

Edin Tabak, MA
Dr. sc. Nevzudin Buzadžija

VJEŠTAČKA INTELIGENCIJA U FUNKCIJI PREPOZNAVANJA OBJEKATA I ZNAKOVA

Sažetak

U radu će biti pojašnjen proces prepoznavanja znakova na registarskim tablicama automobila. Prvi dio rada odnosi se na teorijske aspekte algoritma za dobijanje znakova sa registarskih tablica. Bit će pojašnjen proces od samog dobijanja slike, pripremanja slike za obradu, pronalazak mjesta registarske tablice na slici i na kraju prepoznavanja znakova registarske tablice. Drugi dio rada se sastoji od praktičnog prikaza rada programa. Prikazat će se koraci algoritma na primjeru jedne slike i zbirni rezultati za ukupan broj obrađenih slika. Testirat će se hipoteze postavljene u prvom dijelu rada.

Ključne riječi: *Algoritam, Matlab, matematička morfologija, vještačka inteligencija, neuronske mreže*

Uvod

Svrha istraživanja je da se istraži postupak prepoznavanja automobilskih tablica. Polazeći od same fotografije, ekstrakcije tablice, podjele tablice na segmente i na kraju čitanje karaktera tablice. Korišten je programski paket Matlab za digitalnu obradu slike i detektovanje tablice na fotografiji automobila. Nakon pronalaska najboljeg kandidata za tablicu izvršena je segmentacija iste. Finalni korak je bilo prepoznavanje znakova.

Osvrt na dosadašnje istraživanje

Dušica Lukić i Miloš Radulović su u radu [2] prikazali postupak lociranja i segmentacije registarskih tablica u Srbiji. Autori su sematski prikazali čitav proces dobijanja znakova sa registarske tablice. Nisu prikazali tačnost algoritma na većem broju uzoraka. Jelena Novosel u radu [4] prikazuje postupak prepoznavanja tipa vozila i njihovih registarskih tablica uzetih sa slika dobijenih od videonadzora na autocestama u Hrvatskoj. Segmentaciju znakova i njihovo prepoznavanje je opisano uz pomoć OCR. U svojim rezultatima pojašnjava moguće poteškoće i razloge pogrešnih rezultata. Zheng Hui Hu u svom radu [9] prikazuje proces i algoritam lociranja i segmentiranja automobilskih tablica u Kaliforniji. Za razliku od prethodnih autora

postupak segmentacije i prepoznavanja znakova se zasniva na neuronskim mrežama i vještačkoj inteligenciji.

Jednostavni algoritmi zahtijevaju samo operacije sa rubovima registarskih tablica [1], ali postoje i složeni algoritmi koji se zasnivaju na vektorskoj kvantizaciji [7], fazi grupisanju [6] i fazi logici [8]. Segmentacija tablice zavisi od države do države u kojoj se primjenjuje program. Svaka država ima zasebni format registarske tablice: veličinu, oblik, boju pozadine, boju znakova i broj znakova. Vidimo da ne postoji univerzalni algoritam za sve vrste tablica, nego je potrebno za svaku vrstu tablice prilagoditi algoritam.

Predmet istraživanja

Registarska tablica u Bosni i Hercegovini je sljedećeg izgleda:



Slika 1: Registarska tablica države Bosne i Hercegovine

Prema BIHAMK-ovim podacima za 2015. godinu u Bosni i Hercegovini registrovano je 952595 vozila. Zbog toga se vidi potreba za nekim načinom monitoringa vozila koje se nalaze u saobraćaju. Kako bi se mogli sankcionisati oni koji krše pravila i zakone o saobraćaju potrebno ih je na neki način identifikovati. Zbog toga je kreiran program za automatsko čitanje registarskih tablica automobila ili ANPR (Automatic number plate recognition).

Cilj istraživanja

U ovom radu pojasnit ćemo postupak prepoznavanja automobilskih tablica. Koristit će se programski paket Matlab za digitalnu obradu slike i detektovanje tablice na fotografiji automobila. Nakon pronalaska najboljeg kandidata za tablicu izvršit će se segmentacija iste. Finalni korak je prepoznavanje znakova. Cilj istraživanja je prikazati postupak prepoznavanja registarskih tablica automobila. Pokazati kolika je tačnost samog algoritma pri testiranju u različitim uslovima.

Hipoteze rada

Na osnovu definisanog predmeta istraživanja i ciljeva istraživanja hipoteze rada su:

H1: Kvalitet i postavke kamere sa kojom se uzima fotografa utiče na tačnost dobijenih podataka.

H2: Segmentacija i prepoznavanje znakova tablice uz upotrebu vještačke inteligencije daje bolje rezultate od metode prepoznavanja uzoraka.

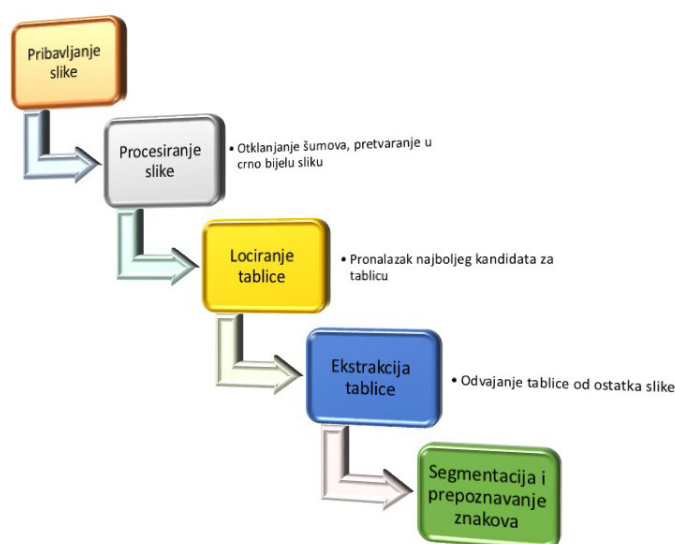
Očekivani rezultati istraživanja

Očekivani rezultati ovog istraživanja su:

1. Postupak prepoznavanja tablica registarskih tablica automobila.
2. Funkcionalan algoritam sa velikom tačnošću očitavanja znakova.
3. Potvrđivanje hipoteza ovog rada.

Proces prepoznavanja tablica

Proces automatskog prepoznavanja automobilskih tablica je veoma složen i sastavljen iz više koraka koji se nadovezuju. Da bi dobili željene rezultate, potrebno je proći od uzimanja slike na kojoj se nalazi automobil sa svim svojim elementima do broja registracije automobila.



Slika 2: Proces prepoznavanja registarskih tablica

Metode segmentacije slike

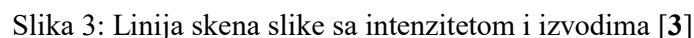
Posmatrat ćemo metode segmentacije slike koje su bazirane na prepoznavanju velikih, lokalnih promjena u intenzitetu. Moguće je posmatrati izolirane tačke, linije i ivice. Pikseli ivice su pikseli kod kojih se intenzitet iznenada mijenja, i ivice (ili rubni segmenti) su povezane setovima rubnih piksela. Detektori ivica su metode dizajnirane da otkriju rubne piksele. Linija može biti prikazana kao rubni segment u kojem je intenzitet pozadine na bilo kojoj strani linije ili puno veći ili puno manji nego intenzitet rubnih piksela. Izolirana tačka može biti posmatrana kao linija čija su dužina i širina jednaka širini i dužini jednog piksela. Iznenadne lokalne promjene intenziteta mogu biti prepoznate koristeći izvode. Prvi i drugi izvod su idealni za ovu svrhu. Izvodi digitalne funkcije su definisani preko diferentnog računa. Postoji mnogo načina za aproksimaciju ovih razlika, ali potrebno je da aproksimacija koja se koristi za prvi izvod

Aproksimaciju prvog izvoda u tački x jednodimenzionalne digitalne funkcije $f(x)$ dobijamo razvijanjem funkcije $f(x + 4x)$ u Tejlorov razvoj kada $4x = 1$ i zadržavanjem samo linearnih članova. Rezultat ove aproksimacije je prvi izvod funkcije $f(x)$

Drugi izvod dobijamo ponovnim diferenciranjem prvog izvoda

Ovaj razvoj je oko tačke $x + 1$. Nas interesuje drugi izvod oko tačke x , tako da ćemo oduzeti jedan od svih argumenata prethodne jednakosti i dobiti

Lahko se provjeri da prvi i drugi izvod funkcije $f(x)$ zadovoljavaju prethodno iskazane uslove.



258

u intenzitetu se odvija između različitih piksela. Broj intenziteta je ograničen sa osam radi pojednostavljenja.

Primijetimo da je prvi izvod različit od nule na početku i tokom cijele rampe intenziteta, dok je drugi izvod različit od nule samo na početku i kraju rampe intenziteta. Zbog toga što ivice digitalne slike čine ovu tranziciju, zaključujemo da prvi izvod prikazuje deblje ivice, a drugi izvod one mnogo sitnije. Ako posmatramo izoliranu tačku, vidimo da je jačina reakcije tačke mnogo veća na drugom izvodu nego na prvom. Ovo se dešava jer drugi izvod agresivnije pojačava oštrije promjene od prvog izvoda. Zbog toga je za očekivati da drugi izvod prikaže sitnije detalje mnogo bolje nego prvi izvod. Drugi izvod se koristi da se otkrije da li je ivica prijelaz sa svijetlog u tamno (drugi izvod je negativan), ili iz tamnog u svijetlo (drugi izvod je pozitivan). Za izračunavanje prvog i drugog izvoda u svakom pikselu jedne slike koriste se prostorni filteri.

ω_1	ω_2	ω_3
ω_4	ω_5	ω_6
ω_7	ω_8	ω_9

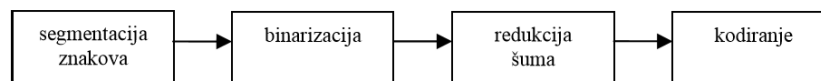
Slika 4: Maska prostornog filtera 3x3

U filteru 3x3 potrebno je izračunati sumu proizvoda koeficijenata maske, sa vrijednostima intenziteta u regionu obuhvaćenih maskom. Pomoću ove maske dobijamo vrijednost centralne tačke u obuhvaćenom regionu:

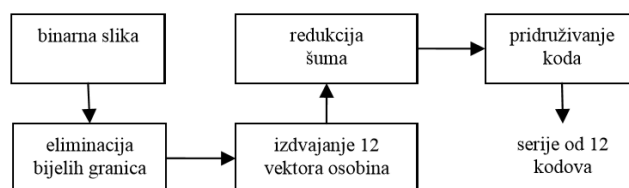
$$R = \omega_1 z_1 + \omega_2 z_2 + \dots + \omega_9 z_9 = \sum_{k=1}^9 \omega_k z_k.$$

Postupak prepoznavanja znakova tablice

Da bi mogli prepoznavati znakove tablice potrebno je da prvo izvršimo proces učenja. Na taj način stvaramo bazu kodiranih znakova tablice. Proces započinje sa segmentacijom svakog znaka zasebno i njegove binarizacije. Kada smo izvršili binarizaciju potrebno je da uklonimo šumove i takve znakove onda kodiramo.

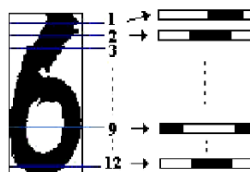


Slika 5: Proces kreiranja baze znakova [5]



Slika 6: Proces kodiranja znakova [5]

Proces kodiranja ulaznih veličina se može uraditi na dva načina. Jedan način se sastoji od izdvajanja 12 vektora osobina. Nakon pretvaranje slike u crno-bijelu (binarizacije) izdvaja se 12 karakterističnih linija od koja se svaka sastoji od bijelog dijela koji ima vrijednost 1 i crnog dijela koji ima vrijednost 0. Nakon što izdvojimo vektore osobina potrebno je još jedanput da se izvrši redukcija šuma. Kada smo završili ovaj postupak svakom od vektora pridružujemo odgovarajući kod.



Slika 7: Izdvajanje 12 linija slike znaka [5]

Na ovaj način dobijamo manje ulaznih neurona, i smanjujemo vrijeme učenja neuronske mreže, ali ostavljamo više mogućnosti za pogrešku. Drugi način je da posmatramo svaki piksel naše slike koja je dimenzije 24x42 piksela. Na taj način imamo 1008 ulaznih neurona i vrijeme našeg učenja se povećava, sam proces se usložnjava, ali daje bolje rezultate prilikom ispitivanja. Ovaj način će biti primijenjen u testiranju hipoteza ovog rada. Pošto su tablice u Bosni i Hercegovini specifične, sastoje se samo od slova "A, E, O, J, K, M, T" i cifri od 0 do 9, sam proces učenja je manji jer imamo manje znakova koje treba obraditi.

Glavni rezultati

Za potrebe ovog istraživanja kreirana je aplikacija u programskom paketu Matlab-u pomoću koje je izvršeno testiranje hipoteza. Prikazat ćemo na jednoj slici način rada algoritma i programa, od same segmentacije slike, lociranja tablice, ekstrakcije tablice i na kraju segmentacije i prepoznavanja znakova.

Proces pripremanja slike za obradu

Prvi korak u postupku prepoznavanja automobilske registracijske tablice je pripremanje slike za obradu. Potrebno je pretvoriti sliku u crno-bijelu, kod za to je:

```
I = imread('fiat.jpg');  
% Pretvaranje slike u crno bijelu  
Igray = rgb2gray(I);
```

Pretvaranje slike u crno-bijelu nam omogućava da primijenimo operacije matematičke morfologije kako bi mogli da odredimo položaj tablice. Nakon što smo promijenili sliku u crno-bijelo potrebno je ukloniti sve smetnje i šumove koji se nalaze na slici. Za to koristimo matematičku morfologiju i u ovom slučaju to je dilatacija i erozija. Već smo vidjeli da dilatacija i erozija pojačavaju granice, u ovom slučaju to su granice tablice koje treba locirati i ekstrahovati. Kod za pojačavanje granica je:

```
%% Dilatacija kako bi se uklonili sumovi  
Idilate = Igray;  
for i = 1:rows  
for j = 2:cols-1  
temp = max(Igray(i,j-1), Igray(i,j)); Idilate(i,j) = max(temp, Igray(i,j+1)); end  
end
```

Dilatacija nije ništa drugo nego primjena lokalnog maksimuma. Za ovako pripremljenu sliku potrebno obraditi ivice u smjerovima kako bi što lakše pronašli poziciju registracijske tablice. Da bi to uradili potrebno je pronaći maksimume u horizontalnom pravcu na histogramu i onda taj histogram pokušati izjednačiti koristeći filtere.

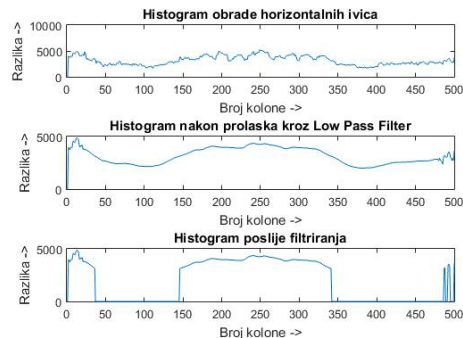
```
disp('Obrada ivica horizontalno...'); max_horz = 0; maximum = 0;  
for i = 2:cols  
sum = 0;  
for j = 2:rows  
if(I(j, i) > I(j-1, i)) difference = uint32(I(j, i) - I(j-1, i));  
else  
difference = uint32(I(j-1, i) - I(j, i)); end  
if(difference > 20) sum = sum + difference; end  
end  
horz1(i) = sum;  
% Pronalazak najveće vrijednosti  
if(sum > maximum) max_horz = i; maximum = sum; end  
total_sum = total_sum + sum; end  
average = total_sum / cols; figure(5);  
% Prikazivanje histograma za analizu  
subplot(3,1,1); plot (horz1);  
title('Histogram obrade horizontalnih ivica'); xlabel('Broj kolona ->'); ylabel('Razlika ->');  
%% Primjena Low Pass Filtera  
disp('Prolazak histograma kroz Low Pass Filter...'); sum = 0; horz = horz1;  
for i = 21:(cols-21) sum = 0; for j = (i-20):(i+20) sum = sum + horz1(j); end  
horz(i) = sum / 41; end  
subplot(3,1,2); plot (horz);  
title('Histogram poslije prolaska kroz Low Pass Filter'); xlabel('Broj kolona ->'); ylabel('Razlika ->');  
%% Filtriranje vrijednosti horizontalnog histograma primjenom dinamičkog praga  
disp('Filtriranje horizontalnog histograma...');  
for i = 1:cols  
if(horz(i) < average) horz(i) = 0;  
for j = 1:rows
```

```

I(j, i) = 0; end
end
end
subplot(3,1,3); plot (horz); title('Histogram poslije filtriranja'); xlabel('Broj kolona ->'); ylabel('Razlika ->');

```

Rezultat prethodnog koda je sljedeća slika:



Slika 8: Modifikacija horizontalnog histograma koristeći filtere

Možemo da vidimo kako se Low Pass Filter pokušava da "izgladi" histogram mijenjajući vrijednosti susjednih piksela, tako da nema više naglih promjena u vrijednostima histograma. Nakon korištenja dinamičkog praga za filtriranje, vidimo da sve vrijednosti koje su bile ispod praga otklanjaju se i dobijaju vrijednost 0.

Sada ćemo isto da uradimo i za vertikalne ivice. Koristi ćemo sljedeći kod:

```

difference = 0; total_sum = 0; difference = uint32(difference);
disp('Vertikalna obrada ivica...'); maximum = 0; max_vert = 0;
for i = 2:rows
    sum = 0;
    for j = 2:cols
        %cols
        if(I(i, j) > I(i, j-1)) difference = uint32(I(i, j) - I(i, j-1)); end
        if(I(i, j) <= I(i, j-1)) difference = uint32(I(i, j-1) - I(i, j)); end
        if(difference > 20) sum = sum + difference; end
    end
    vert1(i) = sum;
    %% Pronalazak maksimuma u vertikalnom histogramu
    if(sum > maximum) max_vert = i; maximum = sum; end
    total_sum = total_sum + sum; end
    average = total_sum / rows; figure(6); subplot(3,1,1); plot (vert1);
    title('Obrada histograma vertikalnih ivica'); xlabel('Broj reda ->'); ylabel('Razlika ->');
    %% Izjednačavanje vertikalnog histograma koristeći Low Pass Filter
    disp('Propustanje vertikalnog histograma kroz Low Pass Filter...');
    sum = 0; vert = vert1;
    for i = 21:(rows-21) sum = 0;
        for j = (i-20):(i+20) sum = sum + vert1(j); end
        vert(i) = sum / 41; end
    subplot(3,1,2); plot (vert);
    title('Histogram poslije prolaska kroz Low Pass Filter'); xlabel('Broj reda ->'); ylabel('Razlika ->');
    %% Filtriranje vertikalnog histograma koristeći dinamički prag
    disp('Filtriranje vertikalnog histograma...');
    for i = 1:rows
        if(vert(i) < average) vert(i) = 0;
    end
end
for j = 1:cols

```

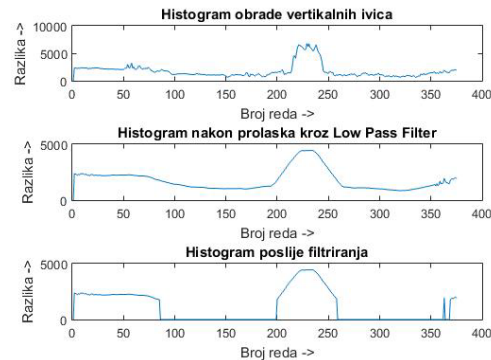


```

I(i, j) = 0; end
end
end
subplot(3,1,3); plot (vert); title('Histogram poslije filtriranja'); xlabel('Broj reda ->'); ylabel('Razlika ->');
figure(7), imshow(I);

```

Rezultat prethodnog koda je sljedeća slika:



Slika 9: Modifikacija vertikalnog histograma koristeći filtere

Lociranje tablice

Nakon što smo pripremili sliku, koristeći matematičku morfologiju i detekciju ivica, potrebno je da ispitamo gdje se nalaze mogući kandidati registarskih tablica automobila.

```

%% Pronalazak mogućih kandidata registarskih tablica
j = 1;
for i = 2:cols-2
    if(horz(i) ~= 0 && horz(i-1) == 0 && horz(i+1) == 0) column(j) = i; column(j+1) = i; j = j + 2;
    elseif((horz(i) ~= 0 && horz(i-1) == 0) || (horz(i) ~= 0 && horz(i+1) == 0)) column(j) = i; j = j+1; end
end
j = 1;
for i = 2:rows-2
    if(vert(i) ~= 0 && vert(i-1) == 0 && vert(i+1) == 0) row(j) = i; row(j+1) = i; j = j + 2;
    elseif((vert(i) ~= 0 && vert(i-1) == 0) || (vert(i) ~= 0 && vert(i+1) == 0)) row(j) = i; j = j+1; end
end
[temp column_size] = size (column); if(mod(column_size, 2)) column(column_size+1) = cols; end
[temp row_size] = size (row);
if(mod(row_size, 2)) row(row_size+1) = rows; end

```

Ovaj dio koda će ukloniti sve ono što sigurno nije registarska tablica automobila i ostavit će samo najpovoljnije kandidate za tablicu. U zavisnosti od modela automobila, kao i svjetla koje pada na automobil možemo da imamo puno mogućih kandidata.

Ekstrakcija registarske tablice

Nakon što smo dobili moguće kandidate za tablice potrebno je pronaći "pravu" tablicu. Da bi to uspjeli moramo otkloniti sve one koje ne odgovaraju izgledu same tablice.

```

% Provjera svakog kandidata pojedinačno
for i = 1:2:row_size
    for j = 1:2:column_size
        % Ako nije mogući kandidat, ukloni ga sa slike
    end
end

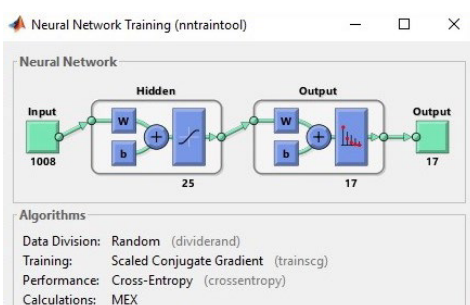
```


naše ulazne neurone. Izlazni neuroni će predstavljati nizove od 17 elemenata, gdje ćemo za svaku moguću vrijednost znaka tablice odrediti jedan red i kreirati izlazne neurone za našu neuronsku mrežu. Niz će se sastojati od nula osim na onom mjestu koji odgovara traženom znaku gdje će se nalaziti broj 1. Tako će na primjer znak 0 imati vrijednost

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1] ili znak 5 će imati vrijednost

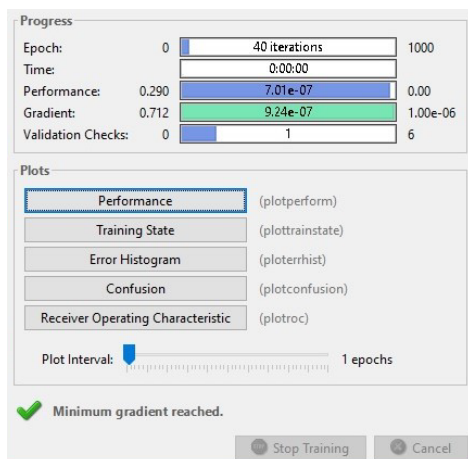
[0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]

Udaljenost jedinice od kraja našeg niza će predstavljati jedan od naših elemenata 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, E, J, K, M, O i T. Za treniranje mreže je korišteno 172 znaka, raspoređena na sljedeći način: 120 znakova za učenje, 26 znakova za validaciju i 26 znakova za testiranje. Za učenje je uzeto 25 skrivenih neurona i proces učenja neuronske mreže je sljedeći:



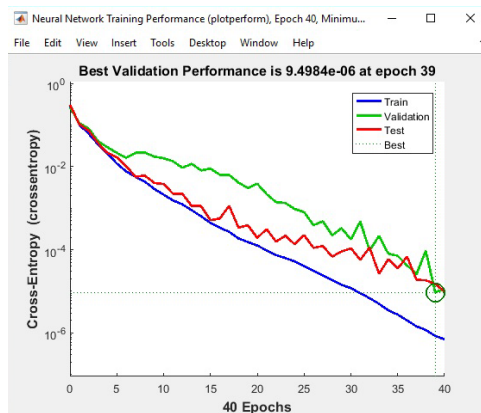
Slika 13: Postavka neuronske mreže

U ovom treniranju je korištena metoda Scaled Conjugate Gradient koja ima brže vrijeme treniranja, ali za jedan ovakav problem je sasvim uredi. Za mjerenje performansi je korištena metoda Cross-Entropy i za prora_cune MEX.



Slika 14: Proces učenja neuronske mreže

Treniranje je završeno u 40 epoha, sa performansom 7.01×10^{-7} i gradientom 9.24×10^{-7} .



Slika 15: Performanse treniranja neuronske mreže

Na prethodnom dijagramu vidimo performansu učenja neuronske mreže kroz epohe, i vidimo da je izlaz kada validacija i testiranje postignu istu performansu i to se realizuje u 39. epohi. Brzina treniranja mreže je veoma mala, manja od jedne sekunde. Naravno sa većim brojem elemenata za treniranje bi dobili veću tačnost, ali bi se i vrijeme učenja povećalo.

Prepoznavanje znakova

Da bi prepoznali znakove, potrebno ih je izdvojiti i svaki od njih provjeriti sa bazom da li odgovara naučenim znakovima. Ako se poklapa sa težinskim vrijednostima nekog od znaka iz baze, to je traženi znak. Izvršimo zamjenu crnih i bijelih piksela da dobijemo sliku:

A54 0 877

Slika 16: Zamjena crnih i bijelih piksela

Potrebno je izdvojiti svako slovo, da bi se moglo "poslati" na prepoznavanje. Algoritam za izdvajanje slova je:

```
fid = fopen('text.txt', 'wt');
load templates : global templates
num_letras=size(templates,2);
while 1
% Odvajanje linija sa slike (odbacivanje nepotrebnog)
[fl re]=lines(re); imgn=fl;
% Prebrojavanje i povezivanje komponenti
[L Ne] = bwlabel(imgn);
for n=2:Ne
```

```

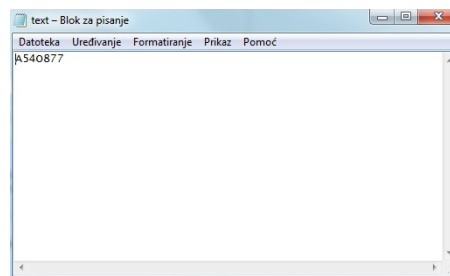
[r,c] = find(L==n);
% Izdvajanje slova
n1=imgn(min(r):max(r),min(c):max(c));
% Prilagodjavanje velicine slova
img_r=imresize(n1,[42 24]);
% Slanje slova u bazu na provjeru
letter=read_letter(img_r,num_letras);
% Snimanje slova
word=[word letter]; end
fprintf(fid,'%s\n',word); word=[ ]; if isempty(re) break
end
end
fclose(fid);
Rezultat su izdvojena slova:

```



Slika 17: Rezultat prebrojavanja i izdvajanja slova

Nakon što su slova izdvojena i poslana u bazu na provjeru upisuju se u text.txt datoteku. Kada se završi prepoznavanje svih znakova, otvara se text.txt datoteka sa upisanom registarskom tablicom automobila.



Slika 18: Završen postupak prepoznavanja znakova tablice

Rezultati testiranja

Da bi se ispitala prva hipoteza rada korištene su sljedeće vrste slika: 40 slika automobila uslikanih fotoaparatom Sony, 40 slika uslikanih mobilni aparatom Samsung i 20 slika koje su preuzete sa web-stranice www.olx.ba koje su uslikane nama nepoznatim vrstama aparata i njihovim osobinama. Korišteni su različiti uslovi osvjetljenja. Dobijeni rezultati su sljedeći:

Tablica 1: Rezultati lociranja tablice slika dobijenih različitim aparatima

Vrsta aparata	Broj testiranih slika	Broj slika sa lociranim tablicama	Broj pravilno izdvojenih znakova max: br. tablica x 7
Sony Cybershot	40	36 (90%)	242 (86,43%)
Samsung Galaxy S4	40	29 (72,5%)	189 (67,5%)
Nepoznato (www.olx.ba)	20	31 (77,5%)	210 (75%)

Program lociranja tablice je prvenstveno bio pravljen i uštimgavan prema slici dobijenoj sa fotoaparata Sony Cybershot. Vidimo da je upravo uspjeh lociranja tablice sa slike dobijene fotoaparatom Sony 90%, dok u oba ostala slučaja je ispod 80%.

Prema dobijenim rezultatima možemo konstantovati da je naša H1 Kvalitet i postavke kamere sa kojom se uzima fotografija utiče na tačnost dobijenih podataka potvrđena.

Za drugu hipotezu uzeto je 100 segmentiranih i izdvojenih znakova tablica koji nisu bili u procesu treniranja i učenja neuronske mreže i dobijeni su sljedeći rezultati:

Tablica 2: Rezultati poređenja neuronske mreže i OCR-a

Korištena metoda	Broj testiranih znakova	Broj uspješno prepoznatih znakova
Neuronska mreža	100	95 (95%)
OCR		72 (72%)

U ovoj tabeli možemo vidjeti da vještačka inteligencija daje znatno bolje rezultate od metode prepoznavanja uzorcima. Tačnost prepoznavanja uz pomoć neuronskih mreža je čak 95%, dok je kod prepoznavanja uzorcima 72%.

Zbog ovakvih rezultata testiranja možemo reći da je naša H2 Segmentacija i prepoznavanje znakova tablice uz upotrebu vještačke inteligencije daje bolje rezultate od metode prepoznavanja uzoraka, također potvrđena.

Zaključak

U ovom radu je prikazan proces prepoznavanja registarskih tablica automobila. Pojašnjena matematička morfologija i operacije matematičke morfologije potrebne da bi se slika što bolje pripremila za dalju obradu. Nakon toga primijenjene su metode za detekciju ivica kako bi se locirala tablica i ista izdvojila od ostatka slike. Vidjeli smo da bolja kvaliteta fotografije daje bolje rezultate lociranih registarskih tablica automobila. Razlog tome je da je prvobitni program kreiran da radi na osnovu kvalitetnije fotografije. Kvalitetnija fotografija ima veću rezoluciju i samim tim veći broj piksela za rad. Ako imamo veći broj piksela za rad moguće je bolje razgraničiti pojedine uslove.

Potrebno je razmisliti šta treba uraditi da bi se povećala tačnost lociranja tablice i šta ako tablica nije postavljena horizontalno. Ako je malo zakrivljena, prvo bi trebalo da se izvrši rotiranje i ispravljanje dobijene tablice. Kako bi se realizovalo prepoznavanje sa slika uzetih iz videa koji snima u stvarnom vremenu. Mnogo je pitanja koja bi se mogla ispitivati i testirati na ovu temu.

Literatura

- [1] Bai H., Liu C. (2004). "*A Hybrid License Plate Extraction Method Based on Edge Statistics and Morphology*", Proceedings of the 17th International Conference on Pattern Recognition.
- [2] Dušica L., Miloš R. (2009). "*Lokalizacija i segmentacija registarskih tablica*", 17. Telekomunikacioni forum TELFOR 2009 Srbija, Beograd, novembar 24.-26., 2009
- [3] Gonzales R., Woods R. (2007). "*Digital Image Processing 3rd edition*", Prentice Hall, New York.
- [4] Jelena N. (2011). "*Sustav računalnog vida za automatsko prepoznavanje vozila u svrhu nadzora prometa*", Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Zagreb, 2011.
- [5] Kraupner K. (2003). "*Uporaba višeslojnog perceptrona za raspoznavanje bročanoslovčanih znakova na registarskim tablicama*" - Diplomski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva.
- [6] Mi-Ae K., Young-Mo K. (2003). "*License Plate Surveillance System Using Weighted Template Matching*", Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop.
- [7] Rodolfo Z., Stefano R. (2000). "*Vector Quantization for License Plate Location and Image coding*", IEEE transactions on Industrial Electronics, Vol 47, No. 1.
- [8] Shyang-Lih C., Li-Shien C., Yun-Chung C., Sei-Wan C. (2004). "*Automatic License Plate Recognition*", IEEE transactions on Intelligent Transportation Systems, Vol 5, No. 1.
- [9] Zheng Hui H. (2007). "*An automatic license plate recognition system using image processing and neural network*", The Faculty of the Department of Computer Science, California State University, Los Angeles.

THE FUNCTION OF ARTIFICIAL INTELLIGENCE IN RECOGNIZING SIGNS AND OBJECTS

Abstract

In this paper I will explain the process of character recognition from car licence plates. The first part of paper is about theoretical aspects of algorithm for getting a character from car licence plates. The whole process shall be explained, including obtaining the picture, preparing the picture for processing, locating licence plate spot and finally character recognition of car licence plate. The second part of this paper is about practical usage of this algorithm. The steps of an algorithm in an example of an image and summary results for the total number of processed images will be displayed. The hypothesis set up in the first part of paper shall be tested.

Key words: Algorithm, Matlab, mathematical morphology, artificial intelligence, neural networks.