

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 812

**PRIMJENA METODE PCA NAD SKUPOM  
SLIKA ZNAKOVA**

Ivana Sučić

Zagreb, srpanj 2009



## Sadržaj

1.	Uvod.....	1
2.	Normalizacija slika znakova .....	2
2.1.	Pretvaranje slike u boji u sivu sliku .....	2
2.2.	Skaliranje slika znakova na zadanu veličinu .....	3
2.3.	Izjednačavanje histograma .....	4
2.4.	Uklanjanje pozadine .....	6
3.	Primjena metode PCA nad skupom znakova .....	8
3.1.	Matematička podloga.....	8
3.1.1.	Statistički postupci.....	8
3.1.2.	Svojstveni vektori i svojstvene vrijednosti .....	10
3.1.3.	Klasifikacija algoritmom K-najbližih susjeda (K-NN).....	11
3.2.	Metoda PCA .....	12
3.3.	Analiza svojstvenih komponenti na skupu znakova .....	15
4.	Eksperimentalni rezultati .....	21
5.	Programska izvedba.....	36
5.1.	Upute za korištenje .....	36
6.	Zaključak .....	37
7.	Literatura .....	38

## Popis tablica

Tablica 4-1 Numerički rezultati prepoznavanja po klasama znakova .....	21
Tablica 4-2 Točnost raspoznavanja u ovisnosti o odabranom broju svojstvenih vektora za skup za učenje od 474 slike.....	23
Tablica 4-3 Točnost raspoznavanja u ovisnosti o veličini skupa za učenje te broju klasa.....	23
Tablica 4-4 Rezultati metode svojstvenih znakova za slike normalizirane na veličinu 35x35 .....	24
Tablica 4-5 Rezultati metode svojstvenih znakova za slike normalizirane na veličinu 140x140 .....	24
Tablica 4-6 Slike iz skupa za testiranje i njihove rekonstrukcije s različitim brojem svojstvenih vektora.....	25
Tablica 4-7 Ulazna i prepoznata slika iz baze te pripadni grafovi.....	26
Tablica 4-8 Testni primjeri slika lošije kvalitete.....	27
Tablica 4-9 Rezultati nepoznatog znaka i slike koja nije znak.....	29
Tablica 4-10 Disperzija unutar klasa .....	30

## Popis slika

<i>Slika 2-1 Postupak normalizacije slike znaka za raspoznavanje.....</i>	<i>2</i>
<i>Slika 2-2 Zasivljene slike.....</i>	<i>3</i>
<i>Slika 2-3 Skalirane slike.....</i>	<i>4</i>
<i>Slika 2-4 Slike smanjenog osvjetljenja.....</i>	<i>6</i>
<i>Slika 2-5 Maska.....</i>	<i>7</i>
<i>Slika 2-6 Normalizirane slike znakova spremne za raspoznavanje.....</i>	<i>7</i>
<i>Slika 3-1 Primjer 2D vektora.....</i>	<i>12</i>
<i>Slika 3-2 Projekcija.....</i>	<i>13</i>
<i>Slika 3-3 Bolji prikaz podatka.....</i>	<i>13</i>
<i>Slika 3-4 Dobar prikaz podatka.....</i>	<i>13</i>
<i>Slika 3-5 Loš prikaz podatka.....</i>	<i>14</i>
<i>Slika 3-6 Kovarijanca varijabli.....</i>	<i>15</i>
<i>Slika 3-7 Prosječna slika.....</i>	<i>16</i>
<i>Slika 3-8 Svojtveni znakovi.....</i>	<i>19</i>
<i>Slika 4-1 Nepravilna slika znaka.....</i>	<i>25</i>
<i>Slika 4-2 Distribucija težinskih faktora pojedinih klasa.....</i>	<i>35</i>

# 1. Uvod

Jedna od ključnih odrednica cestovnog prometa je sigurnost u prometu. U svakoj je ljudskoj aktivnosti prisutan rizik, no daleko najveći je upravo u cestovnom prometu, u kojem sudjeluje gotovo svaki čovjek.

Kada bismo zamislili automobil koji ima sposobnost vožnje pod upravljanjem računala, neizmjerena prednost bi bila sama činjenica da ne zahtjeva osobu u ulozi vozača. Sigurnost u prometu ne bi ovisila o vozačevim sposobnostima kao ni o njegovom psihofizičkom stanju. Takvo vozilo imalo bi mogućnost praćenja situacije u prometu. Pratilo bi situaciju snimanjem događaja ispred sebe te donošenjem odluka o daljnjem tijeku upravljanja na temelju informacija koje bi dobivalo ugrađeno računalo. Jedna od najvažnijih informacija koja bi se dobivala iz slijeda slika snimanjem situacije ispred sebe su informacije koje pružaju prometni znakovi. Stoga, program koji bi pokretao vozilo mora uključivati algoritam pronalaženja i raspoznavanja prometnih znakova duž ceste.

Navedena dva pojma pronalaženje i raspoznavanje, mogu imati u nekim slučajevima isto značenje no bitno je da u ovom kontekstu imaju različita značenja. Pronalaženje znakova je postupak koji odgovara na pitanje: Gdje su objekti (znakovi) i da li ih ima? Raspoznavanje znakova je postupak kojim se raspoznaju znakovi jedan od drugoga. Dok se kod prepoznavanja samo traži gdje se znak nalazi, raspoznavanje pod pretpostavkom da je poznato mjesto znaka otkriva koji je to znak, otkriva tip znaka (npr. „obavezan smjer“).

U ovom radu bit će razrađen jedan od postupaka za raspoznavanje prometnih znakova u slijedu slika pribavljenih iz vozila u pokretu, te njegova programska implementacija. Rad se fokusira na raspoznavanju znakova primjenom analize svojstvenih komponenti (eng. Principal Component Analysis, PCA)

Postupak je implementiran u programskom jeziku Matlab, koji se često koristi za izvedbu prototipa matematički zahtjevnih programa. Svako poglavlje, popraćeno je objašnjenjem kako je pojedini dio postupka implementiran u Matlabu.

Rad je strukturiran na slijedeći način. Normalizacija slika, kao pomoćna metoda, objašnjena je u poglavlju 2. Detaljno objašnjenje metode PCA dano je u poglavlju 3. U poglavlju 4 izneseni su eksperimentalni rezultati, a u poglavlju 5 programska izvedba. Osvrt na čitav rad dan je zaključkom u poglavlju 6.

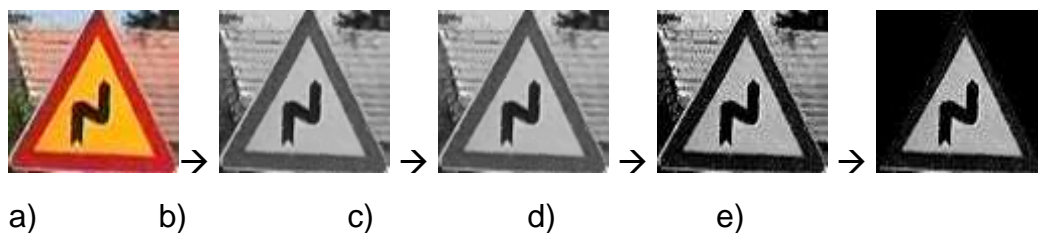
## 2. Normalizacija slika znakova

Prije nego što se počne s raspoznavanjem znakova, slike je potrebno normalizirati. Normalizacija se može shvatiti kao postupak kojim ujednačavamo one parametre na svim slikama znakova koji nam ništa ne govore o samom znaku, a njihova različitost može smetati u fazi raspoznavanja znakova iz tih slika. Takvi parametri su, na primjer, veličina znaka na slici (kao posljedica veličine slike ili udaljenosti znaka od kamere), prisutnost pozadine, uvjeti osvjetljenja itd. Očito je da ako se želi efikasno uspoređivati slike znakova ovakvi parametri na svim slikama trebaju biti isti (ili vrlo malo varirati na slikama), budući da njihove varijacije mogu uvelike utjecati na rezultate raspoznavanja.

Algoritam koristi normalizaciju slika znakova u slijedeća četiri koraka:

- 1) pretvaranje slike u boji u sivu sliku
- 2) skaliranje slika na zadanu veličinu
- 3) izjednačavanje histograma slike
- 4) uklanjanje pozadine

Primjer kompletnog postupka normalizacije prikazan je na slici 2-1.



*Slika 2-1 Postupak normalizacije slike znaka za raspoznavanje: a) originalna slika, b) zasivljena slika, c) skalirana slika na zadanu veličinu, d) slika izjednačenog histograma, e) kvadratna slika s otklonjenom pozadinom*

### 2.1. Pretvaranje slike u boji u sivu sliku

Slike znakova koje će se raspoznavati su predstavljene elementima modela RGB [6]. Model boja RGB temelji se na trikromatskom svojstvu ljudskog oka. Pojedine boje se dobivaju kombinirajući crvenu, zelenu i plavu boju u različitim

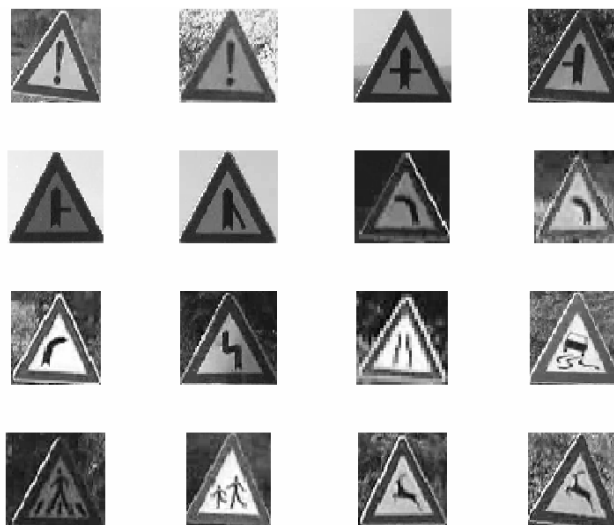
omjerima. Komponente boje obično se izražavaju u intervalu od 0 do 1, odnosno u slikama s jednim bajtom po komponenti u intervalu od 0 do 255.

U ovom radu razmatramo raspoznavanje sivih slika. Siva slika za svaki slikovni element ima samo jednu vrijednost i to vrijednost intenziteta. Vrijednosti se nalaze u rasponu od 0 do 255, s time da je 0 crna boja, a 255 bijela. Intenzitet se iz RGB vrijednosti može dobiti sljedećim izrazom:

$$I = \frac{1}{3}(R + G + B) \quad (2.1)$$

U Matlabu se ova pretvorba izvršava naredbom `J=rgb2gray(I)` gdje je `I` slika učitana naredbom `I=imread('neka_slika.bmp')`, a `J` varijabla u koju će se spremiti siva slika.

Primjeri zasivljenih slika dani su slikom 2-2.



*Slika 2-2 Zasivljene slike*

## 2.2. Skaliranje slika znakova na zadanu veličinu

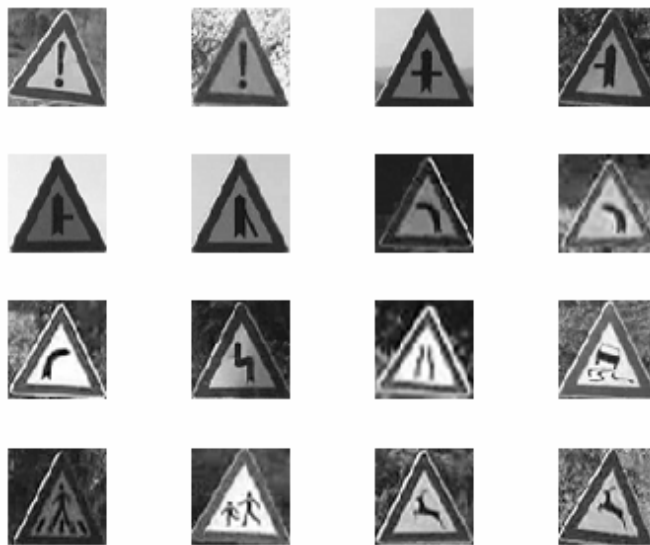
Originalne slike znakova različitih su veličina. Uzrok tomu može biti udaljenost znaka od kamere kao i sama veličina slike iz koje smo izrezali znak. Za metodu PCA nužno je da su sve slike jednake veličine. Podrazumijevana zajednička



veličina normaliziranih slika postavljena je na  $70 \times 70$  slikovnih elemenata. Te dimenzije odgovaraju srednjoj veličini svih slika znakova.

U Matlabu se skaliranje slika izvodi naredbom `B=imresize(A,[mrows ncols])`. U varijablu `A` je učitana originalna slika, koja se skalira na odabrane dimenzije `mrows` $\times$ `ncol` (`mrows` je broj redaka, a `ncol` broj stupaca matrice) te se tako dobivena slika sprema u varijablu `B`. Više o normalizaciji veličine slike može se naći u [3].

Primjeri izlaza iz ovog koraka normalizacije vide se na slici 2-3.



*Slika 2-3 Skalirane slike*

## 2.3. Izjednačavanje histograma

Slika se može zamisliti kao matrica čiji elementi predstavljaju svjetlinu slike u točki. Grafički prikaz distribucije svjetline slike naziva se histogram slike. Histogram prikazuje broj slikovnih elemenata za svaku vrijednost svjetline. Budući da se u ovom radu analiziraju sive slike, svjetlina svakog slikovnog elementa nalazi se u intervalu  $[0,255]$ .

Cilj ovog koraka normalizacije jest da sve slike u bazi imaju što sličnije histograme, odnosno ujednačavanje uvjeta osvjetljenja i kontrasta.

Ideja je da se promjenom srednje vrijednosti svjetline slike mijenja osvjetljenje slike, a promjenom standardne devijacije svjetline namješta kontrast. Prije početka normalizacije korisnik zadaje željenu srednju vrijednost svjetline slike kao i željenu

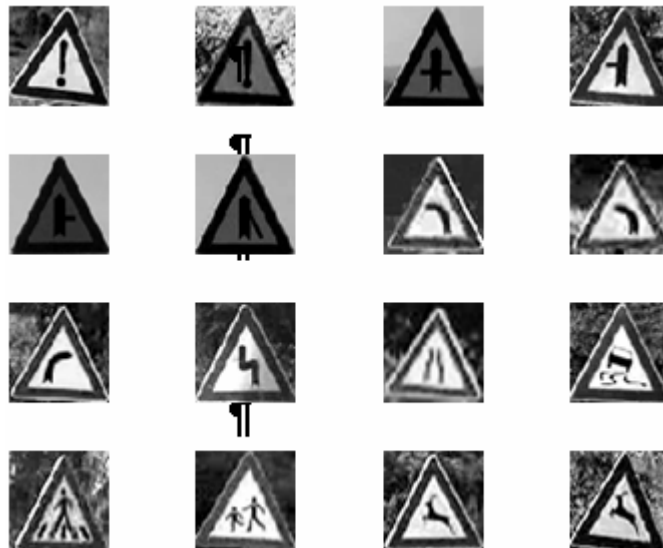
standardnu devijaciju svjetline. Zatim se za svaku ulaznu sliku nalazi njezina standardna devijacija svjetline te srednja vrijednost svjetline svih slikovnih elemenata. Slikovni elementi ulazne slike oduzimaju se od srednje vrijednosti svjetline svih slikovnih elemenata, a dobivena razlika množi se s kvocijentom željene standardne devijacije i standardne devijacije ulazne slike. Tom umnošku se dodaje odabrana srednja vrijednost svjetline. U Matlabu je taj postupak ostvaren slijedećim odsječkom koda.

```
%Izjednačavanje histograma
for i=1:size(S,2)
    %učitava cijeli stupac
    temp=double(S(:,i));
    %srednja vrijednost
    m=mean(temp);
    %standardna devijacija
    st=std(temp);
    S(:,i)=(temp-m)*ustd/st+um;
end
```

U gore navedenom odsječku koda, varijabla *S* je matrica koja je organizirana tako da svaki stupac matrice predstavlja ulaznu sliku. Petljom prolazimo kroz pojedinačni stupac matrice *S* koji spremamo u pomoćnu varijablu *temp*. U varijabli *m* nalazi se izračunata srednja vrijednost stupca pomoću funkcije *mean*, a u varijabli *st* standardna devijacija tog istog stupca odnosno slike. Varijabla *ustd* je željena standardna devijacija definirana od strane korisnika, *um* odabrana srednja vrijednost svjetline.

Što je *um* bliži 255 to je slika svjetlija. Povećanjem *ustd* veća je razlika između svjetline svakog slikovnog elemenata tj. veći je kontrast.

Primjeri izlaza iz ovog koraka normalizacije vide se na slici 2-4.



Slika 2-4 Slike smanjenog osvjetljenja

## 2.4. Uklanjanje pozadine

U ovom radu razmatramo trokutne znakove. Slike za raspoznavanje su kvadratnog oblika stoga sadrže dijelove pozadine. U ovom koraku normalizacije pokušava se otkloniti utjecaj pozadine na postupak raspoznavanja znaka.

Otklanjanje pozadine vrši se logičkom operacijom 'I' s obzirom na zadanu masku. Logička nula u masci zacrnjuje željeni dio slike, dok logička jedinica ne mijenja vrijednost preostalih elemenata.

Maska je slika, jednake veličine kao i normalizirane slike baze ( $70 \times 70$  slikovnih elemenata), na kojoj se nalazi bijeli trokut na crnoj podlozi. Maska prikazana na slici 2-5. Bijeli dio (trokut) predstavlja logičke jedinice, dok ostatak logičke nule. Množenjem svakog slikovnog elementa ulazne slike sa svakim slikovnim elementom maske postavljamo vrijednost svih slikovnih elemenata koji padaju izvan trokuta na vrijednost nula tj. zacrnjujemo pozadinu.

U Matlabu maska je stvorena naredbom  $BW = \text{roipoly}(I, c, r)$ . U varijablu  $I$  učitana je slika čije će dimenzije imati  $i$  slika maske spremljena u varijablu  $BW$ . Varijable  $c$  i  $r$  predstavljaju niz koordinata vrhova željenog poligona, u ovom slučaju trokuta. U  $c$  se stavljaju stupci, a u  $r$  redci vrhova poligona. Aritmetičkim operatorom  $.*$  ostvarujemo matrično množenje svakog elementa sa svakim elementom.

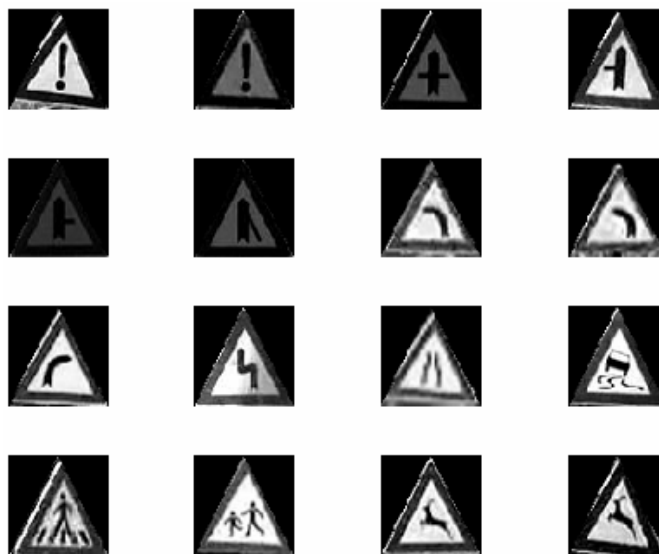
Odsječak koda za uklanjanje pozadine dan je u nastavku.

```
%kreiranje maske za uklanjanje pozadine
temp=ones(irow,icol);
c=[1 35 70];
r=[70 1 70];
%maska ima 1-ice u trokutu, 0-e van trokuta
maska=roipoly(temp,c,r);
%primjena maske na sve slike
for i=1:G
    img=reshape(S(:,i),icol,irow);
    img=img';
    %množenjem slike s maskom uklanja se pozadina
    img=img.*maska
    %stvara (N1*N2)x1 matricu, img'-transponirano
    temp=reshape(img',irow*icol,1);
    %S je N1*N2xM matrica, svaki stupac je jedna slika
    S(:,i)=temp;
end
```

Konačno normalizirani primjeri slika dani su slikom 2-6.



Slika 2-5 Maska



Slika 2-6 Normalizirane slike znakova spremne za raspoznavanje

### 3. Primjena metode PCA nad skupom znakova

Analiza svojstvenih komponenti (eng. Principal Component Analysis , PCA) koja se u literaturi još naziva i Karhunen-Loèveom ekspanzijom te Karhunen-Loèveom transformacijom, vrlo je moćna metoda za analizu podataka [2]. Posebno je bitna mogućnost izlučivanja najvažnijih značajki iz skupa uzoraka. Stoga je metoda našla veliku primjenu i u području raspoznavanja uzoraka.

Prije no što se krene s detaljnijim objašnjavanjem same ideje postupka PCA, naveden je kratak osvrt na matematičku podlogu nužnu za razumijevanje navedene metode.

#### 3.1. Matematička podloga

##### 3.1.1. Statistički postupci

Statistika je grana primijenjene matematike koja se bavi analizom podataka [4]. Predstavlja skup ideja i metoda koje se upotrebljavaju za prikupljanje i interpretaciju podataka u nekom području istraživanja, te za izvođenje zaključaka u situacijama gdje su prisutne nesigurnosti i varijacije. U nastavku slijedi objašnjenje statističkih metoda neophodnih za razumijevanje postupka analize svojstvenih komponenti.

Aritmetička sredina je najpoznatija srednja vrijednost. U statističkoj analizi se računa za neki skup brojeva kao kvocijent zbroja članova i broja članova skupa.

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_{n-1} + x_n}{n} \quad (3.1)$$

Poznavanje srednje vrijednosti pruža mogućnost da se procijeni nepoznato očekivanje  $a$  skupa brojeva  $x$ .

Ako nam je očekivanje skupa brojeva poznato, a disperzija nije, računamo varijancu. Varijanca je aritmetička sredina kvadrata odstupanja vrijednosti numeričkog obilježja od njihove aritmetičke sredine.

$$\sigma^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n} \quad (3.2)$$

Standardna devijacija predstavlja srednje ili prosječno kvadratno odstupanje vrijednosti numeričkog obilježja od aritmetičke sredine. Računa se po formuli:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}} \quad (3.3)$$

Kovarijanca pokazuje koliko se dvije varijable mijenjaju zajedno. To je različito od varijance, koja pokazuje koliko se jedna varijabla mijenja. Kovarijanca postaje više pozitivnom za svaki par vrijednosti koji se razlikuje od njihovih srednjih vrijednosti u istom smjeru, te postaje više negativna za svaki par vrijednosti koji se razlikuje od njihovih srednjih vrijednosti u suprotnim smjerovima. Ako postoje tri skupa vrijednosti  $(x_1, x_2, \dots, x_n)$ ,  $(y_1, y_2, \dots, y_n)$  i  $(z_1, z_2, \dots, z_n)$  tada su njihove kovarijance :

$$\sigma_{xy}^2 = \sigma_{yx}^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n} \quad (3.4)$$

$$\sigma_{xz}^2 = \sigma_{zx}^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})(z_i - \bar{z})}{n} \quad (3.4)$$

$$\sigma_{yz}^2 = \sigma_{zy}^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})(z_i - \bar{z})}{n} \quad (3.4)$$

Ili u matičnom obliku:

$$C = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{xz}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{yz}^2 \\ \sigma_{zx}^2 & \sigma_{zy}^2 & \sigma_{zz}^2 \end{bmatrix} \quad (3.5)$$

Važno je uočiti da vrijedi  $\sigma_{xy}^2 = \sigma_{yx}^2$ ,  $\sigma_{xz}^2 = \sigma_{zx}^2$ ,  $\sigma_{yz}^2 = \sigma_{zy}^2$ . Općenito ako postoji  $m$  skupova od  $n$  mjerenja kovarijanca se može napisati kao matrica  $m \times n$  (matrica kovarijanca je simetrična):

$$C = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1n}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2n}^2 \\ \dots & \dots & \dots & \dots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \dots & \sigma_{mn}^2 \end{bmatrix} \quad (3.6)$$

### 3.1.2. Svojstveni vektori i svojstvene vrijednosti

Vektor  $v \neq 0$  naziva se svojstvenim vektorom [1] matrice  $A$  ako postoji skalar  $\lambda$  tako da vrijedi

$$A \cdot v = \lambda \cdot v \quad (3.7)$$

Skalar  $\lambda$  tada se naziva svojstvena vrijednost matrice  $A$ , koja odgovara svojstvenom vektoru  $v$ . Gornja jednadžba ekvivalentna je s

$$(\lambda I - A)v = 0 \quad (3.8)$$

gdje je  $I$  jedinična matrica. Ako je  $A$  realna i simetrična matrica, svojstvene vrijednosti i vektori su međusobno različiti.

Gornja jednadžba također nam može poslužiti u nalaženju svojstvenih vrijednosti i svojstvenih vektora matrice. Da bi jednadžba imala netrivialno rješenje mora vrijediti da je determinanta matrice  $\lambda I - A$  jednaka nuli, odnosno

$$|\lambda I - A| = \begin{vmatrix} \lambda - a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & \lambda - a_{22} & \dots & -a_{2n} \\ \dots & \dots & \dots & \dots \\ -a_{n1} & -a_{n2} & \dots & \lambda - a_{nn} \end{vmatrix} = 0 \quad (3.9)$$

Ova je determinanta polinom stupnja  $n$  po nepoznanici  $\lambda$ . Navedeni polinom nazivamo karakteristični polinom matrice  $A$  i označavamo s  $\kappa(\lambda)$ ,

$$\kappa(\lambda) = \det(\lambda I - A) \quad (3.10)$$

On ima oblik

$$\kappa(\lambda) = \lambda^n - d_1 \lambda^{n-1} - \dots - d_{n-1} \lambda - d_n \quad (3.11)$$

Jednadžba

$$\kappa(\lambda) = \det(\lambda I - A) = 0 \quad (3.12)$$

naziva se karakteristična jednadžba matrice  $A$ . Njena rješenja su svojstvene vrijednosti te matrice.

Općenito za sve matrice vrijedi da su svojstveni vektori koji odgovaraju različitim svojstvenim vrijednostima međusobno linearno nezavisni, te vrijedi da ako su sve nul-točke karakterističnog polinoma različite postoji baza prostora koju čine svojstveni vektori promatrane matrice.

### 3.1.3. Klasifikacija algoritmom K-najbližih susjeda (K-NN)

K-najbližih susjeda (K-NN) je klasifikacijska metoda kojom se klasificiraju novi primjeri u jednu od definiranih kategorija na temelju predočenih klasificiranih primjera iz skupa za učenje. Algoritam podrazumijeva mogućnost prikazivanja svih primjera u  $n$ -dimenzionalnom prostoru gdje se udaljenost između dva primjera definira kao Euklidska udaljenost.

Primjer  $x$  je opisan vektorom značajki  $[a_1(x), a_2(x), \dots, a_n(x)]$ , gdje  $a_k(x)$  označava  $k$ -ti atribut primjera  $x$ . Euklidska udaljenost između dva vektora  $x_i$  i  $x_j$  je

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n [a_r(x_i) - a_r(x_j)]^2} \quad (3.13)$$



Prilikom klasifikacije definira se broj  $K$ , koji označava koliko se najbližih susjeda traženom primjeru pronalazi. Najbliži susjed je onaj koji ima najmanju udaljenost do neklasificiranog uzorka. Primjer koji se klasificira dodijeliti će se onoj kategoriji kojoj pripada najviše od  $K$  najbližih susjeda.

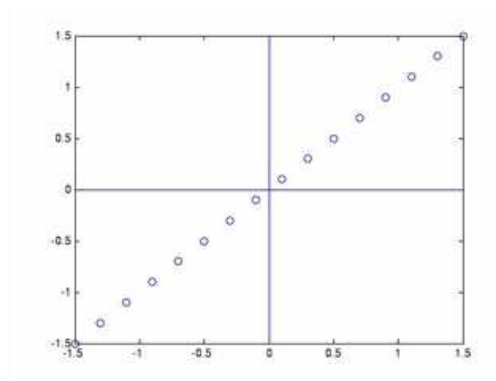
## 3.2. Metoda PCA

PCA (Principal Component Analysis) ili analiza svojstvenih komponenti je statistički postupak za reduciranje dimenzije podataka. Pri reduciranju se želi izgubiti što je manje moguće informacija i ovaj postupak upravo to omogućava. Metoda je našla veliku primjenu i u području raspoznavanja uzoraka.

Primjena metode PCA nad skupom uzoraka omogućava dobivanje potprostora svih mogućih slika koje sadrže znakove. Ideja je da se pronađe takav potprostor zadane niže dimenzionalnosti koji najbolje prikazuje pojedini razred uzoraka u smislu da pogreška između originalnog uzorka i projekcije uzorka u taj potprostor bude minimalna. U nastavku slijedi detaljnije objašnjenje navedene ideje.

Dakle, svaku sliku znaka možemo zamisliti kao vektor slikovnih elemenata (piksela).

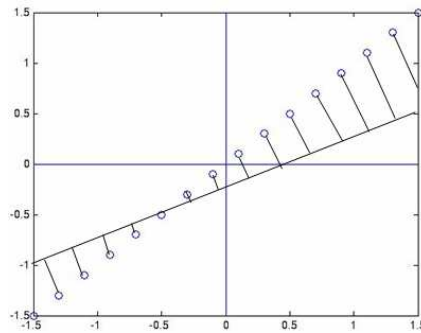
Primjer 2D vektora dan je slikom 3-1.



Slika 3-1 Primjer 2D vektora

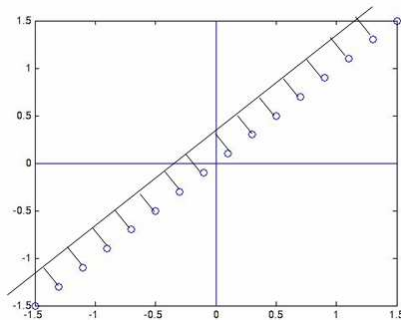
Pretpostavimo da podatke sa slike 3-1 želimo prikazati u 1D prostoru. Da bismo to postigli odaberemo pravac u prostoru na kojeg ortogonalno projiciramo svaku točku.

Slika 3-2. prikazuje projekciju.

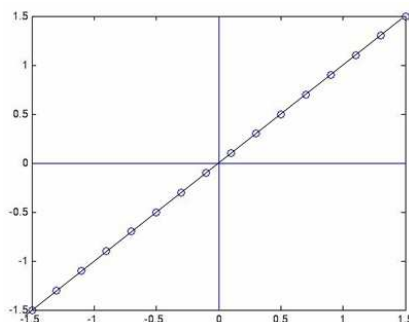


*Slika 3-2 Projekcija*

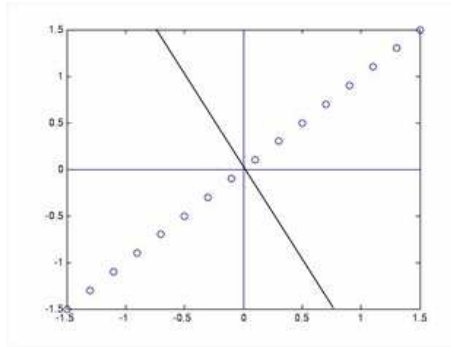
Neki pravci će bolje prikazivati podatke, neki lošije. Pravci koji bolje prikazuju podatke su oni koji pri ortogonalnoj projekciji točkaka na pravac daju malo odstupanje originalnih podatka od projiciranih (slika 3-3), a dobar pravac je onaj gdje je to odstupanje minimalno (slika 3-4.). Loš prikaz podataka predstavljaju pravci za koje je odstupanje originalnih od projiciranih podataka toliko da nereprezentativno prikazuju podatak, gube se bitne informacije (slika 3-5).



*Slika 3-3 Bolji prikaz podatka*



*Slika 3-4 Dobar prikaz podatka*

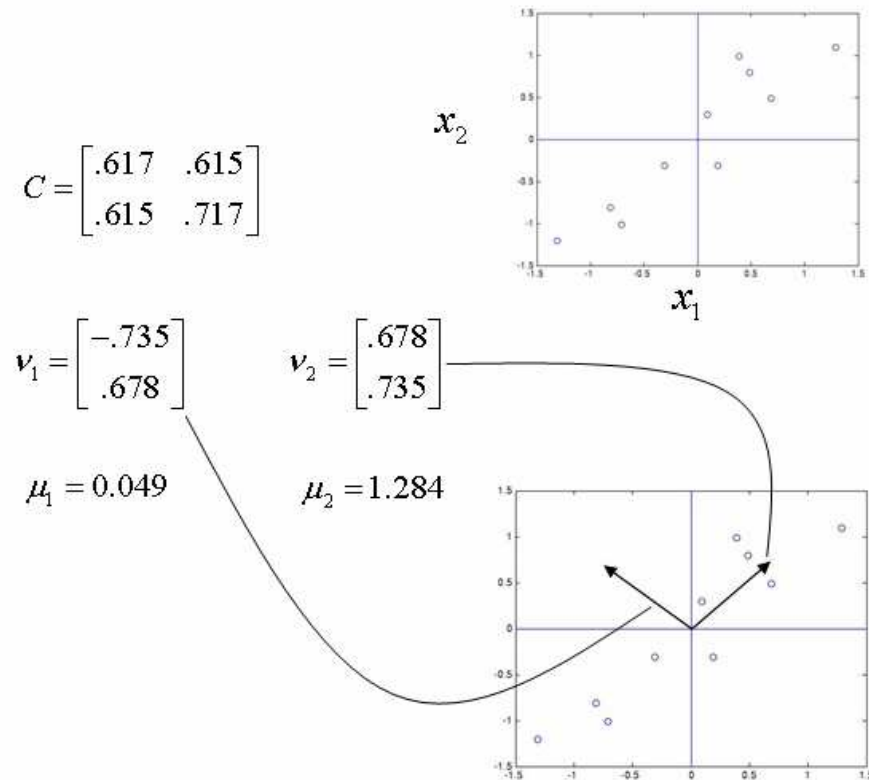


*Slika 3-5 Loš prikaz podatka*

Ostaje pitanje kako se pokazati da traženi potprostor mora prolaziti kroz težište ulaznih vektora. Tu dolazimo do kovarijacijske matrice koja je pojašnjena u poglavlju 3.1.1. Ona prikazuje kako se svake dvije varijable iz više skupina varijabli, mijenjaju zajedno. Svojstveni vektori simetrične kovarijacijske matrice s većim svojstvenim vrijednostima odgovaraju smjerovima u kojima podaci više variraju.

Pronalaženje najvećih svojstvenih vrijednosti i svojstvenih vektora kovarijacijske matrice predstavlja glavnu ideju PCA ili analize svojstvenih komponenti.

Na slici 3-6. možemo vidjeti dvije varijable  $x_1$  i  $x_2$  te njihovu kovarijacijsku matricu  $C$ . Svojstvene vrijednosti te matrice dane su vektorima  $v_1$  i  $v_2$ , kao i njihove svojstvene vrijednosti  $\mu_1$  i  $\mu_2$ . Vidimo da vektor  $v_2$  s većom svojstvenom vrijednošću  $\mu_2$  prikazuje smjer u kojem podaci više variraju.



Slika 3-6 Kovarijanca varijabli

### 3.3. Analiza svojstvenih komponenti na skupu znakova

Broj slika znakova u korištenoj bazi je  $N=574$ . Svaka slika sastoji se od  $n$  slikovnih elemenata (piksela). Odabrani format slike za raspoznavanje je veličine  $70 \times 70$  slikovnih elemenata, što znači da je  $n=4900$ . Slika se može prikazati u jednodimenzionalnom obliku kao stupčasti vektor s  $n$  elemenata. Takvi vektori označeni su s  $\Gamma_1 \dots \Gamma_N$  za svaku normaliziranu sliku iz baze te organizirani u matricu  $S$  dimenzija  $n \times N$ . [5]

$$S = [\Gamma_1 \dots \Gamma_N] \quad (3.14)$$

U Matlabu je pretvorba slike u vektor vršena pomoću naredbe  $B = \text{reshape}(A, m, n)$ . Funkcija *reshape* kao argument prima varijablu  $A$  koja predstavlja matricu, u našem slučaju učitane slike znaka iz baze. Varijable  $m$  i  $n$  predstavljaju dimenzije matrice u koju želimo pretvoriti ulaznu sliku, u ovom slučaju su to dimenzije  $4900 \times 1$ . Tako preoblikovana matrica, sprema se u

varijablu  $B$ . Osim pretvorbe dimenzija, dobivene vektore transponiramo (u Matlabu  $B'$  označava transponiranje) kako bi smo dobili stupčaste vektore (dimenzija  $1 \times 4900$ ) te ih poslagali u matricu  $S$  (dimenzija  $4900 \times 574$ ). Slijedeći korak je pronalaženje prosječne slike znaka  $\Psi$ .

$$\Psi = \frac{1}{N} \sum_{i=1}^N \Gamma_i \quad (3.15)$$

Prosječna slika znaka također se dobiva u obliku vektora. Taj vektor je moguće jednostavno opet prikazati u obliku slike i tako možemo dobiti prosječnu sliku znaka. Prosječna slika znaka u našoj bazi prikazana je na slici 3-7.



Slika 3-7 Prosječna slika

Prosječna slika u Matlabu izračunata je naredbom  $mpro = mean(S,2)$ . Funkcija  $mean$  računa srednju vrijednost svakog stupca matrice  $S$ , te takav vektor redak sprema se u varijablu  $mpro$ .

Potom je potrebno naći vektore koji predstavljaju razliku pojedine slike u bazi i prosječnog znaka kao

$$\Phi_i = \Gamma_i - \Psi, i = 1, 2, \dots, N \quad (3.16)$$

Od ovakvih vektora tvorimo matricu  $A$  kao

$$A = [\phi_1, \phi_2, \dots, \phi_N] \quad (3.17)$$

Matrica  $A$  ima  $N$  stupaca i  $n$  redaka, dakle u ovom slučaju ona je dimenzija  $4900 \times 574$ . Da bi se pronašla svojstvena baza prostora trokutastih znakova,

potrebno je pronaći najveće svojstvene vektore  $v_i$  kovarijacijske matrice  $C$  koja se može dobiti kao

$$C = AA^T \quad (3.18)$$

Matrica  $C$  dimenzija je  $n \times n$ , u ovom slučaju  $4900 \times 4900$ . Računanje svojstvenih vektora matrice  $C$  bilo bi vrlo dugotrajno zbog njenih velikih dimenzija. Zato se svojstvene vrijednosti i svojstveni vektori matrice  $C$  ne računaju izravno nego pomoću matrice  $C'$

$$C' = A^T A \quad (3.19)$$

Do navedene pomoćne matrice  $C'$  došlo se slijedećim trikom.

Razmotrimo na trenutak, umjesto svojstvenih vektora matrice  $C = AA^T$  svojstvene vektore matrice  $A^T A$ .

$$A^T A v_i = \lambda_i v_i, i = 1, 2, \dots, N \quad (3.20)$$

Dimenzije ove matrice su  $N \times N$  (u našem slučaju  $574 \times 574$ ). Vidimo da će izračunavanje svojstvenih vektora ove matrice ići daleko brže nego izračunavanje svojstvenih vektora matrice  $C$ . Sada množenjem gornje jednadžbe s  $A$  s lijeve strane dobivamo

$$AA^T A v_i = \lambda_i A v_i, i = 1, 2, \dots, N \quad (3.21)$$

Usporedimo li dobivenu jednadžbu s definicijom svojstvenih vektora i svojstvenih vrijednosti možemo zaključiti da su vektori  $A v_i$  zapravo svojstveni vektori matrice  $C = AA^T$ .

Također se može pokazati da su svojstvene vrijednosti odgovarajućih svojstvenih vektora za obje matrice jednake.

Označimo li svojstvene vrijednosti i normalizirane svojstvene vektore matrice  $A^T A$  kao  $\lambda_i'$  i  $v_i'$  tada  $N$  najvećih svojstvenih vrijednosti  $\lambda_i$  i odgovarajuće normalizirane svojstvene vektore  $v_i$  matrice  $C = AA^T$  možemo dobiti kao

$$\lambda_i = \lambda_i' \quad i = 1, 2, \dots, N \quad (3.22)$$

$$v_i = \frac{1}{\sqrt{\lambda_i}} A v_i' \quad i = 1, 2, \dots, N \quad (3.23)$$

Računanje svojstvenih vektora i svojstvenih vrijednosti Matlab računa naredbom  $[vv \quad dd] = eig(C)$ . Naredba *eig* stvara dvije matrice. Matricu *vv* čiji su stupci svojstveni vektori, te matricu *dd* na čijoj se dijagonali nalaze svojstvene vrijednosti matrice  $C$ . Obje matrice su istih dimenzija kao i kovarijacijska matrica  $C$ .

Da bi se vidjelo koliko je na brzinu izvođenja programa utjecala promjena dimenzija kovarijacijske matrice  $C$ , korištene su funkcije *tic* i *toc* koje u kombinaciji mjere vrijeme izvođenja željene naredbe. Da se izvede naredba  $eig(Z)$ , gdje je  $Z$  matrica dimenzija  $574 \times 574$  potrebno je 0.15 sekunda, dok se naredba  $eig(C)$ , gdje je  $C$  matrica dimenzija  $4900 \times 4900$  izvodi 150 sekunda.

Na slici 3-8. prikazani su primjeri dobivenih svojstvenih znakova.



Slika 3-8 Svojstveni znakovi

Ovi svojstveni vektori istih su dimenzija kao i oni koji prikazuju slike (4900 elemenata) i također se mogu jednostavno prikazati u obliku slika. Kada se to učini, slike koje odgovaraju najvećim svojstvenim vektorima sličit će znakovima. Te slike, odnosno odgovarajući svojstveni vektori nazivaju se svojstveni znakovi. Svojstveni vektori koji odgovaraju sve manjim svojstvenim vrijednostima sve manje podsjećaju na znakove, a sve više na šum. To odgovara ideji da će upravo u prvim svojstvenim vektorima biti sadržane bitne značajke skupa uzoraka.

Svojstveni znakovi predstavljaju svojstvenu bazu prostora u kojem će se prikazivati i uspoređivati slike znakova. Svaku sliku  $\Phi$  iz svojstvene baze projiciramo na  $k$ -tu komponentu prostora svojstvenih znakova kako slijedi:

$$\omega_k = v_k^T \Phi, k = 1, 2, \dots, m \quad (3.24)$$

Ovdje  $m$  predstavlja dimenzionalnost uzetu za prostor svojstvenih znakova. Što se veći  $m$  odabere to će se više svojstvenih znakova koristiti i prostor će biti to veće dimenzionalnosti. Ideja metode svojstvenih znakova je da se kao baza uzimaju samo oni svojstveni vektori koji su značajni za prikaz znakova. Projekciju



slike  $\Phi$  (iz 3.23) na prostor svojstvenih znakova dobivamo kao vektor  $\Omega$  pojedinačnih projekcija  $\omega_i$ .

$$\Omega^T = [\omega_1, \omega_2, \dots, \omega_m] \quad (3.25)$$

Za ulaznu sliku koju želimo raspoznati potrebno je pronaći vektor koji predstavlja razliku ulazne slike i prosječne slike baze.

$$\Phi = \Gamma - \Psi \quad (3.26)$$

Dobivenu razliku projiciramo u prostor svojstvenih znakova prema formuli 3.24. Dobivamo težinske faktore ulazne slike koje ćemo označiti kao  $\Omega_k$ .

Da bi se odredilo kojoj klasi pripada ulazna slika znaka korišten je klasifikator 1-NN. Više o klasifikacija algoritmom k-NN rečeno je u poglavlju 3.1.3. Neklasificirani znak iz skupa za testiranje pripada onoj klasi znaka iz skupa za učenje kojoj pripada jedan najbliži susjed. Najbliži susjed je onaj koji ima najmanju euklidsku udaljenost do neklasificiranog uzorka.

$$\varepsilon_k = \|\Omega - \Omega_k\| \quad (3.27)$$

Slika  $\Phi$  može se također rekonstruirati iz prikaza u prostoru znaka kao

$$\Phi_r = \sum_{k=1}^m \omega_k v_k \quad (3.28)$$
















## 4. Eksperimentalni rezultati

Metoda PCA ili analiza svojstvenih komponenti primijenjena je za klasifikaciju trokutastih prometnih znakova. Skup za učenje iznosi 474 slike znaka te je odvojen je od skupa za testiranje za testiranje od 50 slika znaka. Razlikujemo 20 klasa znakova uz naglasak da broj slika za svaku klasu znaka nije konstantan.

Rezultati variraju u ovisnosti o broju uzetih svojstvenih vektora. Maksimum točnosti raspoznavanja postiže se za 50 svojstvenih vektora kada točnost iznosi oko 68 %. U tablici 4-1 možemo vidjeti numeričke rezultate prepoznavanja po klasama. Iz priloženog možemo uočiti da za klase A05, A07, A17, A22, A45 točnost prepoznavanja iznosi 0%. Navedenih klasa u skupu za učenje ima relativno malo i loše su kvalitete što je i uzrok netočnoj klasifikaciji. Točnost prepoznavanja klase A03 iznosi 33.3%. Slike znakova u skupu su jasne i kvalitetne te ih ima 29, no klasifikacija je loša zbog sličnosti znaka s klasama A04 te A05.

**Tablica 4-1 Numerički rezultati prepoznavanja po klasama znakova**

Klasa znaka	Slika znaka	Broj znakova u skupu za testiranje	Broje znakova u skupu za učenje	Točnost prepoznavanja
A01		4	36	75%
A03		3	29	33.3%
A04		3	13	66.6%
A05		2	11	0%
A07		1	2	0%

A08		3	57	100%
A09		4	52	25%
A10		4	42	75%
A11		4	43	75%
A12		2	12	100%
A13		1	4	100%
A14		3	20	100%
A17		1	3	0%
A20		4	49	100%
A22		1	3	0%
A23		1	6	100%
A33		2	29	100%
A34		3	21	100%
A44		3	39	66.6%
A45		1	3	0%

U tablici 4-2 možemo vidjeti rezultate klasifikacija u ovisnosti o različitim svojstvenim vektorima. Vidimo da odabir 20 svojstvenih vektora smanjuje točnost raspoznavanja, dok 150 svojstvenih vektora ju povećava no ne znatno pa je 50 optimalan izbor broja svojstvenih vektora.

**Tablica 4-2 Točnost raspoznavanja u ovisnosti o odabranom broju svojstvenih vektora za skup za učenje od 474 slike**

Broj svojstvenih vektora	20	50	150
Točnost raspoznavanja	64%	68%	70%

Točnost raspoznavanja uvelike ovisi i o veličini skupa za učenje te broju klasa. Rezultati koji se dobiju nad manjim skupovima za učenje dani su tablicom 4-3.

**Tablica 4-3 Točnost raspoznavanja u ovisnosti o veličini skupa za učenje te broju klasa**

Broj znakova u skupu za učenje	21	72	194
Broj klasa	7	6	7
Točnost raspoznavanja	95%	80%	75%

Za manje skupove učenja te manji broj klasa kao što je bilo i očekivano, točnost raspoznavanja je veća. Za skup učenja veličine 21 znak korišteno je 10 svojstvenih vektora, dok je za ostale skupove (72 i 194) odabrano 50 svojstvenih znakova.

Da bi se provjerilo da li je odabrana rezolucija slike (70×70 slikovnih elemenata) zaista najpogodnija za raspoznavanje znakova izvršeni su eksperimenti s metodom svojstvenih znakova u kojima su normalizirane slike znakova bile različite veličine.

Rezultati za veličinu slika  $35 \times 35$  dani su tablicom 4-4., a rezultati za veličinu slika  $140 \times 140$  dani su tablicom 4-5.

**Tablica 4-4 Rezultati metode svojstvenih znakova za slike normalizirane na veličinu  $35 \times 35$**

Broj svojstvenih vektora	20	50	150
Točnost raspoznavanja	58%	68%	72%

**Tablica 4-5 Rezultati metode svojstvenih znakova za slike normalizirane na veličinu  $140 \times 140$**

Broj svojstvenih vektora	20	50	150
Točnost raspoznavanja	62%	62%	62%

Eksperimentima smo utvrdili da veličina slike ne utječe bitno na točnost raspoznavanja. Za veće slike ( $140 \times 140$ ) točnost raspoznavanja se smanjuje. Za veličinu  $35 \times 35$  povećava se za 150 svojstvenih vektora te iznosi 72%. Očito je da je odabir rezolucije slike  $70 \times 70$  i broj svojstvenih vektora 50 optimalan izbor.

Zanimljivo je bilo vidjeti utjecaj maske koju smo koristili pri normalizaciji slika za uklanjanje pozadine, na točnost raspoznavanja. Za 50 svojstvenih vektora točnost je iznosila 70%. Vidi se da se točnost povećala, ali ne znatno. Točnije rečeno za jednu sliku. Ako se obrati pažnja koja je slika bila prepoznata očito se radi o dosta nepravilnoj slici (slika 4.1.). To je znak kojeg prekriva grana te je nejasan i ljudskom oku. Kako u skupu učenja imamo sličan znak iz iste klase, metoda ga je prepoznala te je tu pozadina imala veliku ulogu u raspoznavanju. Možemo dakle zaključiti da nam uklanjanje pozadine ipak pomaže u točnosti raspoznavanja jer time eliminiramo nepotrebne dijelove slike koji mogu zavarati pri raspoznavanju.



Slika 4-1 Nepravilna slika znaka

Neki testni primjeri koje algoritam dobro prepoznaje i njihova rekonstrukcija u ovisnosti o broju svojstvenih vektora, dani su tablicom 4-6.



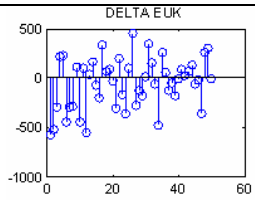
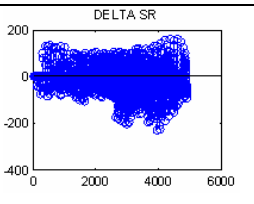


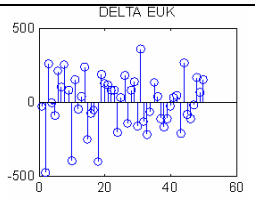
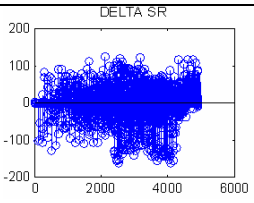

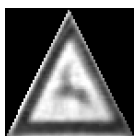
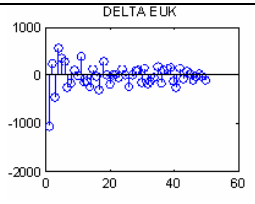
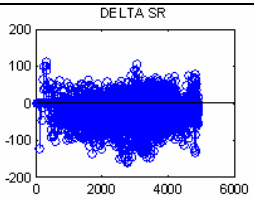

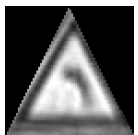
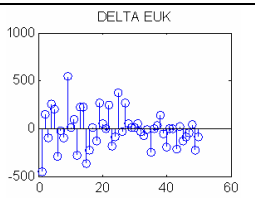
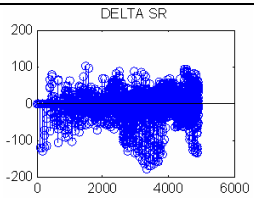


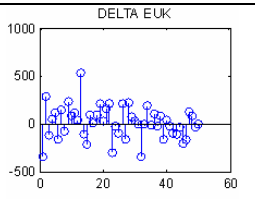
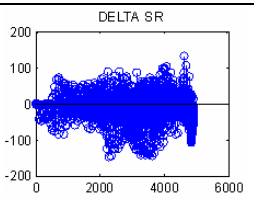
**Tablica 4-6 Slike iz skupa za testiranje i njihove rekonstrukcije s različitim brojem svojstvenih vektora**

Ulazna slika	Prepoznata slika iz baze (m=50)	Prepoznata slika iz baze (m=100)	Prepoznata slika iz baze (m=200)

Iz prethodnih primjera možemo uočiti da je rekonstrukcija to bolja što je broj svojstvenih vektora veći, što je i bilo za očekivati.

Tablicom 4-7 dani su grafovi euklidske udaljenost kao i razlike normalizirane ulazne slike i rekonstruirane prepoznate slike.

**Tablica 4-7 Ulazna i prepoznata slika iz baze te pripadni grafovi**

Ulazna slika	Prepoznata slika	Delta Euk	Standardna devijacija (Delta Euk)	Delta Sr	Standardna devijacija (Delta Sr)
			256.85		38.38
			178.75		29.12
			194.45		27
			332.62		24.35
			332.62		27.77



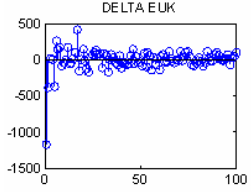
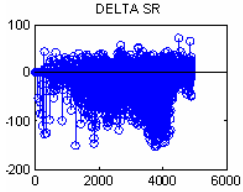

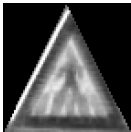
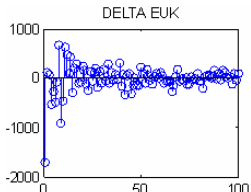
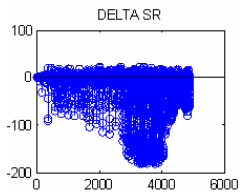
Graf Delta Euk prikazuje razliku težinskih faktora ulazne slike i prepoznate slike iz skupa za učenje. Prepoznata je ona slika koja ima najmanju euklidsku udaljenost od ulazne slike (1-NN klasifikator). Na apscisi se nalazi broj odabranih

svojstvenih vektora, u ovom slučaju 50. Ordinata prikazuje raspon razlike iznosa prvih 50 težinskih faktora ulazne slike znaka i prepoznate slike. Što su slike sličnije to je razlika manja. Standardne devijacije za dobro klasificirane znakove možemo vidjeti u tablici 4-7.



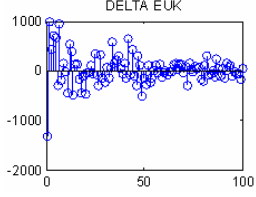
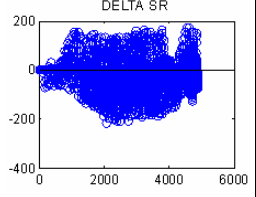


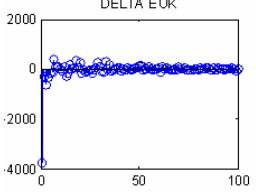
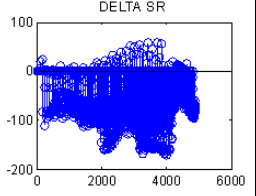


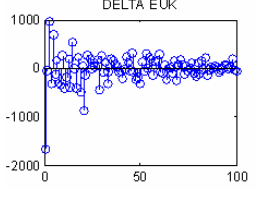
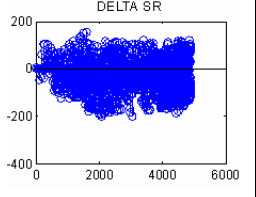
Graf Delta Sr prikazuje razliku ulazne i rekonstruirane prepoznate slike za svaki slikovni element. Slikovni elementi su u intervalu [0,255] jer analiziramo sive slike. Taj raspon nalazi se na ordinati. Na apscisi je ukupni broj slikovnih elemenata 4900 jer su slike dimenzija 70x70. Rekonstrukcija slike je bolja što je razlika slikovnih elemenata ulazne i prepoznate slike manja. Možemo i uočiti da je za prvih 500 slikovnih elemenata razlika manja od ostalih slikovnih elemenata. Uzrok tomu je maska koja otklanja pozadinu te sama činjenica da su svi znakovi trokutni, imaju isti oblik.

Sve gore odabrane slike su jasne, znakovi su dobro geometrijski centrirani pa je proces prepoznavanja i rekonstrukcije protekao bez problema. Za slijedeće primjere odabrat ćemo neke zamućene, nakrivljene te slabo osvjetljene slike znakova. Tablica 4.8. prikazuje navedene primjere.

**Tablica 4-8 Testni primjeri slika lošije kvalitete**

Ulazna slika	Prepoznata slika	Opis	Delta Euk	Standardna devijacija (Delta Euk)	Delta Sr	Standardna devijacija (Delta Sr)
		Zamućena slika		283.5		49
		Slabo osvjetljena		284.6		44.9



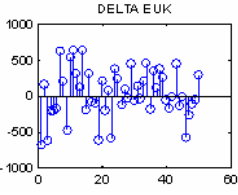
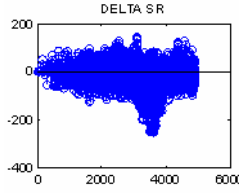


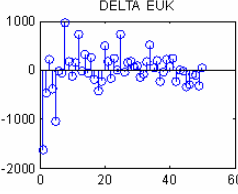
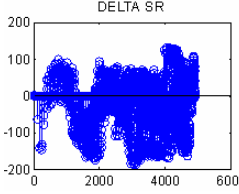


		Prekriven dio znaka		266.1		49.8
		Normalna		150.2		29.8
		Zarotirani znak		262.3		43.7

Na gornjim primjerima možemo uočiti da za slike lošije kvalitete algoritam relativno dobro prepoznaje, ali rekonstrukcija je loša (odabrano je 100 svojstvenih vektora). Možemo uočiti da su standardne devijacije grafa DELTA EUK kao i grafa delta DELTA SR veće od onih u tablici 4-7. Neobično je što se znak iz klase A03 dobro klasificira, ulazna slika je zadovoljavajuće kvalitete, a rekonstrukcija je loša. Uzrok tomu je sličnost ulazne slike s prosječnom slikom skupa za učenje.

Zanimljivo je još pogledati rekonstruiranu sliku za nepoznat znak kao i za sliku koja nije znak te njihove pripadne grafove. Prikazani su tablicom 4-9.

**Tablica 4-9 Rezultati nepoznatog znaka i slike koja nije znak**

Ulazna slika	Prepoznata slika	Opis	Delta Euk	Standardna devijacija (Delta Euk)	Delta Sr	Standardna devijacija (Delta Sr)
		Nepoznati znak		443.21		51.9
		Nepoznata slika		514.63		76.05


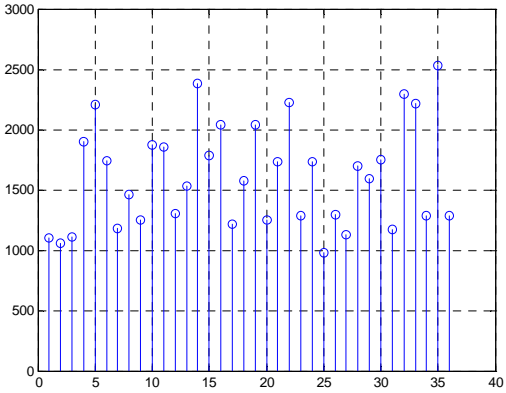

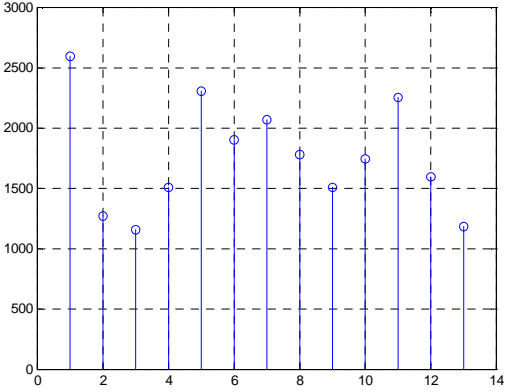

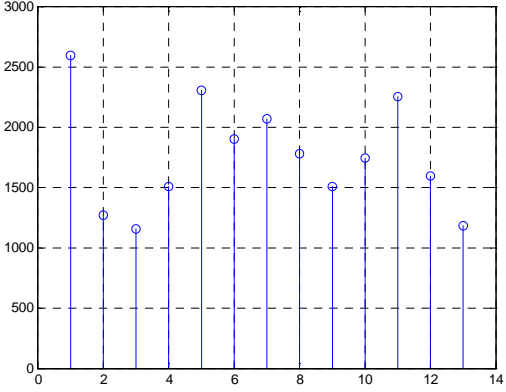
Vrijednosti normi za slike navedene u tablicu 4.9. znatno su veće od onih navedenih u tablicama 4.7. i 4.8.


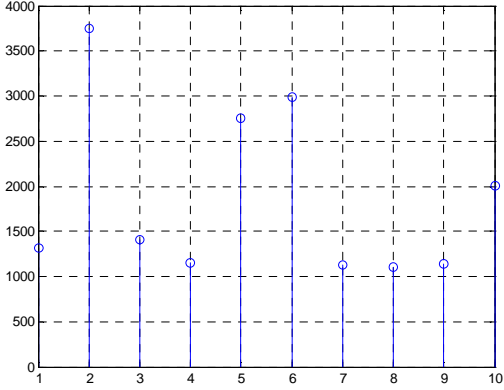

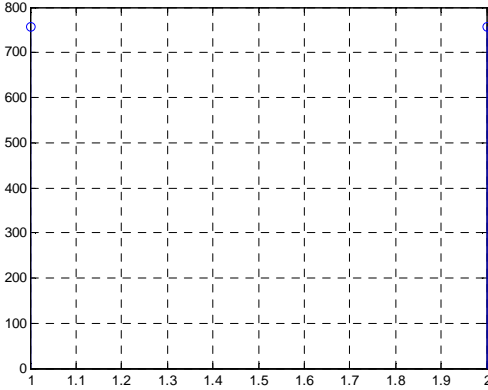

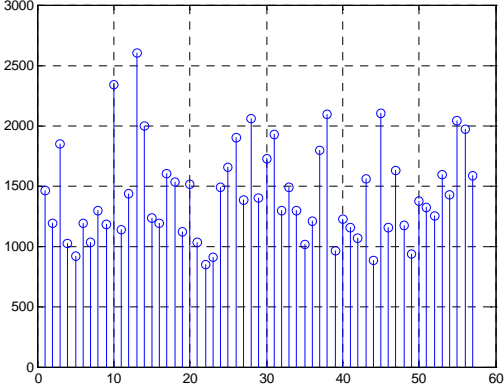

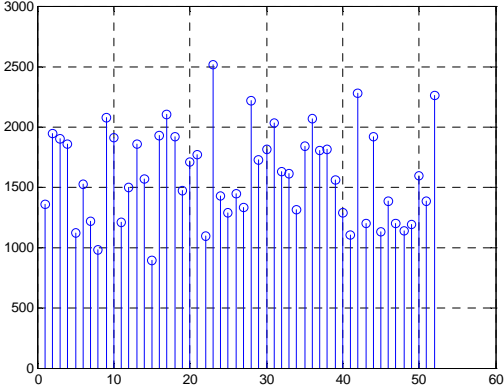
Promatrana je disperzija znakova unutar klase. Za svaku klasu izračunat je vektor srednjih vrijednosti težinskih faktora svih slika unutar te klase.. Svaki element tog vektora predstavlja srednju vrijednost težinskih faktora u tom retku svih slika iz te klase. Srednja vrijednost težinskih faktora  $\bar{\Gamma}_k$  izračunata je formulom 4.1. gdje je  $\Gamma_{ki}$  vektor redak težinskih faktora svih slika iz klase, a  $n_k$  broj slika u klasi. Kao mjera disperzije korištena je euklidska udaljenost vektora težinskih faktora pojedine slike i vektora srednjih vrijednosti težinskih faktora svih slika iz pojedinačne slike (formula 4.2). Dobiveni rezultati za svih klasa prikazani su tablicom 4-10.


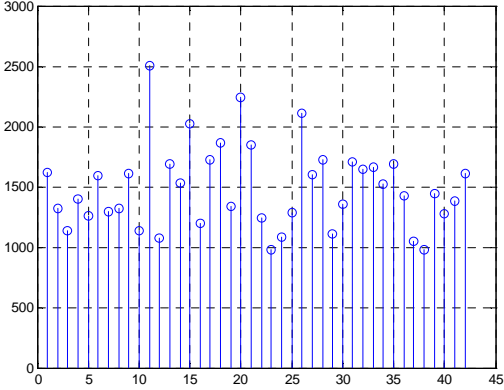


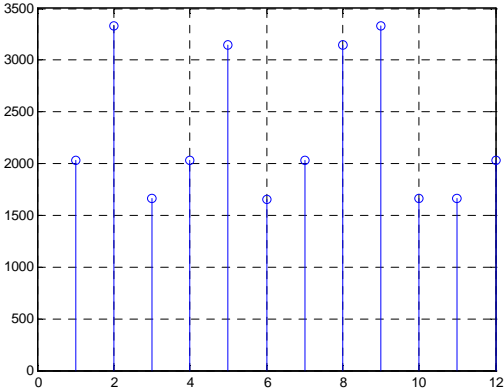

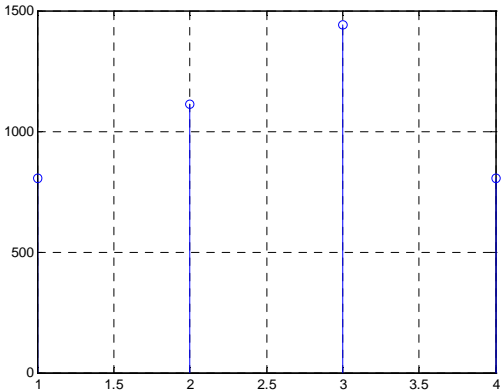
$$\bar{\Gamma}_k = \frac{\sum_{i=1}^m \Gamma_{ki}}{n_k} \quad (4.1)$$


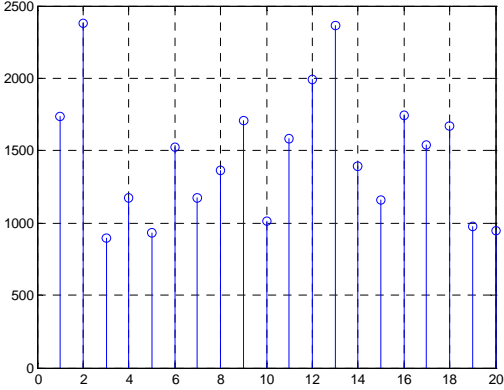

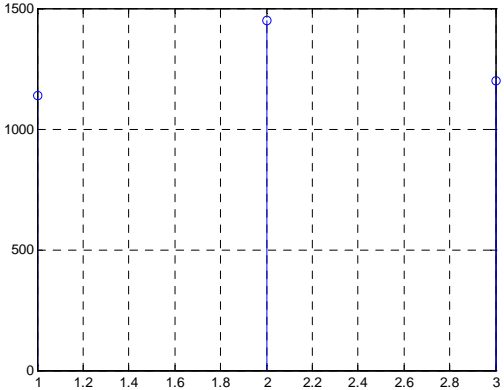

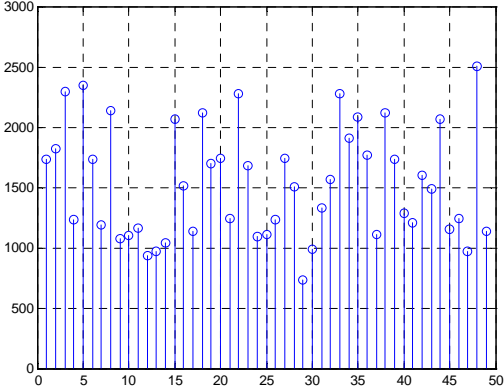

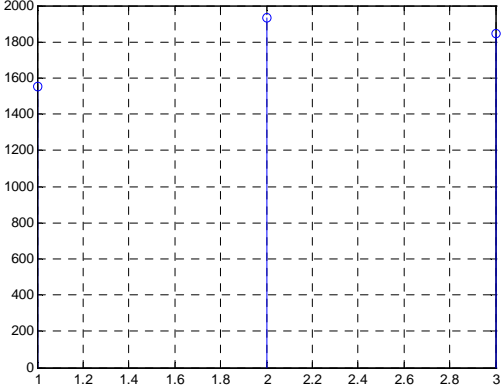
$$\varepsilon_k = \|\Gamma_{ki} - \bar{\Gamma}_k\| \quad (4.2)$$


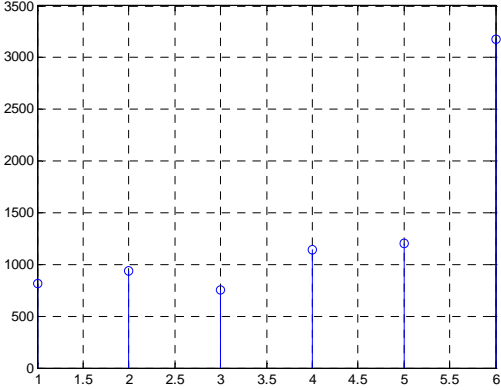

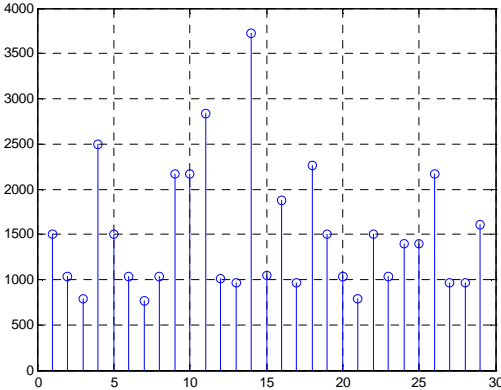

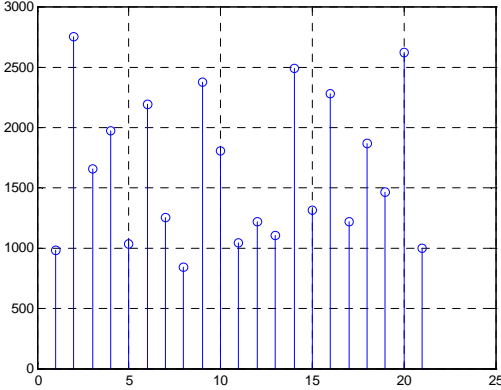

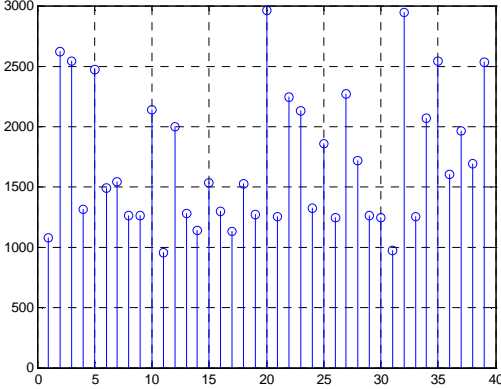
Tablica 4-10 Disperzija unutar klasa

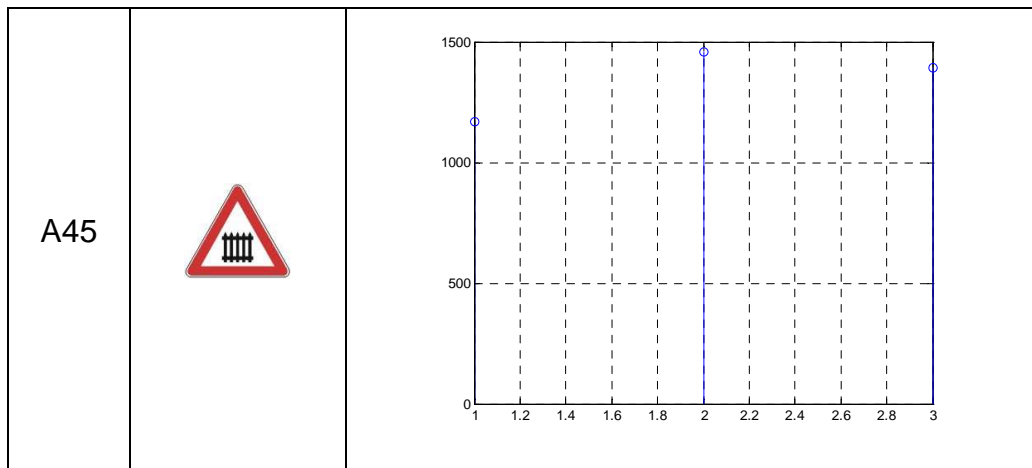
Klasa	Slika znaka	Graf disperzije
A01		
A03		
A04		

<p>A05</p>		
<p>A07</p>		
<p>A08</p>		
<p>A09</p>		

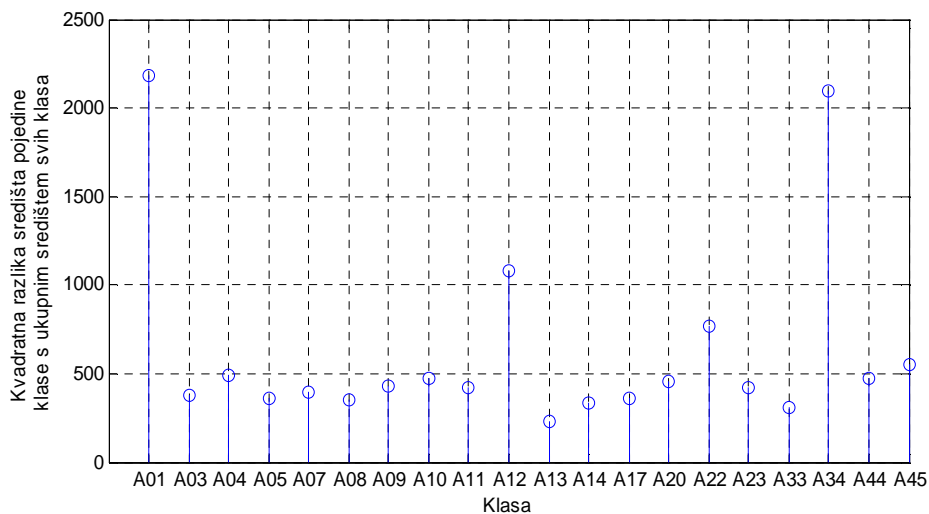
<p>A10</p>		
<p>A11</p>		
<p>A12</p>		
<p>A13</p>		

<p>A14</p>		
<p>A17</p>		
<p>A20</p>		
<p>A22</p>		

<p>A23</p>		
<p>A33</p>		
<p>A34</p>		
<p>A44</p>		



Za potrebe analize dobivenih rezultata na slici 4-2 prikazana je distribucija udaljenosti središta pojedine klase od zajedničkog središta svih klasa. Središte pojedine klase predstavlja vektor srednjih vrijednosti težinskih faktora svih slika unutar te klase. Na osi apscisa prikazanog grafa nalaze se pojedine klase, a na ordinati kvadratna razlika vektora srednjih vrijednosti težinskih faktora klase s srednjom vrijednošću težinskih faktora svih klasa zajedno.



S

Slika 4-2 Distribucija težinskih faktora pojedinih klasa

Varijanca  $\sigma^2$  središta klasa u svojstvenom prostoru izračunata je formulom 4.3. i iznosi 628.45.  $\bar{\Gamma}_k$  predstavlja središte klase, a  $\bar{\Gamma}$  središte svih klasa zajedno.

$$\sigma^2 = \frac{\sum_{i=1}^n (\bar{\Gamma}_k - \bar{\Gamma})^2}{n} \quad (4.1)$$



## 5. Programska izvedba

Cijela implementacija algoritma načinjena je u Matlabu, programskom jeziku za tehničke proračune. Kako je u prethodnim poglavljima uz teoretsko objašnjenje slijedila i izvedba bitnih dijelova algoritma u Matlabu, u ovom poglavlju se navode samo upute za korištenje načinjenog programa.

### 5.1. Upute za korištenje

Pokretanjem Matlaba dobiva se njegov komandni prozor. Potrebno je učitati m-datoteku u kojoj se nalazi program, a naziva se PCA.m. Nakon pokretanja programa traži se korisnika da unese slijedeće parametre:

- 1) Putanja do direktorija u kojem se nalaze slike iz skupa za učenje
- 2) Broj slika u skupu za učenje
- 3) Broj svojstvenih vektora
- 4) Putanja do direktorija u kojem se nalaze slike iz skupa za testiranje
- 5) Broj slika u skupu za testiranje

Nakon obrade svih ulaznih slika program vraća točnost raspoznavanja. Ako se želi provjeriti klasifikacija, u komandnom prozoru utipkamo `klasifikacija`. Vrijednost 1 označava da je znak ispravno klasificiran, a 0 da nije dobro klasificiran..

Algoritam uključuje grafički prikaz svih slika (iz skupa za učenje), njihove faze normalizacije, prosječnu sliku, masku te dobivene svojstvene znakove. Za testne slike grafički se ispisuje ulazna slika, normalizirana slika i rekonstruirana slika. Za svaku od njih prikazuje se graf euklidske udaljenosti ulazne slike s prepoznatom slikom iz skupa za učenje te graf razlike normalizirane ulazne i rekonstruirane slike.

Grafički prikaz usporava izvođenje programa ako se radi o velikoj bazi slika, pa po potrebi dijelove algoritma koji obuhvaćaju grafički ispis možemo kontrolirati varijablom `ispis`. Ako je varijabla `ispis` postavljena na 1 slike će se grafički ispisivati, a ako je postavljena na 0 ispis izostavljamo.

## 6. Zaključak

PCA (Principal Component Analysis) ili analiza glavnih komponenti je statistički postupak za reduciranje dimenzije podataka. Pri reduciranju se želi izgubiti što je manje moguće informacija i ovaj postupak upravo to omogućava. Metoda je našla veliku primjenu i u području raspoznavanja uzoraka.

U ovom završnom radu obrađena je primjena metode PCA nad skupom znakova. Skup znakova čine trokutni prometni znakovi snimljeni iz vozila u pokretu. Poradi poboljšanja performansi, slike su normalizirane, odnosno uklonjeni su dijelovi slike koji ništa ne govore o samom znaku. Nad takvim skupom primijenjena je analiza svojstvenih komponenti te potom iscrpno testirana na primjenu u klasifikaciji. Postignuta točnost raspoznavanja znakova je oko 68%. Točnost raspoznavanja varira ovisno o kvaliteti i dimenzijama ulaznih slika kao i slika iz skupa za učenje, odabranom broju svojstvenih vektora, veličini skupa za učenje te broju klasa.

Razvijeni postupak klasifikacije mogao bi biti dobra polazna točka za budući rad. Slične klase znakova mogle bi se objediniti u prvom koraku analize, a konačnu odluku mogla bi se donijeti posebnim postupkom. Konačno, očekuje se da bi se bolja klasifikacija postigla analizom koja bi garantirala bolju razdvojenost razreda uzoraka u prostoru niže dimenzionalnosti.

## 7. Literatura

- [1] Elezović, N. Linearna algebra: Svojstveni vektori. 0.izdanje. Zagreb: Element, 1995.
- [2] Fratrić, I. Sustav za verifikaciju osoba temeljen na raspoznavanju lica. Diplomski rad. FER, 2003.
- [3] Rojković, D. Raspoznavanje prometnih znakova križnom korelacijom. Završni rad. FER, 2009.
- [4] Šošić, I., Serdar, V. Uvod u Statistiku, Zagreb: Školska knjiga, 1997.
- [5] Turk, M., Pentland, A. Eigenfaces for Recognition. Journal of Cognitive Neuroscience. 71-86.
- [6] Conversion from RGB to HSI, Color Processing, 2007.
- [7] <http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm>
- [8] [www.cs.otago.ac.nz/cosc453/student\\_tutorials/principal\\_components.pdf](http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf)
- [9] [www.cs.bham.ac.uk/~jlw/vision/face\\_recognition.ppt](http://www.cs.bham.ac.uk/~jlw/vision/face_recognition.ppt)
- [10] [http://fourier.eng.hmc.edu/e161/lectures/color\\_processing/node3.html](http://fourier.eng.hmc.edu/e161/lectures/color_processing/node3.html), 2007.

**Sažetak**

U ovom radu razmatra se raspoznavanje prometnih znakova primjenom analize svojstvenih komponenti. Skup slika znakova čine trokutni prometni znakovi snimljeni iz vozila u pokretu. Da bi se poboljšale performanse, slike su normalizirane. Nad takvim skupom primijenjena je metoda PCA, a potom iscrpno testirana primjena u klasifikaciji. Implementacija metode napisana je u programskom jeziku Matlab.

**Ključne riječi**

Analiza svojstvenih komponenti, normalizacija, raspoznavanje prometnih znakova, računalni vid, Matlab

**Abstract**

In this paper we describe traffic signs recognition using principal component analysis. Set of images consist of triangular traffic signs which are recorded from car in movement. In order to improve performances, images are normalized. Method PCA is applied on normalized set of images and tested for classification. Method is implemented in programming language Matlab.

**Key words**

Principal component analysis, normalization, traffic signs recogniton, computer vision, Matlab