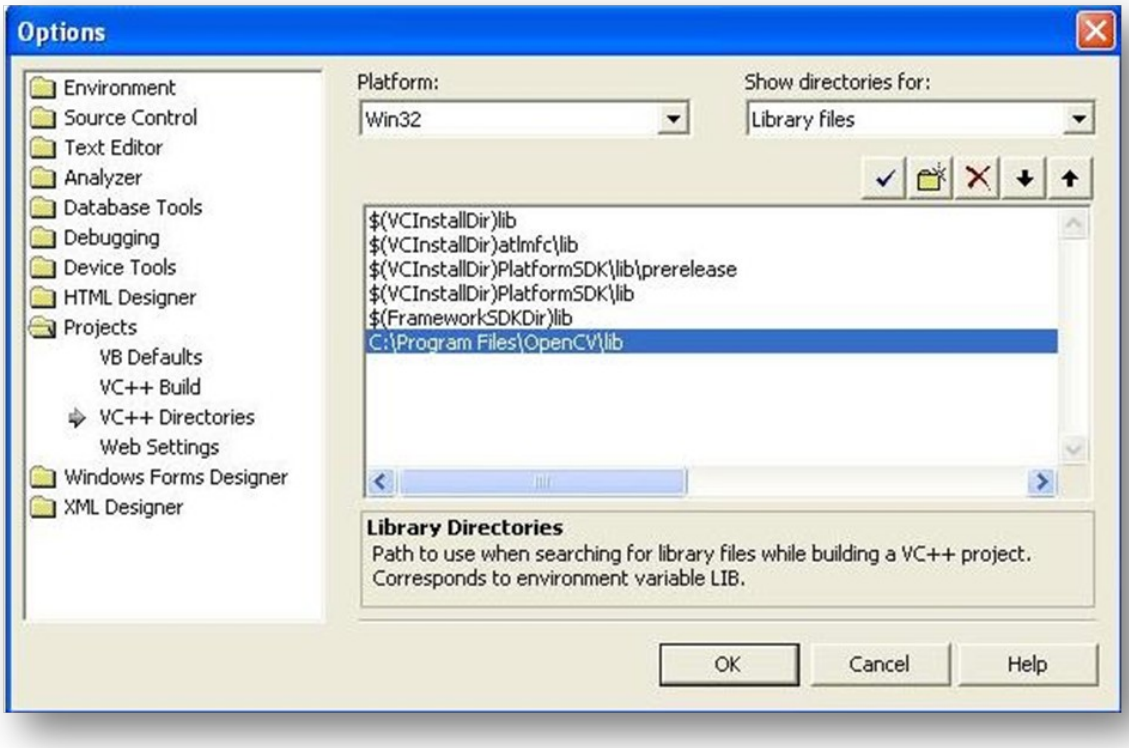
Također je potrebno osigurati OpenCV (*engl.* Open Source Computer Vision Library). OpenCV je biblioteka C i C++ popularnih implementacija funkcija za obradu slika i računalni vid. [Bradski, 2008]

Poželjno je instalirati OpenCV nakon IDE okruženja jer će čarobnjak OpenCV-a sam konfigurirati razvojno okruženje. OpenCV moguće je dohvatiti sa http://sourceforge.net/projects/opencvlibrary ili utipkavanjem OpenCV u Google. (Sourceforge nije uvijek dostupan.)

U slučaju da čarobnjak nije ispravno konfigurirao IDE potrebno je kopirati sve .dll datoteke iz OpenCV arhive u Windows/System32. Nakon toga potrebno je vratiti se u Visual Studio i otvoriti izbornik Tools → Options.

U listi treba otvoriti direktorij Projects → VC++ Directories, te odabrati u Library files i dodati direktorij (Slika 15):

"C:\Program Files\OpenCV\lib".



Slika 15 Prikaz opcija koje treba podesiti u Library files izborniku

Potom odabrati Include Files i dodati sljedeće direktorije:

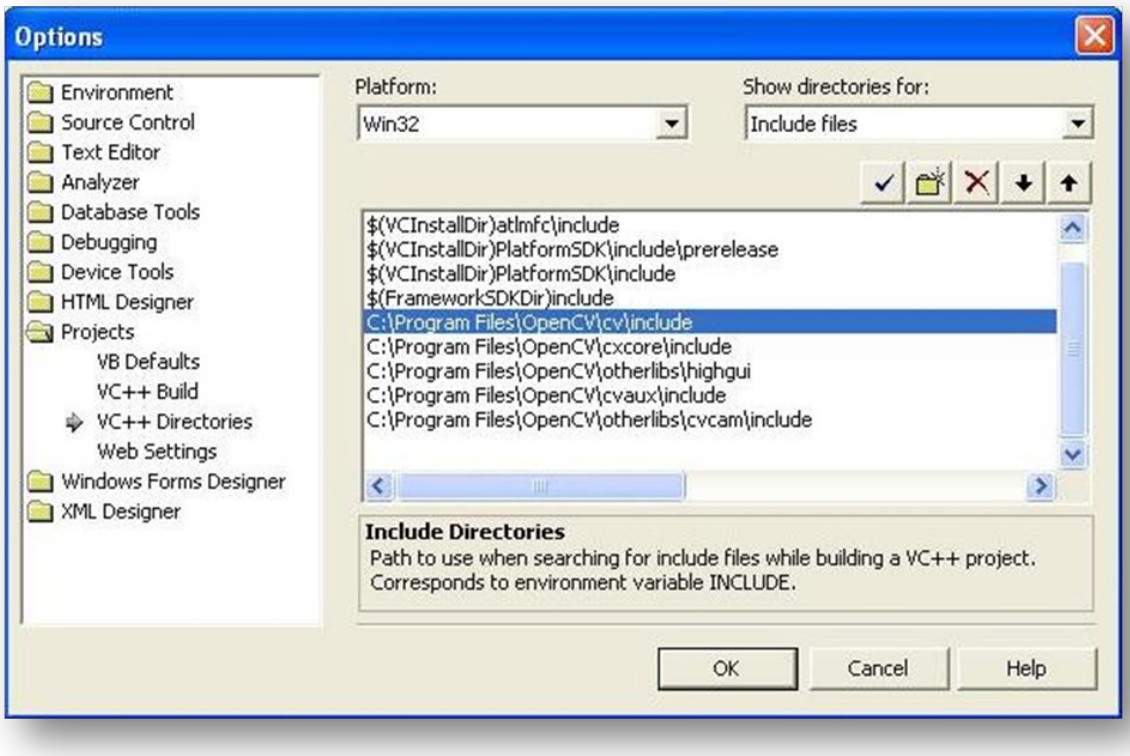
"C:\Program Files\OpenCV\cv\include"

"C:\Program Files\OpenCV\cxcore\include"

"C:\Program Files\OpenCV\otherlibs\highgui"

"C:\Program Files\OpenCV\cvaux\include"

"C:\Program Files\OpenCV\otherlibs\cvcam\include".



Slika 16 Prikaz opcija koje treba podesiti u Include files izborniku

Zatim odabrati Source Files i dodati sljedeće direktorije:

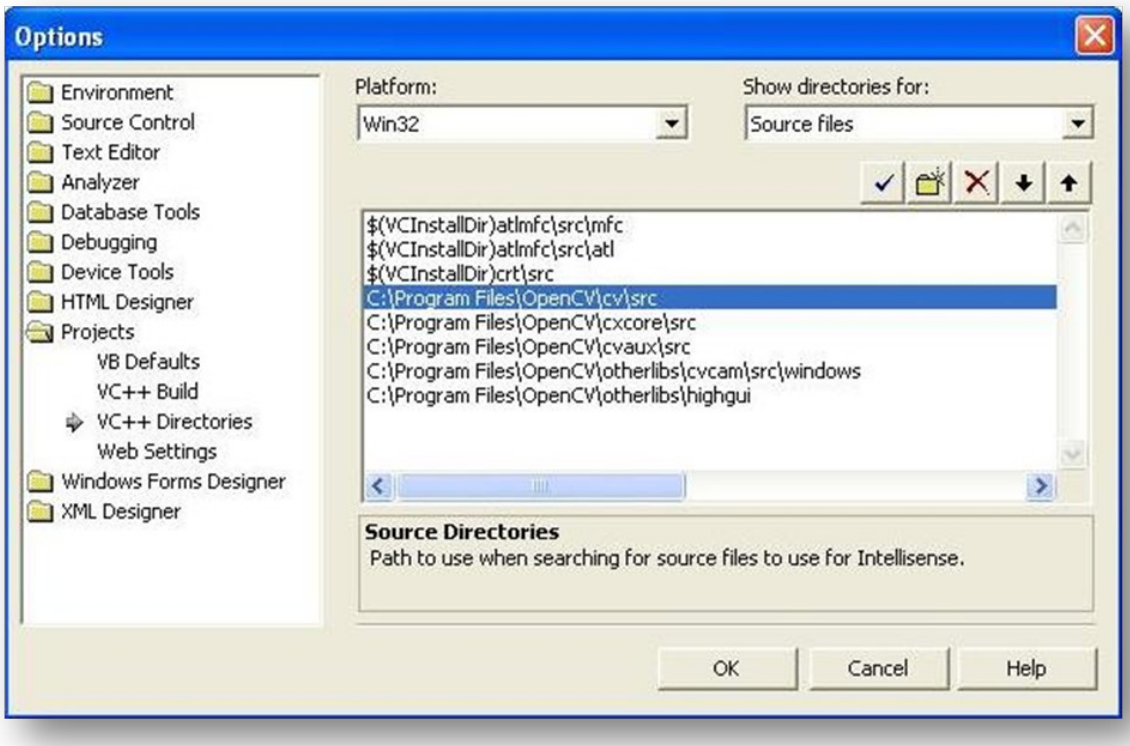
"C:\Program Files\OpenCV\cv\src"

"C:\Program Files\OpenCV\cxcore\src"

"C:\Program Files\OpenCV\cvaux\src"

"C:\Program Files\OpenCV\otherlibs\highgui"

"C:\Program Files\OpenCV\otherlibs\cvcam\src\windows".



Slika 17 Prikaz opcija koje treba podesiti u Source files izborniku

Nakon toga pritisnite OK i IDE okruženje bit će konfigurirano. Još detaljnije informacije mogu se naći na http://opencv.willowgarage.com/wiki/VisualC%2B%2B.

4. Zaključak

U uvodu ovog rada rečeno je da raspoznavanje tiskanog teksta ne predstavlja ljudima problem, ali nije toliko jednostavno računalima. Ovim radom je pokazano da izrada jednog uspješnog sustava za raspoznavanje tiskanog teksta zahtijeva puno teorijske podloge, ali i da je moguće napraviti takav sustav što potvrđuju rezultati ostvarenog programskog sustava za raspoznavanje tiskanog teksta. Točnost ostvarenog sustava je preko 94%. Uz moguća poboljšanja, uvođenje rječnika i prepoznavanje spojenih slova, greška ovako ostvarenog sustava bi se još smanjila.

Teorijska podloga dana je u prva dva poglavlja ovog rada. U prvom poglavlju opisana je osnovna ideja optičkog raspoznavanja znakova, njezini počeci, razvoj i poveznice s znanstvenim disciplinama računalnog vida i raspoznavanja uzoraka. U drugom poglavlju je opisan model jednog sustava za raspoznavanje tiskanog teksta. Pretprocesiranje, izlučivanje značajki, odlučivanje i evaluacija sustava su dijelovi jednog sustava za raspoznavanje tiskanog teksta i njihova važnost i način ostvarenja su opisani i objašnjeni u tom poglavlju. U zadnjem poglavlju rada opisan je uspješno ostvareni programski sustav za raspoznavanje tiskanog teksta. Na konkretnom sustavu opisane su metode koje su korištene pri njegovoj izradi kao i njegovi dijelovi. Ostvareni programski sustav napisan je u programskom jeziku C++ uz korištenje funkcija iz knjižnice OpenCv-a. Sama izrada ovog sustava trajala je par tjedana jer je puno vremena oduzelo podešavanje optimalnih parametara, pronalaženje diskriminatornih značajki i pripremanje uzoraka za učenje i testiranje. Ostvareni sustav podijeljen je u faze učenja i odlučivanja. Sustavu za učenje treba oko jedne minute, a za prepoznavanje jedne stranice tiskanog teksta malo više od minute.

Sustav je testiran na tekstu pisanom fontovima na kojima nije učen. Provedeno testiranje ostvarenog sustava daje grešku sustava od 5.66%, bez računanja neprepoznatih slijepljenih slova, što govori da je ostvareni programski sustav vrlo uspješan pri raspoznavanju slova nepoznatih fontova.

5. Literatura

[1] Cheriet M., Kharma M., Liu C., Sue C.: Character Recognition Systems, Wiley, 2007.

[2] Tou J., Gonzalez R.: Pattern Recognition Principles, Addison-Wesley Publishing Company, 1974.

[3] Mehrotra K., Mohan C., Ranka S.: Elements of Artificial Neural Networks, The MIT Press, 1996.

[4] Theodoridis S., Koutroumbas K.: Pattern Recognition, 2nd Edition, Elsevier Academic Press, Elsevier, 2003.

[5] Bradski G., Koehler A.: Learning OpenCV, O'Reilly Media, Sebastopol, 2008.

6. Naslov, sažetak i ključne riječi

Programski sustav za raspoznavanje tiskanog teksta

Sažetak

Ovaj rad se bavi izradom sustava za optičko raspoznavanje znakova, točnije rečeno raspoznavanje tiskanog teksta. U radu je opisano što je to, kako i kada je nastalo optičko raspoznavanje znakova. Također su opisane i neke popularne metode kod izrade programskih sustava za raspoznavanje znakova poput umjetnih neuronskih mreža. Budući da je optičko raspoznavanje znakova područje istraživanje unutar znanstvenih disciplina računalnog vida i raspoznavanja uzoraka, opisane su osnovne ideje i osnovni pojmovi ovih disciplina. U sklopu ovog rada izrađen je konkretan programski sustav za raspoznavanje tiskanog teksta, tako da su u radu opisani njegovi dijelovi i faze od kojih se sastoji. Sustav je podijeljen u dvije faze, učenje i odlučivanje. Uzorci, odnosno slova su prikazana 5-dimenzionalnim vektorom značajki. Vektor značajki se sastoji od momenata izračunatih iz slova. Odlučivanje je izvedeno pomoću udaljenosti vektora i metode najbližeg susjeda. I konačno, testiranje ovog sustava je također provedeno i rezultati su navedeni pri kraju rada.

Ključne riječi: raspoznavanje tiskanog teksta, optičko raspoznavanje znakova, računalni vid, raspoznavanje uzoraka, klasifikacija uzoraka, umjetna neuronska mreža, strukturalno raspoznavanje, pretprocesiranje, izlučivanje značajki, odlučivanje, evaluacija sustava, vektor značajki, momenti, konture, metoda najbližeg susjeda

7. Title, summary and keywords

Software for printed text recognition

Summary

The subject of this thesis is construction of an optical character recognition system, more precisely construction of a system for printed text recognition. This thesis describes what is the system for optical character recognition, how was it invented and when was the system first introduced. Also, some popular methods, used for developing programming systems for character reconition, are described, such as artificial neural networks. Because optical character recognition is a field of research within scientific disciplines of computer vision and pattern recognition, basic ideas and concepts of those disciplines are also described. As a practical part of this thesis a system for printed text recognition was developed and its parts and phases explained. The system is divided in two phases, learning and deciding in which class the pattern belongs. Patterns, which are in fact printed letters, are presented by 5-dimensional feature vectors. Feature vector is built from moments that were calculated from the letters. Deciding to which class the letter belongs is done with the method of vector distance and the nearest neighbour. Finally, the system was tested and the results are presented at the end of the thesis.

Keywords: printed text recognition, optical character recognition, computer vision, pattern recognition, pattern classification, artificial neural network, structural recognition, preprocessing, feature extraction, decision, system evaluation, feature vector, moments, contours, the nearest neighbour method