

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 315

**EKSPERIMENTALNA EVALUACIJA
UTJECAJA ODABIRA ZNAČAJKI NA
REZULTATE RASPOZNAVANJA
PROMETNIH ZNAKOVA**

Ivana Sučić

Zagreb, lipanj 2011.

Zahvala

Zahvaljujem se svom mentoru Doc. dr. sc. Siniši Šegviću na savjetima i pomoći tijekom izrade diplomskog rada. Zahvaljujem se svojoj obitelji na pruženoj podršci i razumijevanju tijekom mog studiranja.

Sadržaj

1.	Uvod.....	1
2.	Značajke slike	2
2.1.	Obrada slike	2
2.1.1.	Označavanje prometnih znakova	2
2.1.2.	Ispravljanje smetnji uzrokovanih preplitanjem slike	3
2.2.	Histogram orijentacije gradijenta	4
3.	Stroj s potpornim vektorima	5
3.1.	Linearno odvojivi razredi.....	5
3.2.	Neodvojivi razredi.....	10
4.	Analiza svojstvenih komponenti	15
4.1.	Analiza svojstvenih komponenti na skupu znakova.....	18
5.	Fisherova linearna diskriminantna analiza	21
5.1.	FLD za više razreda	26
5.2.	Usporedba FLD i SVM.....	29
6.	Evaluacija.....	31
6.1.	Skup za učenje	31
6.2.	Odabir ulaznih značajki	31
6.2.1.	Histogrami orijentacije gradijenata	31
6.3.	Klasifikacija.....	32
6.3.1.	Skup za testiranje.....	32
6.3.2.	Stroj s potpornim vektorima.....	33
6.3.3.	Analiza svojstvenih komponenti	34
6.3.4.	Fisherova linearna diskriminanta.....	37
6.3.5.	Veličina skupa za učenje	40
6.3.6.	Problem nepreciznog lociranja	41

6.3.7. Klasifikacija SVM-om za više razreda	45
7. ZAKLJUČAK	49
8. LITERATURA.....	50
Dodatak A: Opis implementacije	51
Označavanje prometnih znakova.....	51
Obrada slike.....	51
Histogram orijentacije gradijenta.....	52
Stroj s potpornim vektorima	53
Analiza svojstvenih komponenti	54
Linerana diskriminativna analiza.....	58
Naslov, sažetak i ključne riječi.....	59
Abstract, keywords, title	60

Popis oznaka i kratica

DAGSVM	Directed Acyclic Graph Support Vector Machines
FLD	Fisher Linear Discriminant
HOG	Histogram of Gradient
k-NN	k Nearest Neighbor
KKT	Karush-Kuhn-Tucker
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
PIL	Python Imaging Library
SVM	Support Vector Machine
VC	Vapnik-Chervonenkis

Popis tablica

<i>Tablica 1 Rezultati klasifikacije HOG+SVM.....</i>	<i>33</i>
<i>Tablica 2 Klasifikacija metodom FLD (64x64)</i>	<i>37</i>
<i>Tablica 3 Klasifikacija metodom FLD (48x48)</i>	<i>37</i>
<i>Tablica 4 HOG+FLD (64x64)</i>	<i>38</i>
<i>Tablica 5 HOG+FLD (48x48)</i>	<i>38</i>
<i>Tablica 6 Klasifikacija velikih klasa SVM-om.....</i>	<i>40</i>
<i>Tablica 7 Klasifikacija SVM s novim, proširenim skupom za učenje.</i>	<i>41</i>
<i>Tablica 8 Rezultati SVM klasifikacije korištenje strategije DAGSVM.....</i>	<i>46</i>
<i>Tablica 9 Statistika loše klasificiranih klasa.....</i>	<i>47</i>

Popis slika

<i>Slika 1</i> Primjer ulazne slike snimljene iz vozila u pokretu.....	2
<i>Slika 2</i> Preplitanje slike	3
<i>Slika 3</i> Originalna slika, slika nastala odbacivanjem parnih linija i slika nastala odbacivanjem neparnih linija.....	3
<i>Slika 4</i> Histogram orijentacije gradijenta	4
<i>Slika 5</i> Primjer optimalne hiperravnine za dva linearno odvojiva razreda	5
<i>Slika 6</i> Primjer decizijske funkcije.....	6
<i>Slika 7</i> Prikaz slučajevne povrede uvjeta $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1$	11
<i>Slika 8</i> Primjer 2D vektora.....	15
<i>Slika 9</i> Ortogonalna projekcija.....	16
<i>Slika 10</i> Dobar prikaz podataka	17
<i>Slika 11</i> Loš prikaz podataka	17
<i>Slika 12</i> Kovarijanca varijabli.....	18
<i>Slika 13</i> Projekcije uzoraka na dva različita pravca. Desna slika prikazuje bolju razdvojenost projekcija crvenih i zelenih točaka nego lijeva slika	21
<i>Slika 14</i> Projekcija uzorka $\vec{x}_i$ na pravac u smjeru $\vec{w}$	22
<i>Slika 15</i> Utjecaj udaljenosti srednjih projiciranih uzoraka na klasifikaciju uzoraka	23
<i>Slika 16</i> Da bi razdvajanje projiciranih uzoraka bilo najbolje, potrebno je imati što veću razliku srednjih vrijednosti različitih razreda i što manju raspršenost unutar projiciranih uzoraka istog razreda.	24
<i>Slika 17</i> Projekcija 3-D distribucije na 2-D potprostore koji su opisani.....	29
<i>Slika 18</i> Usporedba FLD i SVM.....	30
<i>Slika 19</i> Zastupljeni razredi znakova i broj primjeraka po razredima.....	31
<i>Slika 20</i> Raspored blokova.....	32
<i>Slika 21</i> Zastupljenost znakova u skupu za testiranje.....	32

<i>Slika 22 Neprecizno locirani znakovi.....</i>	<i>33</i>
<i>Slika 23 Rezultati klasifikacije SVM-a po klasama.</i>	<i>34</i>
<i>Slika 24 Uzorci iz klase A07.....</i>	<i>34</i>
<i>Slika 25 PCA klasifikacija za ručno i automatski označene znakove (64X64)..</i>	<i>35</i>
<i>Slika 26 PCA klasifikacija za ručno i automatski označene znakove (48X48)..</i>	<i>36</i>
<i>Slika 27 Rezultati klasifikacija PCA po klasama.....</i>	<i>36</i>
<i>Slika 28 Klasa A01</i>	<i>37</i>
<i>Slika 29 Točnost raspoznavanja anotacija u ovisnosti o izlučivanju značajki HOG-om.....</i>	<i>39</i>
<i>Slika 30 Točnost raspoznavanja detekcija u ovisnosti o izlučivanju značajki HOG-om.....</i>	<i>39</i>
<i>Slika 31 Klase sa velikim brojem uzoraka</i>	<i>40</i>
<i>Slika 32 10 novih pomaknutih slika</i>	<i>41</i>
<i>Slika 33 Usporedba SVM klasifikacije nad različito treniranim skupovima za učenje.</i>	<i>42</i>
<i>Slika 34 Usporedba klasifikacije detekcija nad različito treniranim skupovima za učenje PCA metodom.</i>	<i>43</i>
<i>Slika 35 Usporedba klasifikacije anotacija nad različito treniranim skupovima za učenje PCA metodom.</i>	<i>43</i>
<i>Slika 36 Usporedba klasifikacije metodom FLD nad različito treniranim skupovima za učenje.....</i>	<i>44</i>
<i>Slika 37 FLD nad neprotresenim skupom za učenje, klasifikacija 1-NN nad protresenim skupom za učenje</i>	<i>45</i>
<i>Slika 38 SVM za više razreda [3]</i>	<i>46</i>
<i>Slika 39 Uzorci iz klase A23.....</i>	<i>47</i>
<i>Slika 40 Uzorci iz klase A33.....</i>	<i>47</i>
<i>Slika 41 Maska korištena za uklanjanje pozadine.....</i>	<i>54</i>
<i>Slika 42 Primjena maske na ulaznoj slici</i>	<i>55</i>

<i>Slika 43 Slika znaka sa izjednačenim histogramom.....</i>	<i>55</i>
--	-----------

1. Uvod

Posljednjih nekoliko desetljeća, računala postaju sve prisutnija u ljudskoj svakodnevici. Postoji mnoštvo problema koje računala rješavaju puno efikasnije od ljudi i stoga su mnogi poslovi nezamislivi bez računalne potpore. Unatoč izrazitoj moći obrade podataka, računalima se i dalje ne mogu efikasno rješavati neki problemi koje ljudi rješavaju svakodnevno i bez velikog napora. Umjetna inteligencija, kao grana računalne znanosti, izučava takve probleme i traži njihova rješenja. Računalni vid je grana umjetne inteligencije koja razmatra probleme obrade slike i videa te izdvajanja korisnih informacija iz njih.

U ovom se radu rješava problem raspoznavanja prometnih znakova koji su automatski detektirani u sljedovima slika iz perspektive vozača. Razmatra se eksperimentalna evaluacija utjecaja odabira značajki na rezultate raspoznavanja. Ovaj je problem vrlo zanimljiv te bi njegovo rješenje imalo široku praktičnu primjenu: kao pomoć u održavanju prometnica, kao pomoć vozačima tijekom vožnje, kao sigurnosni sustav u vozilima, itd.

U okviru rada, proučeni su pristupi za raspoznavanje prometnih znakova. Uhodani su i ispitani postupci klasifikacije k-NN i SVM te pristupi za smanjenje dimenzionalnosti uzoraka (PCA, FLD). Predloženi postupci su primijenjeni na prometne znakove upozorenja, no primjenjivi su i na druge vrste prometnih znakova. Korišteni postupci, njihova teorijska osnova i dobiveni rezultati slijede u nastavku.

2. Značajke slike

Da bi se iz slike izdvojila korisna informacija, potrebno je odrediti način izlučivanja značajki iz slike. Također, prije izlučivanja značajki potrebno je obraditi sliku da bi se ispravile eventualne smetnje ili postigli željeni efekti. Ovo poglavlje obrađuje navedene probleme: obrada slike te izlučivanje značajki.

2.1. Obrada slike

2.1.1. Označavanje prometnih znakova

Slike nad kojima se vrši klasifikacija ručno su izdvojene iz video snimke. Za izdvajanje je korišten program Marker razvijen u sklopu rada [1]. Originalne slike nalik su slici 1.



Slika 1 Primjer ulazne slike snimljene iz vozila u pokretu

Pri izdvajanju iz video snimke, slike su imenovane po prometnom znaku koji se nalazi na njima. Slika znaka prikazanog slikom 1 ima naziv A01_0015. A01 jest klasa znaka, a 0015 redni broj znaka te klase. Zajedno sa izdvojenim slikama,

dobivena je datoteka koja sadrži pripadajuće koordinate znaka koji se nalazi na toj slici (A01_0015.bmp 1 600 168 84 80) (Dodatak A).

2.1.2. Ispravljanje smetnji uzrokovanih preplitanjem slike

Pri snimanju prve grupe video snimaka korištena je kamera koja na izlazu daje prepleteni video signal (eng. *interlacing*). Kao rezultat toga, javljaju se karakteristične smetnje na slici pri brzim pokretima kamere. Slika 2 prikazuje primjer slike sa smetnjom.



Slika 2 Preplitanje slike

U skupu slika koje se koriste u ovom radu, postoji okvirno 10% slika sa smetnjom prouzročenom preplitanjem te bi adekvatna obrada takvih slika poboljšala rezultate klasifikacije. Jednostavna metoda s dobrim rezultatima je odbacivanje polovice linija te interpolacija linija koje nedostaju (eng. *deinterlacing*) [2]. Na taj način od jedne slike nastaju dvije, jedna odbacivanjem parnih linija, druga odbacivanjem neparnih. Slika 3 prikazuje primjer slika nastalih odbacivanjem linija.

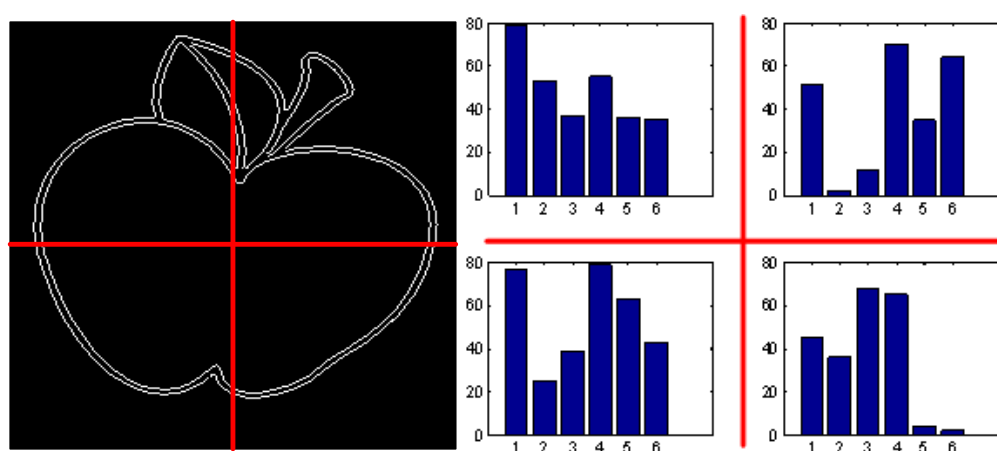


Slika 3 Originalna slika, slika nastala odbacivanjem parnih linija i slika nastala odbacivanjem neparnih linija

2.2. Histogram orijentacije gradijenta

Upotreba histograma je jednostavan način izlučivanja značajki nad dijelovima slike. Adekvatnim odabirom veličine i položaja segmenata slika nad kojima se rade histogrami značajki postiže se određena otpornost na pomake slike.

Posebno zanimljiv je histogram orijentacije gradijenata. Nakon odabira segmenata slike nad kojima se rade histogrami, za svaki slikovni element u segmentu, promatra se kut gradijenta. Histogram kutova gradijenata daje opis orijentacije rubova za dotični segment slike te time daje informaciju o obliku koji se nalazi u segmentu.



Slika 4 Histogram orijentacije gradijenta

Slika 4 prikazuje sliku koja je podijeljena na četiri segmenta te histograme smjera gradijenata za svaki od odabranih dijelova. Vrijednosti kutova linearno su raspoređene od 0 do -150 stupnjeva (1=0, 2=-30 itd) [3]. Histogrami vjerno prikazuju mogućnost izdvajanja informacije o smjeru rubova slike.

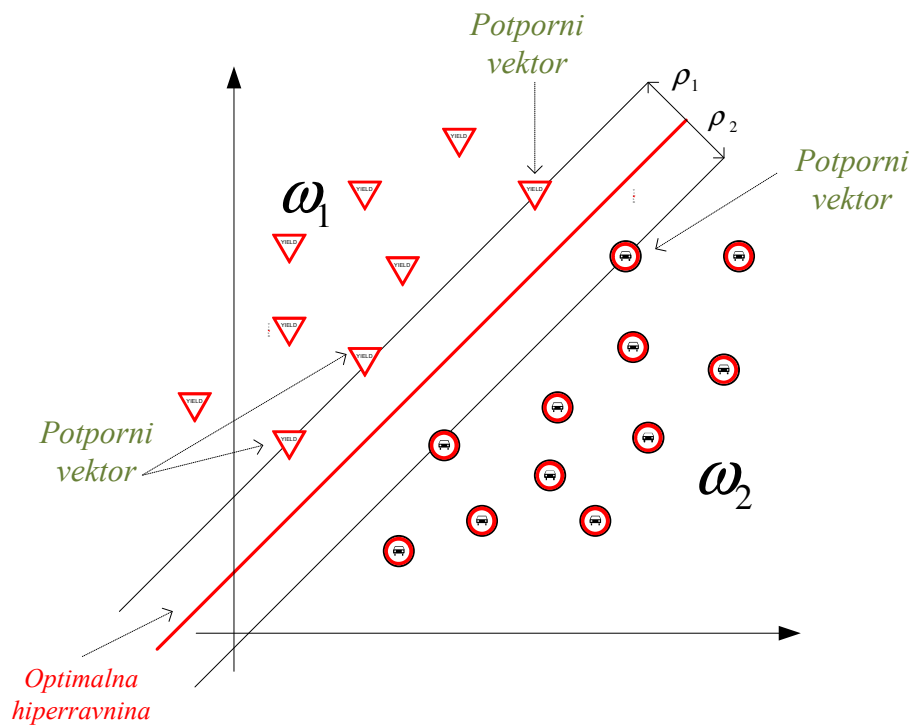
U navedenom primjeru histogram se temelji na smjeru to jest na rasponu od 180 stupnjeva. Detaljniju informaciju o rubovima može se dobiti proširenjem histograma na orijentaciju, to jest na 360 stupnjeva te proširenjem na dvodimenzionalni histogram dodavajući dimenziju iznosa gradijenta [4].

Uz histogram orijentacije gradijenta često se koriste i histogrami bazirani na značajkama iz različitih sustava boja. Takve značajke su otporne na rotaciju i translaciju objekta [5].

3. Stroj s potpornim vektorima

3.1. Linearno odvojivi razredi

Stroj s potpornim vektorima (eng. *Support vector machine*, SVM) izvorno je linearni stroj. Osnovna zamisao SVM-a je konstrukcija hiperravnine kao decizijske plohe, ali tako da je margina odvajanja između “pozitivne” i “negativne” skupine uzoraka za učenje maksimalna [6], [7].



Slika 5 Primjer optimalne hiperravnine za dva linearno odvojiva razreda

Neka je skup uzoraka za učenje $\{(\vec{X}_i, d_i)\}_{i=1}^N$, gdje je X_i , $i = 1, 2, \dots, N$, ulazni vektor uzoraka za i -ti primjer, a d_i željeni odgovor klasifikatora:

$$\begin{aligned}\omega_1 &\rightarrow d_i = +1 \\ \omega_2 &\rightarrow d_i = -1\end{aligned}\tag{3.1}$$

Pretpostavka je da su razredi ω_1 i ω_2 linearno odvojivi (slika 5). Jednadžba decizijske ravnine jest:

$$\vec{W}^T \vec{X} + b = 0\tag{3.2}$$

$\vec{W}$ je vektor težinskih koeficijenata, $\vec{X}$ je ulazni vektor te b je pomak. Vrijedi:

$$\vec{W}^T \vec{X} + b \geq 0 \text{ za } d_i = +1 \quad (3.3)$$

$$\vec{W}^T \vec{X} + b < 0 \text{ za } d_i = -1 \quad (3.4)$$

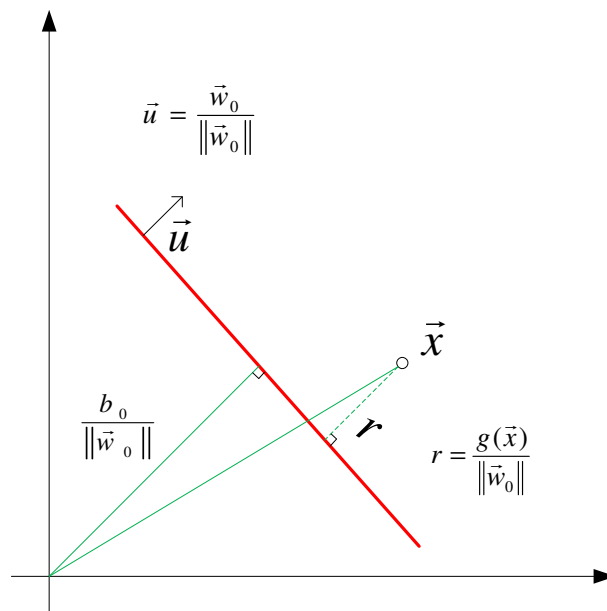
Za zadani vektor težinskih koeficijenata $\vec{W}$ i pomak b , udaljenost između hiperravnine i najbliže točke (uzorka) u n -dimenzionalnom prostoru naziva se margina odvajanja i označena je s ρ (slika 5). Cilj je pronaći onu hiperravninu za koju je margina odvajanja ρ maksimalna. Takva hiperravnina naziva se optimalna ravnina (slika 5). Optimalna hiperravnina sa optimalnim vrijednostima $\{\vec{W}_0, b_0\}$ je :

$$\vec{W}_0^T \vec{X} + b_0 = 0 \quad (3.5)$$

Decizijska funkcija

$$g(\vec{X}) = \vec{W}_0^T \vec{X} + b_0 \quad (3.6)$$

daje mjeru udaljenosti ulaznog vektora $\vec{X}$ od optimalne hiperravnine. Na slici 6 je primjer hiperravnine [7]. Svaka hiperravnina karakterizirana je svojim smjerom (označeno s $\vec{u}$) te točnim položajem u prostoru (označeno s b). Udaljenost vektora $\vec{X}$ od hiperravnine dana je sa r .



Slika 6 Primjer decizijske funkcije

Par $\{\vec{W}_0, b_0\}$ mora zadovoljavati slijedeća ograničenja:

$$\vec{W}_0^T \vec{X}_i + b_0 \geq 1 \text{ za } d_i = +1 \quad (3.7)$$

$$\vec{W}_0^T \vec{X}_i + b_0 < -1 \text{ za } d_i = -1 \quad (3.8)$$

Sada se može skalirati $\vec{W}_0$ i b_0 tako da za najbliže (hiperravnini $g(\vec{X})$) uzorke iz razreda ω_1 i ω_2 bude:

$$g(\vec{X}) = 1 \text{ za } \omega_1 \quad (3.9)$$

$$g(\vec{X}) = -1 \text{ za } \omega_2 \quad (3.10)$$

Za uzorke (točke u n -dimenzionalnom prostoru) iz skupa za učenje i to za one za koje vrijedi:

$$\vec{W}_0^T \vec{X}_i + b_0 = 1 \text{ za } d_i = +1 \quad (3.11)$$

$$\vec{W}_0^T \vec{X}_i + b_0 = -1 \text{ za } d_i = -1 \quad (3.12)$$

kažemo da su potporni vektori (slika 5). Potporni vektori su one točke koje leže najbliže decizijskoj hiperravnini i zato se najteže klasificiraju. Zbog toga one imaju izravan utjecaj na optimalni položaj decizijske hiperravnine. Potporni vektor $\vec{X}^{(s)}$:

$$g(\vec{X}^{(s)}) = \vec{W}^T \vec{X}^{(s)} + b_0 = \pm 1 \text{ za } d_i = \pm 1 \quad (3.13)$$

Algebarska udaljenost potpornog vektora $\vec{X}^{(s)}$ od optimalne hiperravnine je

$$r = \frac{|g(\vec{X}^{(s)})|}{\|\vec{W}_0\|} \quad (3.14)$$

$$r = \begin{cases} \frac{1}{\|w_0\|} \dots \text{ako} \dots \text{je} \dots d^{(s)} = +1 \\ -\frac{1}{\|w_0\|} \dots \text{ako} \dots \text{je} \dots d^{(s)} = -1 \end{cases} \quad (3.15)$$

gdje predznak + označava da $\vec{X}^{(s)}$ leži na pozitivnoj strani optimalne hiperravnine, a predznak – pokazuje da je $\vec{X}^{(s)}$ na negativnoj strani optimalne hiperravnine. Optimalna vrijednost margine odvajanja između dva razreda koji definiraju skup za učenje je:

$$\rho = 2r = \frac{2}{\|\vec{W}_0\|} \quad (3.16)$$

iz čega slijedi da se maksimiziranje margine odvajanja temelji na minimizaciji norme vektora težinskih koeficijenata $\vec{W}_0$. Optimalna hiperravnina $\vec{W}_0^T \vec{X} + b_0 = 0$ je jedinstvena u tom smislu da vektor $\vec{W}_0$ daje maksimalnu separaciju između pozitivnih i negativnih uzoraka iz skupa za učenje. Cilj je razvoj djelotvorne procedure (uporabom skupa uzoraka za učenje) tako da se nađe optimalna hiperravnina uz zadovoljenje ograničenja $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1$ za $i = 1, 2, \dots, N$.

Formalno postavljen problem glasi:

- Zadan je skup uzoraka za učenje $\{(\vec{X}_i, d_i)\}_{i=1}^N$
- Nađi optimalnu vrijednost vektora težinskih koeficijenata $\vec{W}$ i pomak b tako da su zadovoljena ograničenja,

$$d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1 \text{ za } i = 1, 2, \dots, N \quad (3.17)$$

a pri tomu vektor težinskih koeficijenata $\vec{W}$ minimizira kriterijsku funkciju

$$J(\vec{W}) \equiv \frac{1}{2} \vec{W}^T \vec{W} = \|\vec{W}\|^2 \quad (3.18)$$

Optimizacijski problem rješava se metodom Lagrangeovih multiplikatora.

Primjer [7]:

Određivanje vezanih ekstrema funkcije $z = f(x, y)$ uz uvjet $\varphi(x, y) = 0$ svodi se na računanje slobodnih ekstrema Lagrangeove funkcije:

$$F = f(x, y) + \lambda \varphi(x, y)$$

$$\frac{\partial F}{\partial x} = 0; \frac{\partial F}{\partial y} = 0; \varphi(x, y) = 0$$

Iz tog se sustava jednadžbi određuju vrijednosti x, y i Lagrangeov multiplikator λ .

- Ako je $d^2 F < 0$ u izračunatoj točki, funkcija $z = f(x, y)$ ima maksimum
- Ako je $d^2 F > 0$ u izračunatoj točki, funkcija $z = f(x, y)$ ima minimum

Dakle imamo , $J(\vec{W}) \equiv \frac{1}{2} \vec{W}^T \vec{W}$ uz uvjet $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1$ za $i = 1, 2, \dots, N$.

Lagrangeova funkcija definira se kao:

$$L(\vec{W}, b, \lambda) = \frac{1}{2} \vec{W}^T \vec{W} - \sum_{i=1}^N \lambda_i [d_i (\vec{W}^T \vec{x}_i + b) - 1] \quad (3.19)$$

gdje su λ_i Lagrangeovi multiplikatori. Uvjeti koji moraju biti zadovoljeni su:

$$\frac{\partial}{\partial \vec{W}} L(\vec{W}, b, \lambda) = 0 \quad (3.20)$$

$$\frac{\partial}{\partial b} L(\vec{W}, b, \lambda) = 0 \quad (3.21)$$

$$\lambda \geq 0, \quad i = 1, 2, \dots, N \quad (3.22)$$

$$\lambda_i [d_i \cdot (\vec{W}^T \vec{X}_i + b) - 1] = 0, \quad i = 1, 2, \dots, N \quad (3.23)$$

Deriviranje $\frac{\partial}{\partial \vec{W}} L(\vec{W}, b, \lambda) = \vec{W} - \sum_{i=1}^N \lambda_i d_i \vec{X}_i = 0$ daje rezultat:

$$\vec{W} = \sum_{i=1}^N \lambda_i d_i \vec{X}_i \quad (3.24)$$

Vektor $\vec{W}$ je optimalno rješenje!

Deriviranje $\frac{\partial}{\partial b} L(\vec{W}, b, \lambda) = -\sum_{i=1}^N \lambda_i d_i = 0$ daje rezultat:

$$\sum_{i=1}^N \lambda_i d_i = 0 \quad (3.25)$$

Nakon postavljanja svih svojstava optimalne hiperravnine stroja s potpornim vektorima, slijedeći korak je računanje svih umiješanih parametara. Izračunavanje često nije jednostavan zadatak te postoje brojni algoritmi upravo u tu svrhu [6]. U obzir se uzima Lagrangeova dualnost (dualni problem) i problem se postavlja kao maksimizirati

$$L(\vec{W}, b, \lambda) \quad (3.26)$$

uz uvjet

$$\vec{W} = \sum_{i=1}^N \lambda_i d_i \vec{X}_i \quad (3.27)$$

$$\sum_{i=1}^N \lambda_i d_i = 0 \quad (3.28)$$

$$\lambda_i \geq 0, \quad i = 1, 2, \dots, N \quad (3.29)$$

Uvrštavanjem (3.27) i (3.28) u (3.26) i sređivanjem izraza, dobiva se slijedeći optimizacijski zadatak

$$\max_{\lambda} \left(\sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j d_i \cdot d_j, \vec{X}_i^T \vec{X}_j \right) \quad (3.30)$$

$$\text{Uz uvjet} \quad \sum_{i=1}^N \lambda_i d_i = 0 \quad (3.31)$$

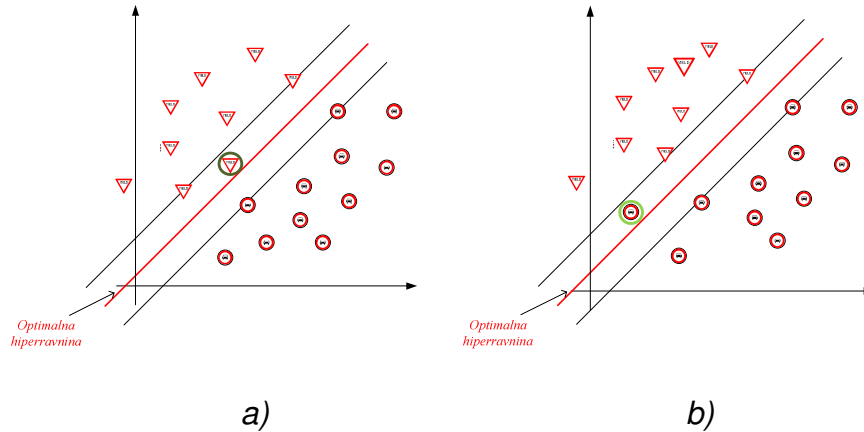
$$\lambda \geq 0 \quad (3.32)$$

Jednom kad se izračunaju optimalni Lagrangeovi multiplikatori, maksimizacijom (3.30), optimalna se hiperravnina računa preko $\vec{W} = \sum_{i=1}^N \lambda_i d_i \vec{X}_i$ gdje su λ_i optimalni Lagrangeovi multiplikatori, a b_0 preko jednog od uvjeta $\lambda_i [d_i \cdot (\vec{W}^T \vec{X}_i + b) - 1] = 0$, $i = 1, 2, \dots, N$ za $\lambda_i \neq 0$.

3.2. Neodvojivi razredi

Sve gore navedeno vrijedi za slučaj kada su razredi linearno odvojivi. Teži slučaj stroja s potpornim vektorima nastupa kod linearno neodvojivih uzoraka. Za zadani skup uzoraka za učenje nije moguće konstruirati hiperravninu koja razdvaja uzorke bez pogreške. Zbog toga želimo naći optimalnu hiperravninu koja minimizira vjerojatnost klasifikacijske pogreške. Margina razdvajanja naziva se mekom ako za uzorak $(\vec{X}_i, d_i)$ ne vrijedi $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1$, $i = 1, 2, \dots, N$. Povreda uvjeta $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1$ može nastupiti na dva načina (slika 7):

- i) uzorak $(\vec{X}_i, d_i)$ pada unutar područja odvajanja, ali na „pravu“ stranu decizijske hiperravnine (na slici a)
- ii) uzorak $(\vec{X}_i, d_i)$ pada na „krivu“ stranu decizijske hiperravnine (na slici b)



Slika 7 Prikaz slučajeve povrede uvjeta $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1$

Margina je definirana kao udaljenost između para paralelnih hiperravnina opisanih s $\vec{W}^T \vec{X}_i + b = \pm 1$. Uzorci za učenje pripadaju jednoj od tri kategorije uzoraka:

- i) vektori koji padaju izvan pojasa i koju su ispravno klasificirani (vektori koji zadovoljavaju $d_i \cdot (\vec{W}^T \vec{X}_i + b) \geq 1, i = 1, 2, \dots, N$)
- ii) vektori koji padaju unutar pojasa (margine), ali su pravilno razvrstani (slika a) (vektori za koje vrijedi: $0 \leq d_i \cdot (\vec{W}^T \vec{X}_i + b) < 1$)
- iii) vektori s pogrešnom klasifikacijom (slika b) (vektori za koje vrijedi $d_i \cdot (\vec{W}^T \vec{X}_i + b) < 0$)

Sva tri slučaja mogu se objediniti u jedinstveni oblik:

$$d_i (\vec{W}^T \vec{X}_i + b) \geq 1 - \xi_i \quad (3.3)$$

- i) slučaj: odgovaraju uzorci za koje je $\xi_i = 0$
- ii) slučaj: odgovaraju uzorci za koje je $0 < \xi_i \leq 1$
- iii) slučaj: odgovaraju uzorci za koje je $\xi_i > 1$

Varijable ξ_i se nazivaju „labave“ varijable (eng. *slack variables*), a predstavljaju mjeru odstupanja vektora od idealnog uvjeta separabilnosti uzoraka. Za $0 < \xi_i \leq 1$ uzorak se nalazi unutar područja odvajanja, ali s prave strane decizijske plohe. Za $\xi_i > 1$ uzorak pada na krivu stranu decizijske ravnine.

Potporni vektori su oni posebni uzorci koji zadovoljavaju jednadžbu $d_i(\vec{W}^T \vec{X}_i + b) \geq 1 - \xi_i$, $i = 1, 2, \dots, N$ čak i za slučaj $\xi_i > 1$.

Zadatak je naći decizijsku funkciju (hiperravninu razdvajanja) za koju je klasifikacijska pogreška minimalna. Drugačije interpretirano, učiniti marginu što je moguće većom, ali uz uvjet da je broj uzoraka sa $\xi_i > 1$ što je moguće manji.

Matematička interpretacija: minimizirati funkciju koštanja:

$$J(\vec{W}, b, \vec{\xi}) \equiv \frac{1}{2} \|\vec{W}\|^2 + C \sum_{i=1}^N I(\xi_i) \quad (3.34)$$

gdje je $\vec{\xi}$ vektor s komponentama ξ_i

$$I(\xi) = \begin{cases} 1 & \xi_i > 0 \\ 0 & \xi = 0 \end{cases} \quad (3.35)$$

Parametar C je pozitivna konstanta kojom se upravlja relativan utjecaj $\frac{1}{2} \|\vec{W}\|^2$ i

$\sum_{i=1}^N I(\xi_i)$. Dva su moguća načina izbora C :

- i) eksperimentalno preko standardne uporabe uzoraka za učenje i validaciju
- ii) C se određuje analitički procjenom VC (Vapnik-Chervonenkis) dimenzije koja je mjera ekspresivnosti porodice klasifikacijskih funkcija koje su ostvarene strojem za učenje dimenzija $h \leq \min \left\{ \left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0 \right\} + 1$, D -
dijametar najmanje kugle koja sadrži sve ulazne vektore $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N$, ρ -
margina = $\frac{2}{\|\vec{W}\|}$ i m_0 je dimenzionalnost ulaznog prostora.

Optimizacija izraza $J(\vec{W}, b, \vec{\xi}) \equiv \frac{1}{2} \|\vec{W}\|^2 + C \sum_{i=1}^N \xi_i$ teška je jer je uključena i funkcija $I(\cdot)$ koja je diskontinuirana. Uobičajen pristup je izabrati funkciju koja je „bliska“ funkciji koštanja:

$$J(\vec{W}, b, \xi) = \frac{1}{2} \|\vec{W}\|^2 + C \sum_{i=1}^N \xi_i \quad (3.36)$$

uz uvjet $d_i(\vec{W}^T \vec{X}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$ (3.37)

$$\xi_i > 0, \quad i = 1, 2, \dots, N \quad (3.38)$$

Odgovarajući Lagrange dan je kao

$$L(\vec{W}, b, \xi, \vec{\lambda}, \vec{\mu}) = \frac{1}{2} \|\vec{W}\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \mu_i \xi_i - \sum_{i=1}^N \lambda_i [d_i(\vec{W}^T \vec{X}_i + b) - 1 + \xi_i] \quad (3.39)$$

Odgovarajući Karush-Kuhn-Tucker uvjeti su

$$\frac{\partial L}{\partial \vec{W}} = 0 \quad \text{ili} \quad \vec{W} = \sum_{i=1}^N \lambda_i d_i \vec{X}_i \quad (3.40)$$

$$\frac{\partial L}{\partial b} = 0 \quad \text{ili} \quad \sum_{i=1}^N \lambda_i d_i = 0 \quad (3.41)$$

Pridružena Wolfeova dualna prezentacija postaje

Maksimizirati $L(\vec{W}, b, \xi, \vec{\lambda}, \vec{\mu})$ (3.42)

uz uvjet $\vec{W} = \sum_{i=1}^N \lambda_i d_i \vec{X}_i$ (3.43)

$$\sum_{i=1}^N \lambda_i d_i = 0 \quad (3.44)$$

$$C - \mu_i - \lambda_i = 0, \quad i = 1, 2, \dots, N \quad (3.45)$$

$$\lambda_i \geq 0, \quad \mu_i \geq 0, \quad i = 1, 2, \dots, N \quad (3.46)$$

Uvrštavanjem gornjih eksplicitnih ograničenja u Lagrange dobiva se

$$\max_{\lambda} \left(\sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j d_i d_j \vec{X}_i^T \vec{X}_j \right) \quad (3.47)$$

uz uvjet $0 \leq \lambda_i \leq C, \quad i = 1, 2, \dots, N$ (3.48)

$$\sum_{i=1}^N \lambda_i d_i = 0 \quad (3.49)$$

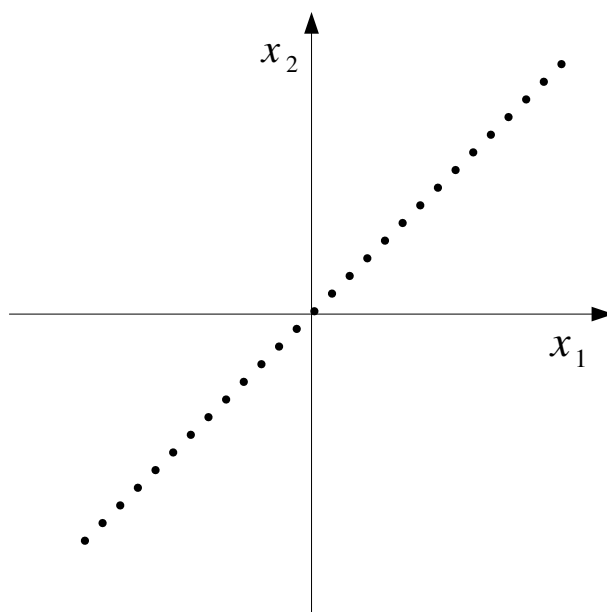
Lagrangeovi multiplikatori koji odgovaraju točkama (uzorcima) koji leže unutar margine ili na krivoj strani klasifikatora, tj. $\xi_i > 1$ su jednaki maksimalnoj dopuštenoj vrijednosti C . Ako se promotri izraz $\max_{\lambda} \left(\sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j d_i d_j \vec{X}_i^T \vec{X}_j \right)$ vidi se da niti labave varijable ξ_i niti Lagrangeovi multiplikatori pridruženi labavim varijablama μ_i ne ulaze eksplicitno u problem. Oni su neizravno prisutni kroz C odnosno kroz uvjet $0 \leq \lambda_i \leq C$. Razlika u odnosu na linearno separabilni slučaj je što Lagrangeovi multiplikatori moraju biti ograničeni na C ($0 \leq \lambda_i \leq C$). Z linearno separabilni slučaj $C \rightarrow \infty$ ($\lambda_i \geq 0$). Osim navedene modifikacije postupak računanja optimalnih vrijednosti vektora $\vec{W}$ i pomaka b je jednak kao i kod separabilnog slučaja. Rješenje je $\vec{W}_0 = \sum_{i=1}^{N_s} \lambda_{0,i} d_i \vec{X}_i$ gdje je N_s broj vektora koji su potporni. Pomak b_0 se može odrediti uzimajući bilo koji $(\vec{X}_i, d_i)$ iz skupa za učenje za koji vrijedi $0 \leq \lambda_{0,i} \leq C$.

4. Analiza svojstvenih komponenti

PCA (eng. *Principal Component Analysis*) ili analiza svojstvenih komponenti je statistički postupak za reduciranje dimenzionalnosti podataka [8]. 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 .

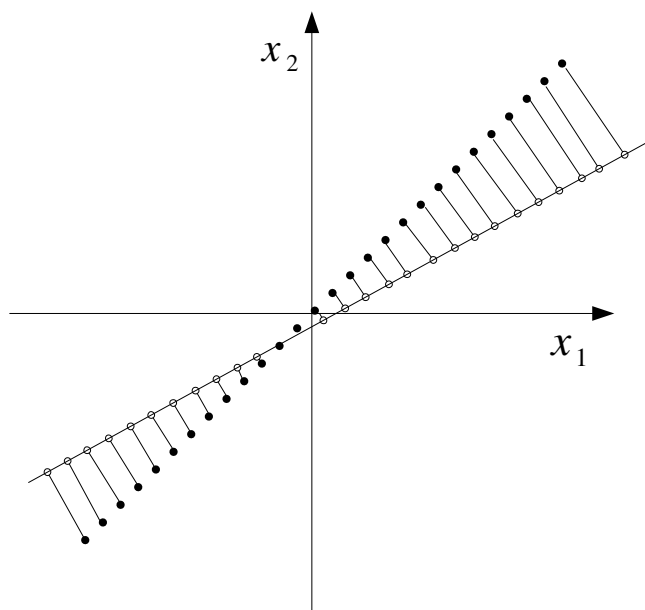
Primjena metode PCA nad skupom slika omogućava dobivanje potprostora svih mogućih slika koje sadrže znakove. Ideja je da se pronađe takav potprostor zadane niže dimenzionalnosti koji najbolje prikazuje pojedini razred uzoraka u smislu da pogreška između originalnog uzorka i projekcije uzorka u taj potprostor bude minimalna. U nastavku slijedi detaljnije objašnjenje navedene ideje [9].

Svaku sliku znaka možemo zamisliti kao vektor slikovnih elemenata. Primjer 2D vektora dan je slikom 8.



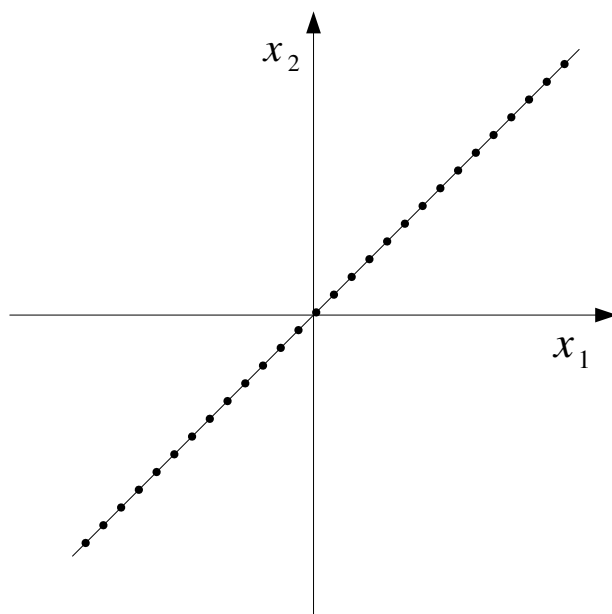
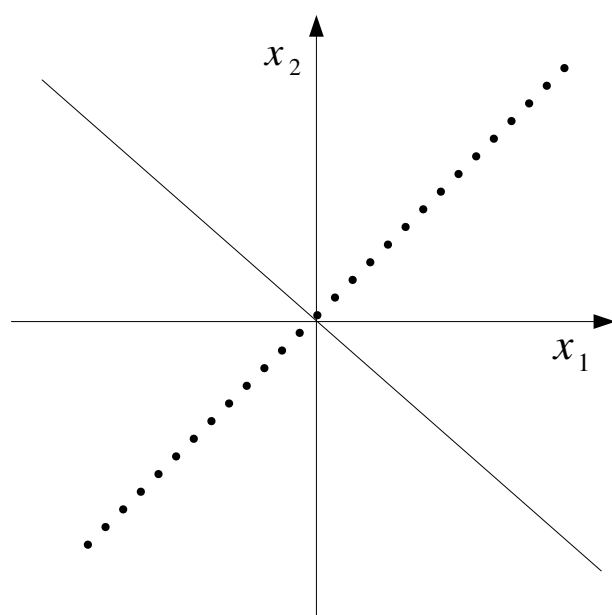
Slika 8 Primjer 2D vektora

Pretpostavimo da podatke sa slike 8 želimo prikazati u 1D prostoru. Da bismo to postigli odaberemo pravac u prostoru na kojeg ortogonalno projiciramo svaku točku. Slika 9 prikazuje ortogonalnu projekciju 2D vektora sa slike 8.



Slika 9 Ortogonalna projekcija

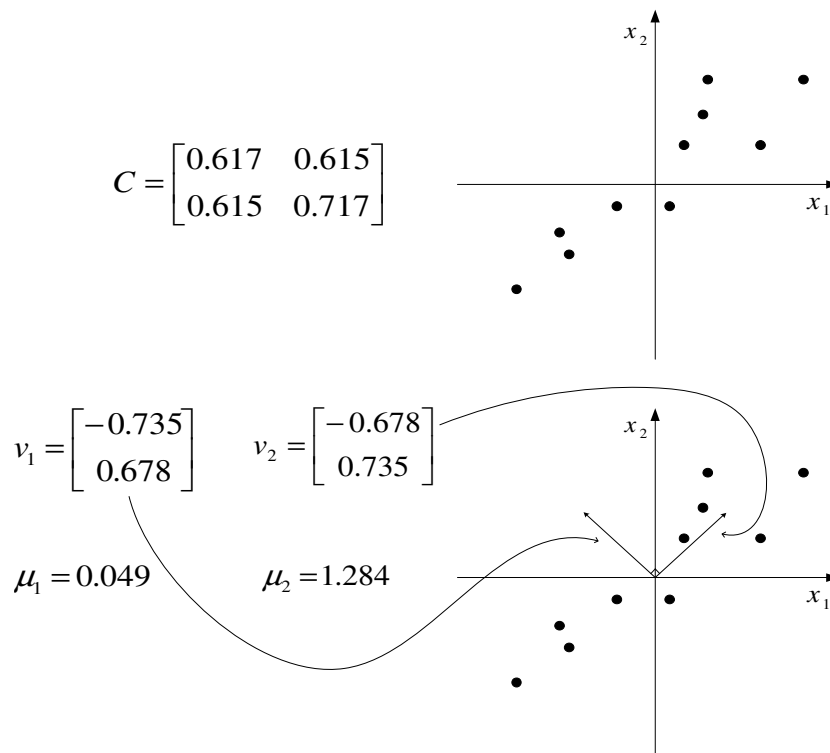
Neki pravci će bolje prikazivati podatke, neki lošije. Pravci koji dobro prikazuju podatke su oni koji pri ortogonalnoj projekciji točaka na pravac daju malo odstupanje originalnih podatka od projiciranih (slika 10). Loš prikaz podataka predstavljaju pravci za koje je odstupanje originalnih podataka od projiciranih podataka toliko da nereprezentativno prikazuju podatak, gube se bitne informacije (Slika 11).

*Slika 10 Dobar prikaz podataka**Slika 11 Loš prikaz podataka*

Umjesto pravca na koji projiciramo uzorke, bolje je pronaći vektor koji pokazuje smjer u kojem podaci variraju. Da bi se to odredilo, potrebno je promatrati suodnose uzoraka. Tu dolazimo do kovarijacijske matrice. 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 [10].

Pronalaženje najvećih svojstvenih vrijednosti i svojstvenih vektora kovarijacijske matrice predstavlja glavnu ideju PCA ili analize svojstvenih komponenti [8].

Na slici 12 možemo vidjeti dvije varijable x_1 i x_2 te njihovu kovarijacijsku matricu C [9]. Svojstvene vrijednosti te matrice dane su vektorima v_1 i v_2 , kao i njihove svojstvene vrijednosti μ_1 i μ_2 . Vidimo da vektor v_2 s većom svojstvenom vrijednošću μ_2 prikazuje smjer u kojem podaci više variraju.



Slika 12 Kovarijanca varijabli

4.1. Analiza svojstvenih komponenti na skupu znakova

Svaka slika znaka sastoji se od n slikovnih elemenata (piksela). Slika se može prikazati u jednodimenzionalnom obliku kao stupčasti vektor s n elemenata. Takvi vektori označeni su s $\Gamma_1 \dots \Gamma_N$ (N je broj slika znakova) za svaku normaliziranu sliku iz baze te organizirani u matricu S dimenzija $n \times N$ [11].

$$S = [\Gamma_1 \dots \Gamma_N] \quad (4.1)$$

Slijedeći korak je pronalaženje prosječne slike znaka Ψ .

$$\Psi = \frac{1}{N} \sum_{i=1}^N \Gamma_i \quad (4.2)$$

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

$$\Phi_i = \Gamma_i - \Psi, i = 1, 2, \dots, N \quad (4.3)$$

Od tako dobivenih vektora tvori se matrica A kao

$$A = [\phi_1, \phi_2 \dots \phi_N] \quad (4.4)$$

Matrica A ima N stupaca i n redaka, $n \times N$. 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

$$C = AA^T \quad (4.5)$$

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'

$$C' = A^T A \quad (4.6)$$

Do navedene pomoćne matrice C' došlo se na slijedeći način. Razmotri li se na trenutak, umjesto svojstvenih vektora matrice $C = AA^T$ svojstvene vektore matrice $A^T A$

$$A^T A v_i = \lambda_i v_i, i = 1, 2, \dots, N \quad (4.7)$$

Dimenzije ove matrice su $N \times N$. Očito je da će izračunavanje svojstvenih vektora ove matrice ići daleko brže nego izračunavanje svojstvenih vektora matrice C dimenzija $n \times n$ gdje je $n \gg N$. Sada množenjem jednadžbe (4.7) s A s lijeve strane dobiva se

$$AA^T A v_i = \lambda_i A v_i, i = 1, 2, \dots, N \quad (4.8)$$

Usporedi li se dobivena jednadžba s definicijom svojstvenih vektora i svojstvenih vrijednosti može se 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.

Označe li se svojstvene vrijednosti i normalizirane svojstvene vektore matrice $A^T A$ kao λ'_i i v'_i tada N najvećih svojstvenih vrijednosti λ_i i odgovarajuće normalizirane svojstvene vektore v_i matrice $C = AA^T$ mogu se dobiti kao

$$\lambda_i = \lambda'_i, i = 1, 2, \dots, N \quad (4.9)$$

$$v_i = \frac{1}{\sqrt{\lambda_i}} A v'_i, i = 1, 2, \dots, N \quad (4.10)$$

Svojstveni znakovi predstavljaju svojstvenu bazu prostora u kojem će se prikazivati i uspoređivati slike znakova. Svaka slika Φ iz svojstvene baze projicirana je na k -tu komponentu prostora svojstvenih znakova kako slijedi:

$$\omega_k = v_k^T \Phi, k = 1, 2, \dots, m \quad (4.11)$$

Oznaka ω_k je projekcija podataka na k -tu dimenziju, 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. Projekcija slike Φ na prostor svojstvenih znakova dobiva se kao vektor Ω pojedinačnih projekcija ω_i .

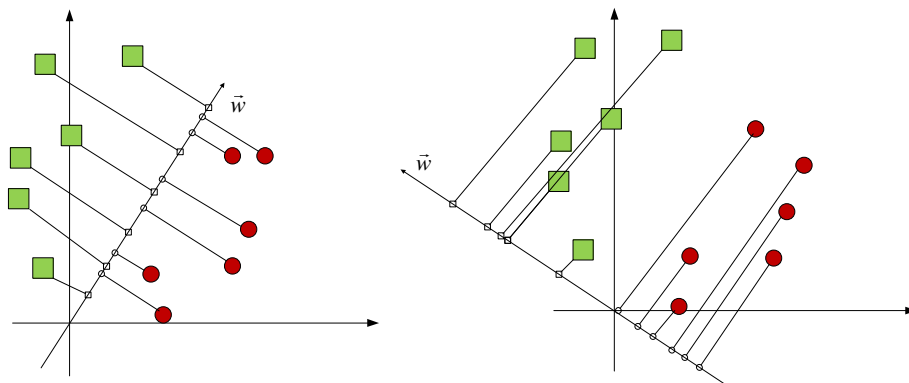
$$\Omega^T = [\omega_1, \omega_2, \dots, \omega_m] \quad (4.12)$$

Da bi se odredilo kojoj klasi pripada ulazna slika znaka može se koristiti klasifikator 1-NN [12]. To je specijalan slučaj metode k -najbližih susjeda. Neklasificirani znak iz skupa za testiranje pripada onoj klasi znaka iz skupa za učenje kojoj pripada najbliži susjed. Najbliži susjed je onaj koji ima najmanju euklidsku udaljenost do neklasificiranog uzorka.

$$\mathcal{E}_k = \|\Omega - \Omega_k\| \quad (4.13)$$

5. Fisherova linearna diskriminantna analiza

Fisherova linearna diskriminantna analiza jedan je od glavnih pristupa linearnoj klasifikaciji [6], [13]. Osnovna ideja Fisherove analize je d -dimenzionalan vektor značajki reducirati na jednu dimenziju (u slučaju za dva razreda) i tada ga upotrijebiti za klasifikaciju. Potrebno je pronaći orijentaciju pravca na koji se projiciraju d -dimenzionalni uzorci $\vec{x}_i$ pri čemu je $i=1,2,\dots,n$, ali tako da su projicirani uzorci budi što više odvojivi. Upravo to predstavlja cilj klasične diskriminantne analize.

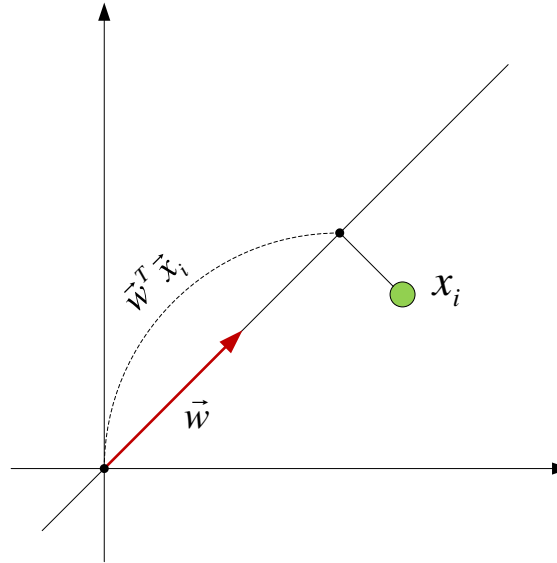


Slika 13 Projekcije uzoraka na dva različita pravca. Desna slika prikazuje bolju razdvojenost projekcija crvenih i zelenih točaka nego lijeva slika

Pretpostavimo da imamo skup od n d -dimenzionalnih uzoraka $x_1, \dots, x_n$ od čega n_1 uzoraka čini razred D_1 (označeni s w_1) i n_2 uzoraka razred D_2 (označeni s w_2). Tvorimo linearnu kombinaciju komponenti $\vec{x}$:

$$y_i = \vec{w}^T \vec{x}_i, i = 1, 2, \dots, n \quad (5.1)$$

Dobivamo odgovarajući skup od n uzoraka $y_1, \dots, y_n$ koji su podijeljeni u podrazrede Y_1 i Y_2 . Ako je $\|\vec{w}\|=1$, svaki y_i je projekcija odgovarajućeg vektora $\vec{x}_i$ na pravac (u smjeru $\vec{w}$, slika 14). Norma vektora $\vec{w}$ nema posebno značenje, ona samo skalira y_i , bitan je smjer vektora $\vec{w}$.



Slika 14 Projekcija uzorka $\vec{x}_i$ na pravac u smjeru $\vec{w}$

Tražimo $\vec{w}$ za koji se projekcije uzoraka mogu dobro odvojiti. Mjera odvojivosti između projiciranih uzoraka je razlika srednjih vrijednosti projiciranih uzoraka. Srednja vrijednost zadanih uzoraka definirana je izrazom:

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x \quad (5.2)$$

Srednja vrijednost projiciranih uzoraka:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y \quad (5.3)$$

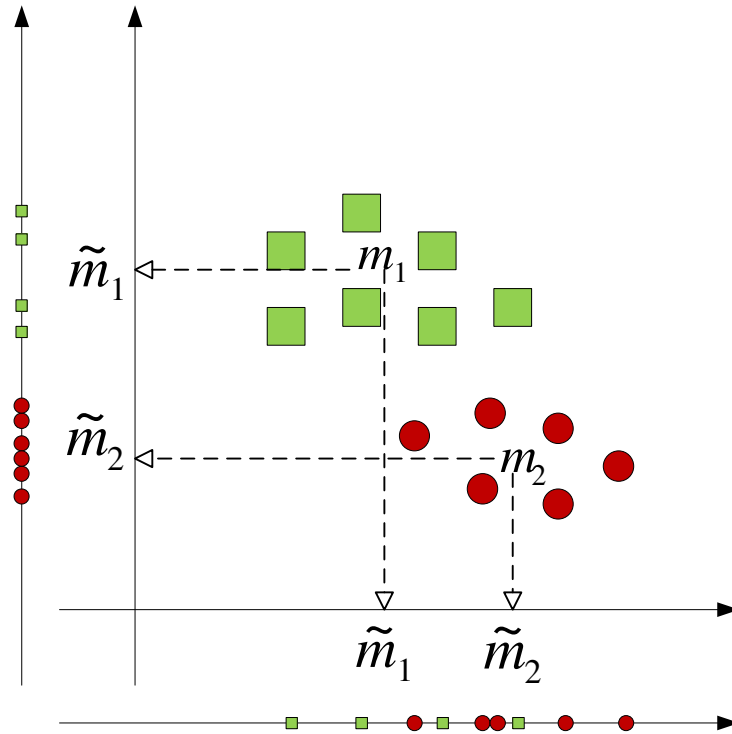
Uvrštavanje (5.1) i (5.2) u (5.3) dobivamo relaciju:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} w^T x = w^T m_i. \quad (5.4)$$

Slijedi da je razlika između srednjih vrijednosti projiciranih uzoraka:

$$|\tilde{m}_1 - \tilde{m}_2| = |w^T (m_1 - m_2)| \quad (5.5)$$

Da li je udaljenost srednjih vrijednosti projiciranih uzoraka „dobra mjera“ da se odijele projicirani uzorci različitih razreda?



Slika 15 Utjecaj udaljenosti srednjih projiciranih uzoraka na klasifikaciju uzoraka

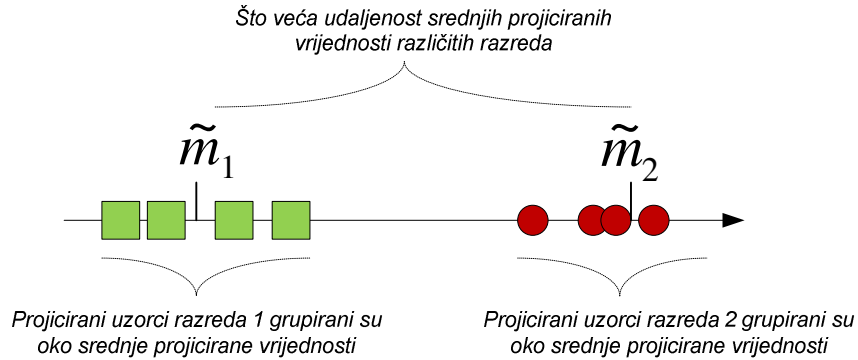
Na slici 15 se vidi da vertikalna os bolje razdvaja uzorke unatoč manjoj udaljenosti srednjih projiciranih uzoraka naspram udaljenosti srednjih projiciranih uzoraka na horizontalnu os. Razlog tomu jest činjenica da definirana mjera odvojivosti ne uračunava varijancu unutar razreda zadanih uzoraka. Kako bi se to izbjeglo, definira se mjera raspršenosti za projicirane uzorke:

$$\tilde{s}_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2. \quad (5.6)$$

Primijetimo da $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$ je procjena varijance projiciranih uzoraka, a $\tilde{s}_1^2 + \tilde{s}_2^2$ je ukupna mjera raspršenosti projiciranih uzoraka unutar razreda. Fisherova linearna diskriminanta određuje da linearna funkcija $\vec{w}^T \vec{x}_i$ za koju je kriterijska funkcija $J(\vec{w})$ maksimalna vodi najboljem razdvajanju između projiciranih skupova.

$$J(\vec{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad (5.7)$$

Kako bi gore navedeni izraz bio maksimalan, potrebno je imati što veću razliku srednjih vrijednosti i što manju raspršenost unutar projiciranih uzoraka istog razreda (slika 16).



Slika 16 Da bi razdvajanje projiciranih uzoraka bilo najbolje, potrebno je imati što veću razliku srednjih vrijednosti različitih razreda i što manju raspršenost unutar projiciranih uzoraka istog razreda.

Potrebno je izraziti kriterijsku funkciju $J(\vec{w})$ kao eksplicitnu funkciju od $\vec{w}$. Definirajmo matricu S_i :

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T \quad (5.8)$$

I matricu S_w :

$$S_w = S_1 + S_2 \quad (5.9)$$

Uvrsti se (5.1) i (5.4) u izraz (5.6):

$$\begin{aligned} \tilde{s}_i^2 &= \sum_{x \in D_i} (w^T x - w^T m_i)^2 \\ &= \sum_{x \in D_i} w^T (x - m_i)(x - m_i)^T w \\ &= w^T S_i w \end{aligned} \quad (5.10)$$

Ukupna mjera raspršenosti je:

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_w w \quad (5.11)$$

Slično, razlika srednjih vrijednosti projiciranih uzoraka je:

$$\begin{aligned}
 (\tilde{m}_1^2 - \tilde{m}_2^2)^2 &= (w^T m_1 - w^T m_2)^2 \\
 &= w^T (m_1 - m_2)(m_1 - m_2)^T w \\
 &= w^T S_B w
 \end{aligned} \tag{5.12}$$

Gdje je:

$$S_B = (m_1 - m_2)(m_1 - m_2)^T \tag{5.13}$$

S_w je matrica raspršenosti unutar razreda, a S_B matrica raspršenosti između razreda. Rang matrice S_B za slučaj dva razreda je jedan.

Uvrštavanjem (5.11) i (5.12) u (5.7) dolazimo do kriterijske funkcije koja ovisi samo o parametru $\vec{w}$:

$$J(\vec{w}) = \frac{\vec{w}^T S_B \vec{w}}{\vec{w}^T S_w \vec{w}} \tag{5.15}$$

Preostalo je pronaći maksimum funkcije $J(\vec{w})$. Do njega se dolazi deriviranjem funkcije i izjednačavanjem sa nulom:

$$\begin{aligned}
 \frac{\partial J(\vec{w})}{\partial w} &= 0 \\
 \frac{\partial J(\vec{w})}{\partial w} &= \frac{\partial}{\partial \vec{w}} \left(\frac{\vec{w}^T S_B \vec{w}}{\vec{w}^T S_w \vec{w}} \right) = 0 \\
 \frac{(\vec{w}^T S_w \vec{w}) \frac{\partial}{\partial w} (\vec{w}^T S_B \vec{w}) - (\vec{w}^T S_B \vec{w}) \frac{\partial}{\partial w} (\vec{w}^T S_w \vec{w})}{(\vec{w}^T S_w \vec{w})^2} &= 0 \\
 (\vec{w}^T S_w \vec{w}) \cdot 2S_B \vec{w} - (\vec{w}^T S_B \vec{w}) \cdot 2S_w \vec{w} &= 0 \\
 S_w \vec{w} (\vec{w}^T S_B \vec{w}) (\vec{w}^T S_w \vec{w})^{-1} &= S_B \vec{w} \\
 \lambda &= (\vec{w}^T S_B \vec{w}) (\vec{w}^T S_w \vec{w})^{-1}
 \end{aligned} \tag{5.16}$$

$$\lambda S_w w = S_B w \tag{5.17}$$

Dobiveni izraz $\lambda S_w w = S_B w$ naziva se generalizirani problem svojstvenih vektora. Međutim, nije uvijek potrebno pristupiti ovom problemu tako se traže svojstvene vrijednosti i svojstveni vektori za $S_w^{-1} S_B$ zato što je $S_B \vec{w}$ uvijek usmjeren kao $\vec{m}_1 - \vec{m}_2$. Budući da nas faktor skaliranja za vektor $\vec{w}$ ne zanima (zanima nas samo smjer), može se napisati rješenje za w :

$$w = S_w^{-1}(m_1 - m_2) \quad (5.18)$$

Dobiveni izraz za vektor $\vec{w}$ određuje linearnu funkciju koja maksimizira omjer između udaljenosti srednjih vrijednosti projiciranih uzoraka i raspršenja unutar razreda. Drugim riječima, pronađen je smjer vektora $\vec{w}$ za kojeg se može vršiti najbolja klasifikacija.

5.1. FLD za više razreda

Fisherova linearna diskriminanta se može poopćiti odnosno primijeniti na problem više razreda [6], [12]. Imamo c razreda dimenzionalnosti $D_1, \dots, D_c$ iliti $w_1, \dots, w_c$. Generalizirana FLD $c-1$ diskriminantnih funkcija tj. projekcija d -dimenzionalnog prostora u $(c-1)$ -dimenzionalni prostor $d \geq c$.

Generalizirana matrica raspršenosti unutar razreda je:

$$S_w = \sum_{i=1}^c S_i \quad (5.19)$$

gdje je

$$S_i = \sum_{x \in D_i} (\vec{x} - \vec{m}_i)(\vec{x} - \vec{m}_i)^T \quad (5.20)$$

$$\vec{m}_i = \frac{1}{n_i} \sum_{x \in D_i} \vec{x} \quad (5.21)$$

Matrica raspršenosti između razreda S_B ne dobiva se tako očigledno. Ukupan vektor srednje vrijednosti $\vec{m}$ jest:

$$\vec{m} = \frac{1}{n} \sum_x \vec{x} = \frac{1}{n} \sum_{i=1}^c n_i \vec{m}_i \quad (5.22)$$

gdje je n_i broj uzoraka u razredu $D_i(w_i)$, a $\vec{m}_i$ vektor srednjih vrijednosti w_i . Ukupna matrica raspršenosti S_T je:

$$\begin{aligned}
 S_T &= \sum_{\vec{x}} (\vec{x} - \vec{m})(\vec{x} - \vec{m})^T \\
 S_T &= \sum_{i=1}^c \sum_{\vec{x} \in D_i} (\vec{x} - \vec{m}_i + \vec{m}_i - \vec{m})(\vec{x} - \vec{m}_i + \vec{m}_i - \vec{m})^T \\
 S_T &= \sum_{i=1}^c \sum_{\vec{x} \in D_i} (\vec{x} - \vec{m}_i)(\vec{x} - \vec{m})^T + \sum_{i=1}^c \sum_{\vec{x} \in D_i} (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T \\
 &+ \sum_{i=1}^c \sum_{\vec{x} \in D_i} (\vec{m}_i - \vec{m})(\vec{x} - \vec{m})^T + \sum_{i=1}^c \sum_{\vec{x} \in D_i} (\vec{m}_i - \vec{m})^T (\vec{x} - \vec{m}) \\
 S_T &= S_w + \sum_{i=1}^c n_i (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T
 \end{aligned} \tag{5.23}$$

Drugi član $\sum_{i=1}^c n_i (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T$ je poopćena matrica raspršenosti između razreda S_B :

$$S_B = \sum_{i=1}^c n_i (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T \tag{5.24}$$

$$S_T = S_w + S_B \tag{5.25}$$

Projekcija iz d -dimenzionalnog prostora u $(c-1)$ -dimenzionalni prostor postiže se uporabom $(c-1)$ -diskriminantnih funkcija te je dana izrazom $y_i = \vec{w}_i \vec{x}$, $i=1,2,\dots,c-1$. Ako y_i promatramo kao komponentu vektora $\vec{y}$, težinske faktore $\vec{w}$ kao stupce matrice W ($d \times (c-1)$) tada je projekcija:

$$\vec{y} = W^T \vec{x} \tag{5.26}$$

Uzorci $\vec{x}_1, \dots, \vec{x}_n$ projiciraju se u odgovarajući skup uzoraka $\vec{y}_1, \dots, \vec{y}_n$ koji mogu biti opisani svojim srednjim uzorcima i matricama raspršenosti:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{\vec{y} \in Y_i} \vec{y} \tag{5.27}$$

$$\tilde{m} = \frac{1}{n_i} \sum_{i=1}^c n_i \tilde{m}_i \tag{5.28}$$

$$\tilde{S}_w = \sum_{i=1}^c \sum_{\vec{y} \in Y_i} (\vec{y} - \tilde{m}_i)(\vec{y} - \tilde{m}_i)^T \tag{5.29}$$

$$\tilde{S}_B = \sum_{i=1}^c n_i (\tilde{m}_i - \tilde{m})(\tilde{m}_i - \tilde{m})^T \quad (5.30)$$

$$\tilde{S}_w = W^T S_w W \quad (5.31)$$

$$\tilde{S}_B = W^T S_B W \quad (5.32)$$

Jednadžbe (5.31) i (5.32) pokazuju kako se matrice raspršenosti unutar i između razreda transformiraju projekcijom u nižedimenzionalni prostor. Tražimo transformacijsku matricu W koja maksimizira omjer raspršenosti između razreda s raspršenošću unutar razreda. Jednostavna skalarna mjera raspršenosti je determinanta matrice raspršenosti.

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_w|} = \frac{|W^T S_B W|}{|W^T S_w W|} \quad (5.33)$$

Determinanta je produkt svojstvenih vrijednosti matrice. Problem traženja (i nalaženja) pravokutne matrice W koja maksimizira $J(\cdot)$ je težak problem. Rješenje je slijedeće. Stupci optimalne matrice W su generalizirani svojstveni vektori koji odgovaraju najvećim svojstvenim vrijednostima u :

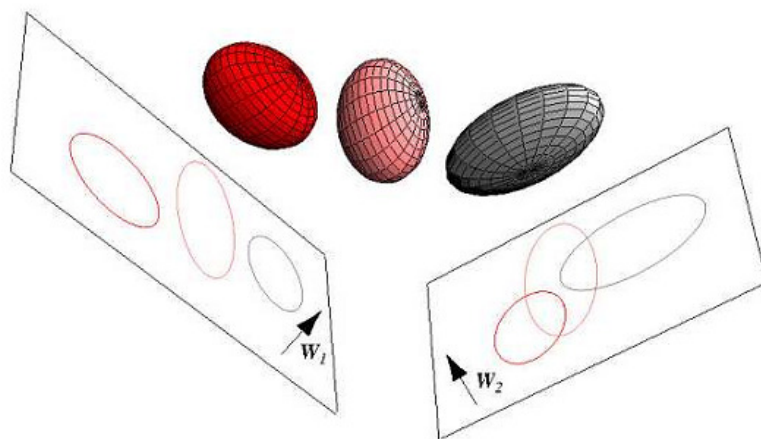
$$S_B \vec{w}_i = \lambda_i S_w \vec{w}_i \quad (5.34)$$

Ako je S_w nesingularna onda se problem može pretvoriti u konvencionalni problem svojstvenih vrijednosti [10]. Međutim, to zahtijeva računanje inverzne matrice S_w^{-1} ($S_w^{-1} S_B \vec{w} = \lambda \vec{w}$). Umjesto toga može se naći svojstvene vrijednosti kao korijene karakterističnog polinoma:

$$|S_B - \lambda_i S_w| = 0 \quad (5.35)$$

i zatim riješiti:

$$(S_B - \lambda_i S_w) \vec{w}_i = 0 \quad (5.36)$$



*Slika 17 Projekcija 3-D distribucije na 2-D potprostore koji su opisani
normaliziranim vektorima $\bar{w}_1$ i $\bar{w}_2$ [6]*

Na slici 17 prikazana je projekcija triju trodimenzionalnih distribucija na dvodimenzionalne podprostore opisane normalama $\bar{w}_1$ i $\bar{w}_2$. Ideja FLD je da se pronađe onaj podprostor za koji je su projicirani uzorci najbolje razdvojeni. U ovom primjeru taj podprostor je definiran normalom $\bar{w}_1$.

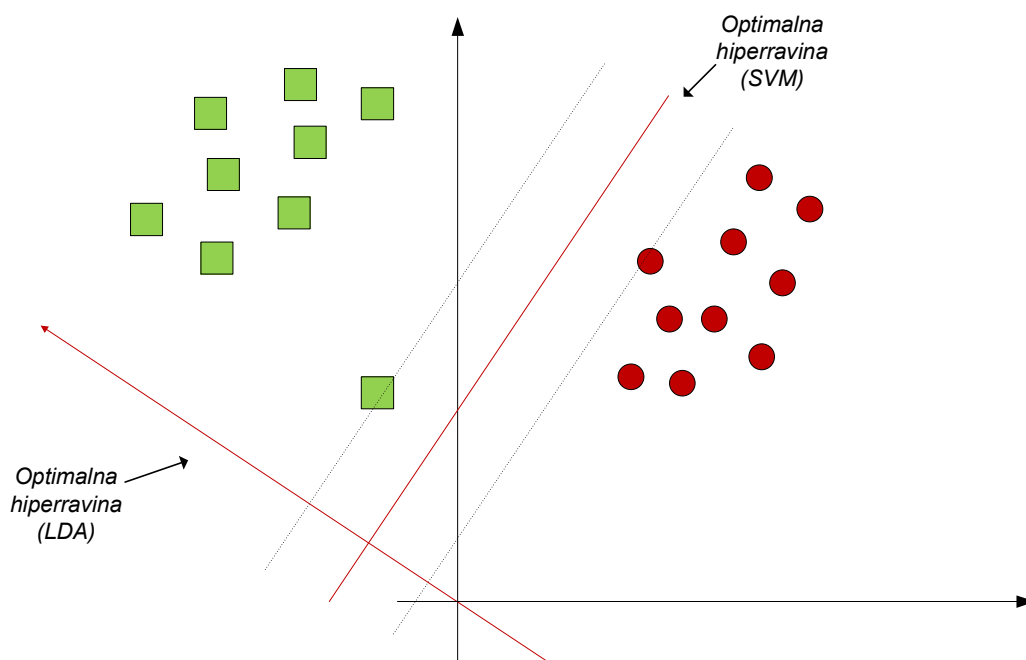
5.2. Usporedba FLD i SVM

Linearna diskriminatna analiza kao i stroj s potpornim vektorima pronalaze optimalne hiperravnine uzimajući u obzir različite kriterije.

FLD pronalazi optimalnu hiperravninu na način da maksimizira udaljenost između razreda, a minimizira udaljenost unutar razreda. Matrica raspršenosti koja se dobiva pri izvođenju metode FLD velikih je dimenzija u slučaju kada su dimenzije vektora značajki velike. Osim računalne zahtjevnosti pri radu s velikim matricama, problem predstavlja i to što se često dobiva singularna matrica (broj uzoraka manji od dimenzionalnosti uzorka). Pronalazak inverza matrice u tom slučaju nije moguć, pa se FLD ne koristi direktno, nego u kombinaciji s metodom PCA koja prethodno smanji dimenzije.

SVM predstavlja vrh klasifikacijskih algoritama. Cilj je pronaći onu hiperravninu za koju je margina odvajanja maksimalna. Optimalni položaj hiperravnine uzima u obzir potporne vektore, one točke koje leže najbliže decizijskoj hiperravnini i najteže se klasificiraju.

U većini slučajeva korištenje metode SVM daje bolje klasifikacijske rezultate od FLD [14]. Stroj s potpornim vektorima pokazao se stabilan i pri velikim dimenzijama. Ipak postoje slučajevi u kojima metoda FLD daje bolje rezultate od SVM-a. Takav primjer dan je slikom 18.



Slika 18 Usporedba FLD i SVM

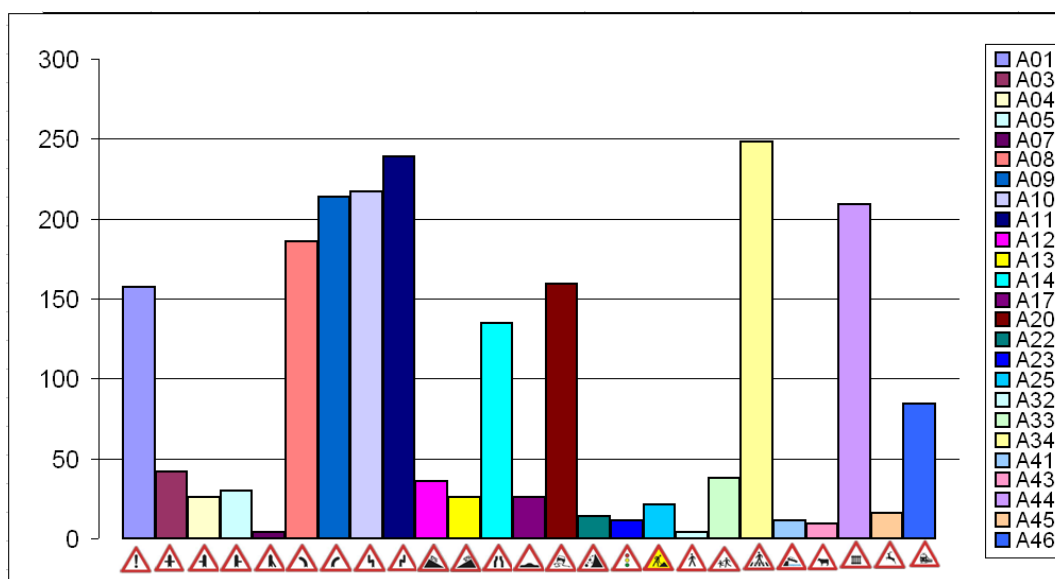
Na slici 18 dani su uzorci dviju klasa. Prva klasa (zeleni kvadratići) nije jednoliko raspršena, postoji uzorak koji se izdvaja iz grupe. Kako SVM pronalazi granicu na temelju potpornih vektora, može se primijetiti da je u ovom slučaju pronašao prestrogu granicu jer u obzir uzima samo lokalne karakteristike, a ne globalne karakteristike uzoraka. FLD naime uzima u obzir čitavo raspršenje uzoraka zbog čega nalazi bolju granicu. Može se zaključiti da je FLD robusnija metoda od SVM-a.

Optimum bi bio korištenje kombinacije klasifikatora. Dakle radilo bi se o proširenju stroja s potpornim vektorima na način da uzima u obzir globalna svojstva uzoraka kao što to čini linearna diskriminatna analiza [15] .

6. Evaluacija

6.1. Skup za učenje

Skup za učenje klasifikatora sastoji se od 2162 ručno označena isječka slike na kojima se nalaze znakovi. Radi se o skupu T2009 prikupljenom u radu [1]. U skupu je zastupljeno 25 od ukupno 50 razreda znakova upozorenja. Pri tome je broj uzoraka po razredu različit te se kreće od 4 do 248. Slika 19 prikazuje zastupljene razrede i razdiobu broja primjeraka po razredima.



Slika 19 Zastupljeni razredi znakova i broj primjeraka po razredima.

Slike iz skupa su pod utjecajem smetnje uzrokovane preplitanjem te se za njih provodi uklanjanje smetnje, na način opisan u poglavlju 2.1.2.

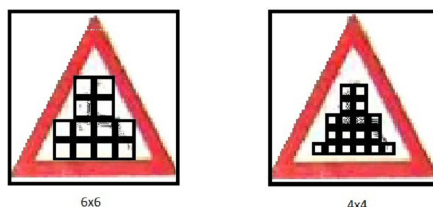
6.2. Odabir ulaznih značajki

Pri klasifikaciji znakova poželjno je odabrati ulazne značajke koje dobro opisuju razliku između različitih klasa znakova. Informacija o razlici između klasa se nalazi na području središta znaka.

6.2.1. Histogrami orijentacije gradijenata

Razlika između različitih klasa znakova je u oblicima koji se nalaze u središtu te se histogram orijentacije gradijenata (poglavlje 2.2) koji nosi informaciju o rubovima na središtu slike nameće kao logičan odabir ulaznih značajki.

Slike se skaliraju na veličinu 48x48 te se luminantna komponenta normalizira na raspon od 0 do 255. Slijedi filtriranje Gaussovom filtrom, a potom se izrađuju histogrami orijentacije gradijenata nad blokovima smještenim u središtu slike. Postoje dva skupa blokova, jedni veličine 4x4, te drugi veličine 6x6. Slika 20 prikazuje raspored blokova [16].



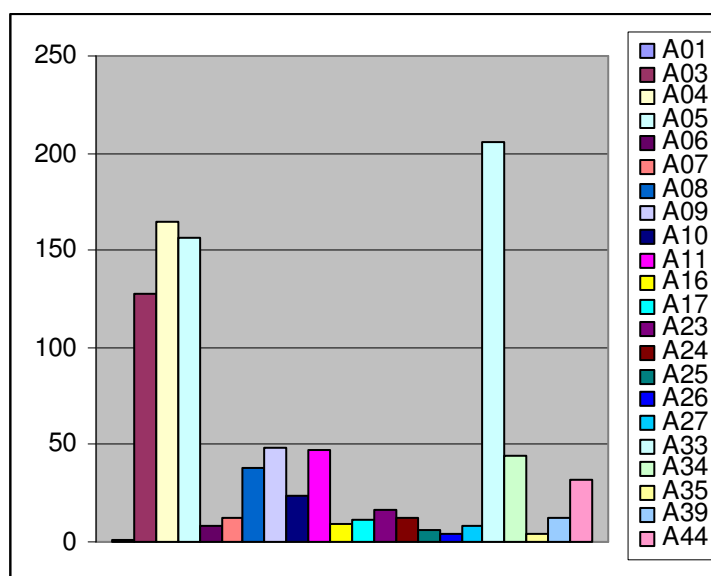
Slika 20 Raspored blokova

Histogrami nad blokovima veličine 6x6 se rade za četiri kuta i 180 stupnjeva, dok se histogrami nad blokovima veličine 4x4 rade za sedam kutova i 360 stupnjeva.

6.3. Klasifikacija

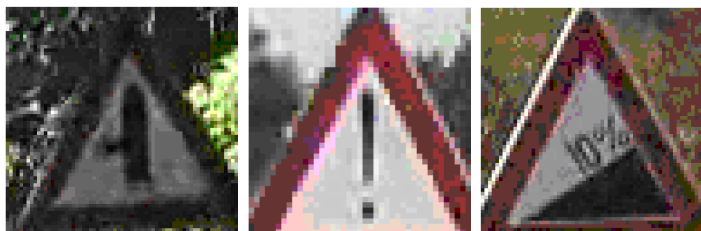
6.3.1. Skup za testiranje

Skup za testiranje (T2010) sastoji se od dvije vrste slike, detekcija i anotacija. Svaki skup sadrži po 991 znaka. Slika 21 prikazuje zastupljene razrede i razdiobu broja primjeraka po razredima.



Slika 21 Zastupljenost znakova u skupu za testiranje.

Anotacije predstavljaju ručno označene znakove, koji su pravilno centrirani. Za razliku od anotacija, detekcije su automatski označeni znakovi. Pri takvom označavanju, detektirani znakovi su često neprecizno locirani. Slika 22 prikazuje znakove na kojima se vidi pomak od idealne pozicije.



Slika 22 Neprecizno locirani znakovi

Kako skup za testiranje sadrži nepostojeće klase znakova iz skupa za učenje, dolje navedeni rezultati klasifikacije odnose se samo na one znakove koji se nalaze u skupu za učenje.

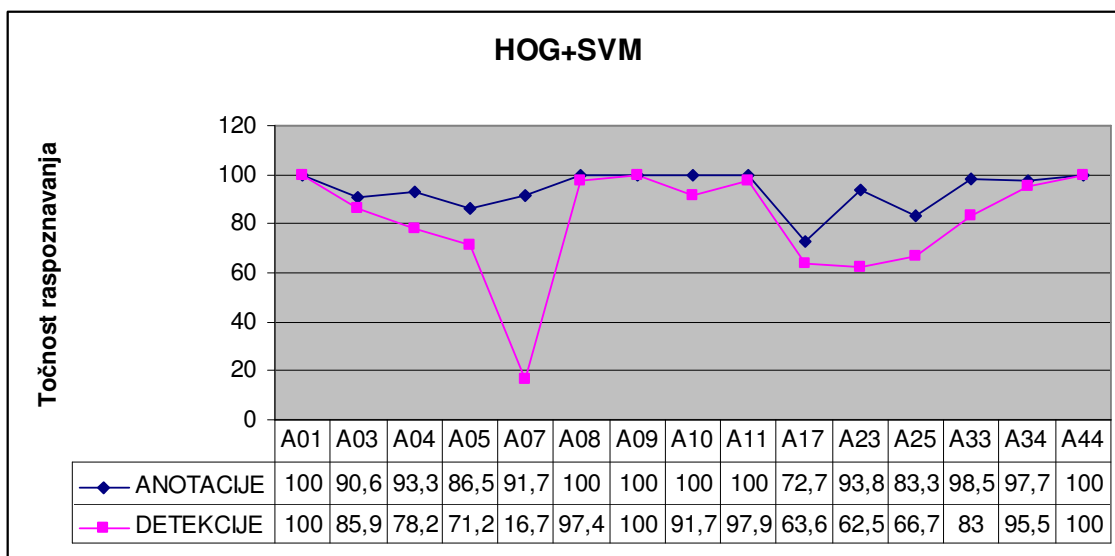
6.3.2. Stroj s potpornim vektorima

Rezultati navedeni u tablici 1 prikazuju postotak točnosti klasifikacije stroja s potpornim vektorima nad znakovima čije su značajke izlučene histogramom orijentacije gradijenta. Implementacija algoritma je preuzeta iz rada [3].

	HOG+SVM	
Anotacije	96,6089%	903/934
Detekcije	83,6188%	781/934

Tablica 1 Rezultati klasifikacije HOG+SVM

Rezultati klasifikacije ručno označenih znakova (anotacije) daleko su bolji od automatski detektiranih znakova (detekcije). Razlog je decentraliziranost detektiranih slika, pri čemu se gube bitne informacije za raspoznavanje znaka



Slika 23 Rezultati klasifikacije SVM-a po klasama.

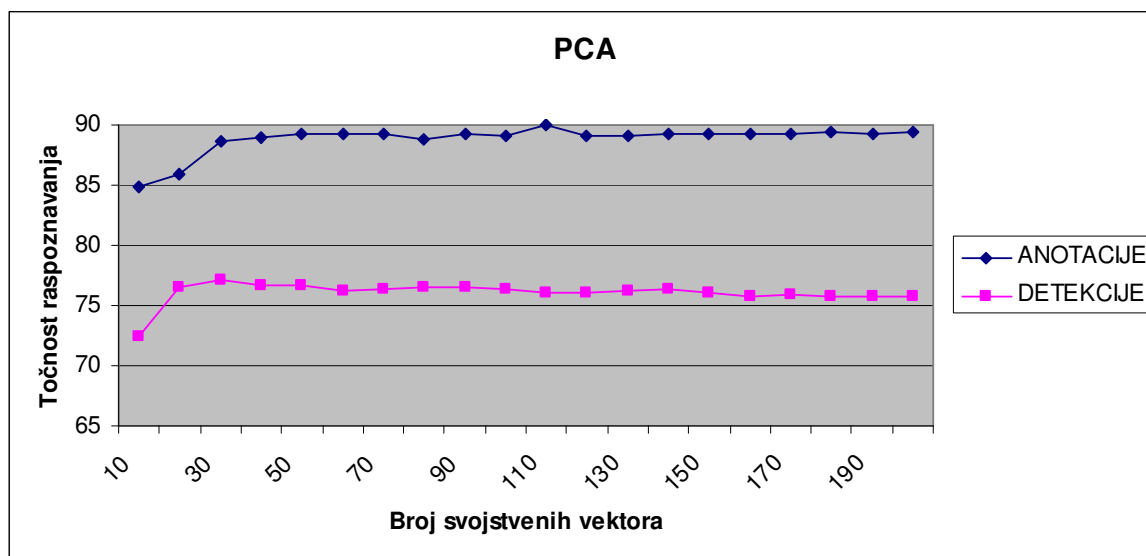
Na slici 23 prikazani su rezultati klasifikacije za pojedinu klasu. Najlošiji rezultati klasifikacije su za klase A07, A17, A23 i A25. Razlog tomu je mali broj uzoraka pojedinih klasa u skupu za učenje (redom 4, 26, 11, 21) kao i jako loša preciznost lociranja znaka pri automatskoj detekciji. Klasa A07 osim što sadrži najmanje uzoraka u skupu za učenje, u skupu za testiranje (detekcije) ima uzorke malih dimenzije, loše kvalitete te neprecizno locirane. Upravo zbog toga odstupanje u točnosti raspoznavanja za detekcije je najprimjetnije. Slika 24 prikazuje neke od uzoraka klase A07.



Slika 24 Uzorci iz klase A07

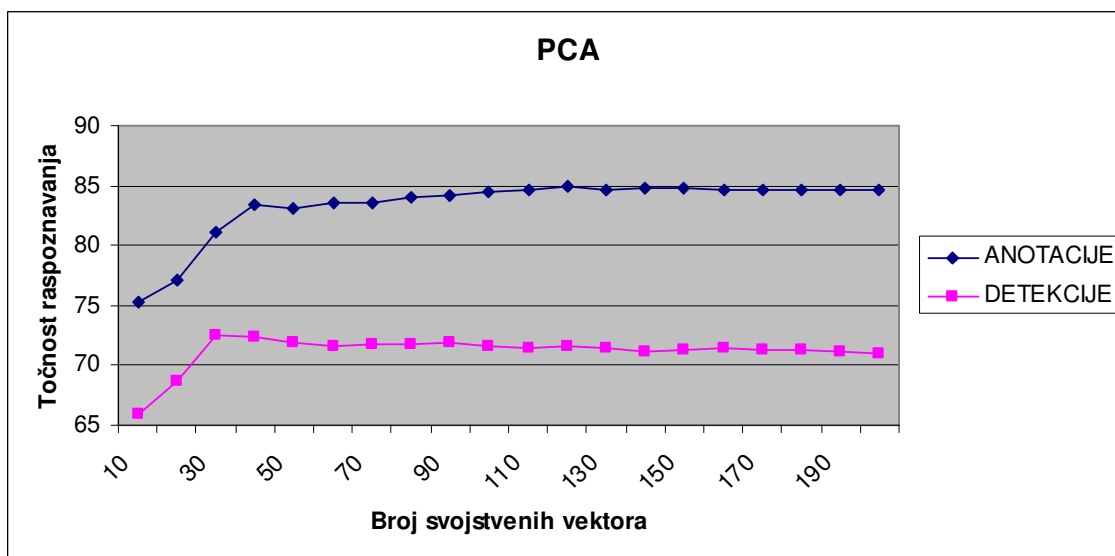
6.3.3. Analiza svojstvenih komponenti

Točnost raspoznavanja analizom svojstvenih komponenti (PCA) uvelike ovisi o broju svojstvenih vektora [17]. Svojstveni vektori s najvećim svojstvenim vrijednostima su upravo oni koji sadrže najvažnije informacije.



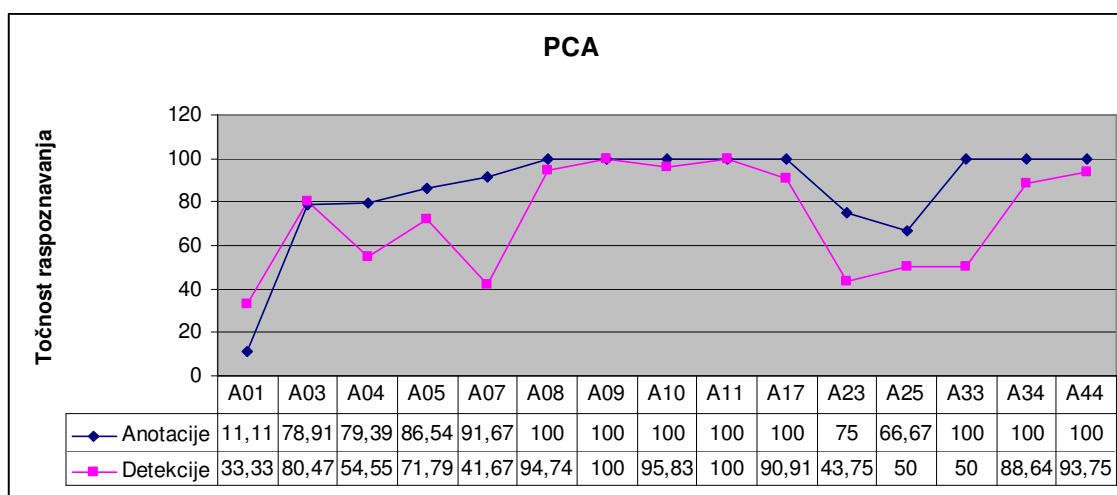
Slika 25 PCA klasifikacija za ručno i automatski označene znakove (64X64).

Slika 25 prikazuje točnost raspoznavanja ručno označenih znakova (anotacija) i automatski označenih znakova (detekcija) u ovisnosti o broju svojstvenih vektora u intervalu od 10 do 200 (korak 10). Plava linija prikazuje točnost raspoznavanja anotacija. Ispravnost klasifikacije raste do 40 svojstvenih vektora. Najbolja točnost klasifikacije anotacija se postiže za 110 svojstvenih vektora. Iznad tog broja, točnost se smanjuje i potom stagnira. Ljubičasta linija prikazuje točnost raspoznavanja automatski označenih znakova (detekcija) u ovisnosti o broju svojstvenih vektora. Maksimum se postiže za 30 svojstvenih vektora. U intervalu od 40 do 150, znatno se ne mijenja. Nakon 150 svojstvenih vektora, smanjuje se. Gledajući prosječnu točnost raspoznavanja detekcija naspram anotacija, vidi se da je klasifikacija detekcija lošija. Razlog je decentraliziranost detektiranih slika, pri čemu se gube bitne informacije za raspoznavanje znaka. Dimenzije slika koje su testirane su 64x64. Dolje navedena slika prikazuje rezultate za slike dimenzija 48x48.



Slika 26 PCA klasifikacija za ručno i automatski označene znakove (48X48).

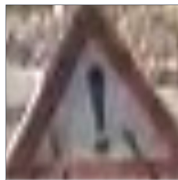
Rezultati nad slikama manjih dimenzija su lošiji i to za zamjetnih 5%. Maksimum raspoznavanja kod anotacija postiže se za 110 svojstvenih vektora, a kod detekcija za 40. Vidimo da dimenzije slike utječu na točnost klasifikacije. Promjenom veličine slike (*eng. resize*), smanjenjem dimenzija, pikseli se interpoliraju te dolazi do gubitka informacija koje u ovom slučaju bitno utječu na rezultate raspoznavanja.



Slika 27 Rezultati klasifikacija PCA po klasama.

Na slici 27 dani su rezultati klasifikacije pojedine klase metodom PCA. Slike su dimenzija 64x64. Odabran je broj svojstvenih vektora koji je dao najbolje rezultate. Za anotacije broj svojstvenih vektora je 110, a za detekcije je 40. Najlošije

klasificirane klase su A01, A07, A23, A25 i A33. Radi se o klasama koje imaju mali broj slika u skupu za učenje (A07, A23, A25, A33) kao i klasama koje su loše kvalitete u skupu za testiranje (A01). Na slici 28 dan je uzorak iz klase A01.



Slika 28 Klasa A01

6.3.4. Fisherova linearna diskriminanta

Točnost raspoznavanja FLD metodom dana je donjim tablicama. Slike su prethodno obrađene na jednak način kao i za PCA metodu (*Dodatak A*). Posivljene su i skalirane na dimenzije 64x64, odnosno 48x48. Korištenjem maske izdvojeno je područje interesa i izjednačeni su histogrami u svrhu pojačanja kontrasta.

Prethodno su dimenzije uzoraka sa originalnih dimenzija smanjene na 40, 75, 110 metodom PCA [17]. Tablica 2 prikazuje rezultate u ovisnosti o različitom broju svojstvenih vektora za slike dimenzija 64x64, tablica 3 za slike dimenzija 48x48.

FLD	40	75	110
Anotacije	93,7901	93,3619	92,3983
Detekcije	77,5161	77,409	78,9079

Tablica 2 Klasifikacija metodom FLD (64x64)

FLD	40	75	110
Anotacije	92,0686	92,0686	93,462
Detekcije	79,7428	79,7428	75,4555

Tablica 3 Klasifikacija metodom FLD (48x48)

Iz tablice 2 vidimo da su za anotacije najbolji rezultati za 40 svojstvenih vektora, a kod detekcija za 110. Za dimenzije 48x48 (tablica 3), najbolji rezultati za

anotacije su za 110 svojstvenih vektora, a za detekcije za 40 svojstvenih vektora. Zanimljivo je primijetiti da smanjenje dimenzija slike na 48x48 poboljšava raspoznavanje detekcija, a pogoršava raspoznavanje anotacija.

Tablice 4 i 5 prikazuje rezultate FLD kada se prethodno odaberu značajke korištenjem histograma orijentacije gradijenta.

HOG+FLD	40	75	110
Anotacije	92,1842	92,3983	91,863
Detekcije	82,9764	78,9079	78,1585

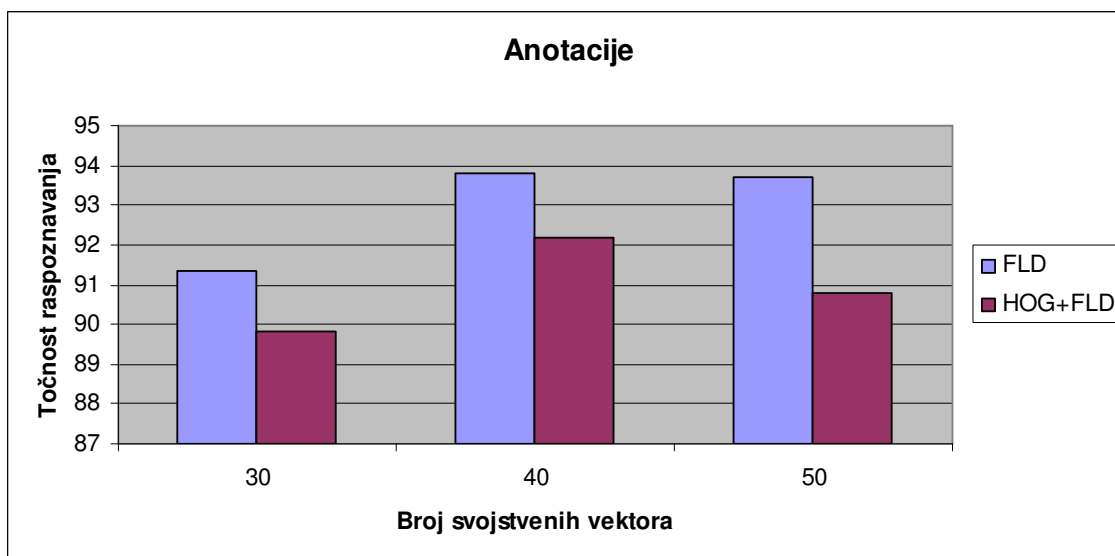
Tablica 4 HOG+FLD (64x64)

HOG+FLD	40	75	110
Anotacije	88,2227	88,2227	90,257
Detekcije	81,7987	81,7987	78,3726

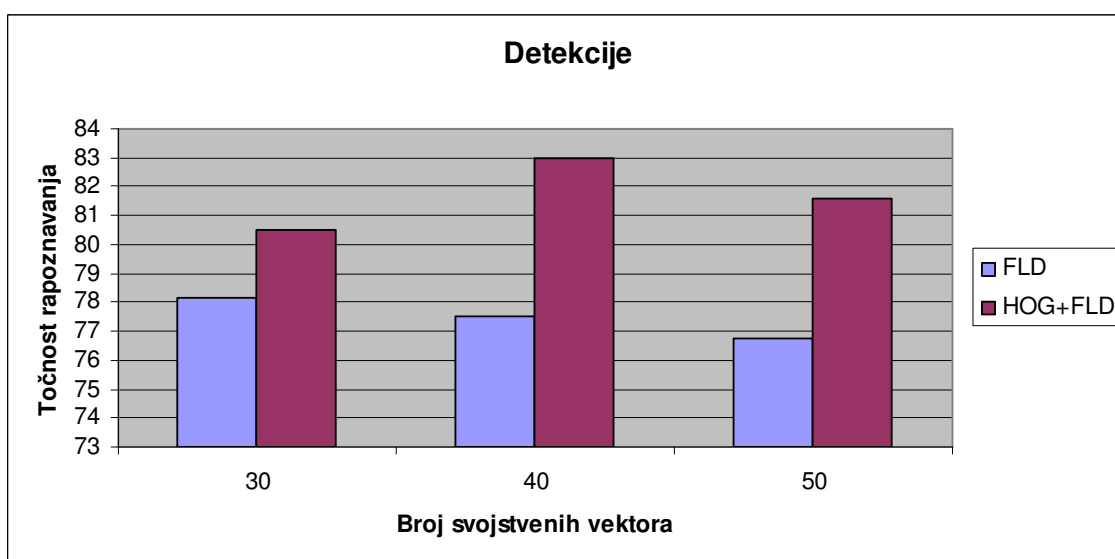
Tablica 5 HOG+FLD (48x48)

Iz priloženih rezultata vidi se da je korištenje histograma orijentacije povećalo točnost klasifikacije detekcija, a smanjilo anotacija. Histogrami orijentacije gradijenta se računaju nad blokovima smještenim u središtu slike, gdje se nazale relevantne informacije po kojima se znakovi razlikuju. Kako automatski detektirani znakovi nisu dobro centrirani, točnost se povećava jer se HOG-om izlučuju bitne značajke čime se povećava otpornost na pomake slike. Vidno povećanje točnosti klasifikacije za detekcije vidi se za odabrani broj svojstvenih vektora 40 (64x64.) te 110 (48x48). Ukupna klasifikacija bolja je pri dimenzijama 64x64.

Radi provjere rezultata vezanih za smanjenje točnosti klasifikacije pri kombinaciji HOG+FLD za anotacije, dodatno je testirano raspoznavanje za broj svojstvenih vektora 30 i 50 (dimenzije 64x64).



Slika 29 Točnost raspoznavanja anotacija u ovisnosti o izlučivanju značajki HOG-om.



Slika 30 Točnost raspoznavanja detekcija u ovisnosti o izlučivanju značajki HOG-om.

Iz priloženih rezultata vidimo da izlučivanje značajki histogramom orijentacije gradijenta povećava točnost klasifikacije detekcije, što je bilo i očekivano, a iznenađujuće je da smanjuje točnost klasifikacije anotacija. Za očekivati je bilo da neće znatno utjecati na njihove rezultate raspoznavanja.

6.3.5. Veličina skupa za učenje

U poglavlju 6.1. opisan je korišteni skup za učenje. Broj uzoraka po razredu kreće se od 4 do 248. Razlike u broju uzoraka za učenje pojedine klase su dosta velike. Slijedeće testiranje provest će se samo nad klasam koje imaju preko 80 uzoraka u skupu za učenje. Takvih razreda je 10 i prikazane su slikom 31.



Slika 31 Klase sa velikim brojem uzoraka

Rezultati testiranja dani su tablicom 6. Iz priloženog se vidi da broj uzoraka u klasi uvelike utječe na točnost klasifikacije. Što je broj uzoraka u skupu za učenje veći to su rezultati klasifikacije bolji.

SVM		
Anotacije	99,57%	233/234
Detekcije	97,44%	228/234
PCA		
Anotacije	95,65%	223/234
Detekcije	92,46%	217/234
FLD		
Anotacije	97,35%	228/234
Detekcije	94,48%	221/234

Tablica 6 Klasifikacija velikih klasa SVM-om

6.3.6. Problem nepreciznog lociranja

Uspješnost klasifikacije detekcija pokazala se relativno niskom. Pri raspoznavanju uzoraka, u ovom slučaju slika znakova, trebala bi klasifikacija biti invarijantna na translaciju i skaliranje ulaznih podataka [6]. Jedan od pristupa rješavanja ovog problema je prilagodba skupa za učenje. Osnovna ideja je dodavanje lošije pozicioniranih primjeraka znakova u skup, poput isječaka prikazanih na slici 22. Dodavanje lošije pozicioniranih primjeraka u skup za učenje svodi se na generiranje istih pomoću operacija pomaka, uvećanja i smanjenja nad početnim skupom za učenje.

Horizontalni i vertikalni pomaci kao i uvećanje modelirani su normalnom razdiobom s odgovarajućim parametrima [16]. U skup je dodano 10 novih, pomaknutih znakova za svaki znak (slika 32).



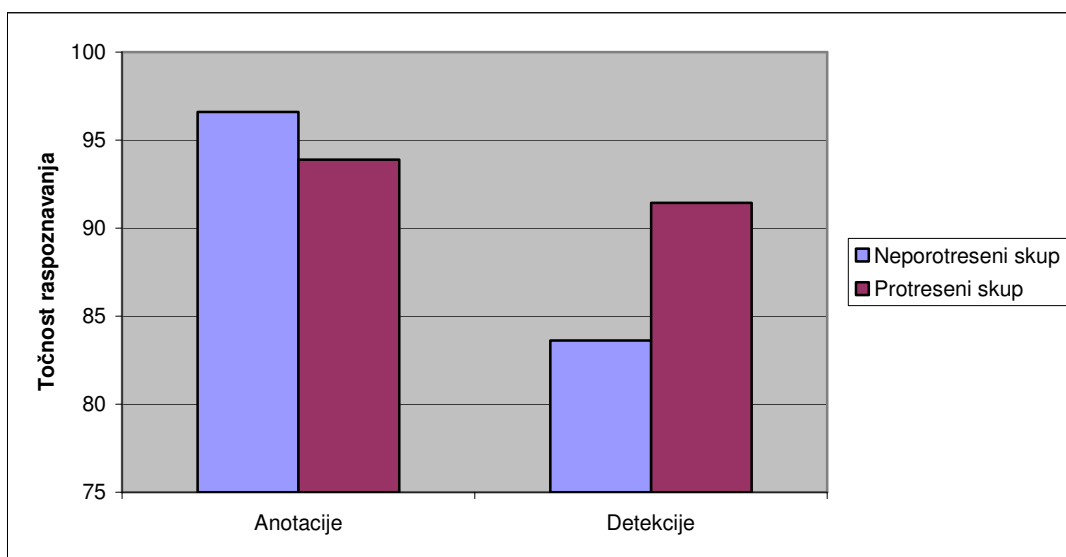
Slika 32 10 novih pomaknutih slika

Tablica 7 prikazuje rezultate klasifikacije SVM-a za detekcije i anotacije. Razdioba pomaka po x-osi modelira se normalnom razdiobom s parametrima ($\mu=-0.48$, $\delta=1.15$). Razdioba pomaka po y-osi modelira se normalnom razdiobom s parametrima ($\mu=-0.74$, $\delta=1.31$). Razdioba uvećanja se modelira s normalnom razdiobom s parametrima ($\mu=0.83$, $\delta=22$).

HOG+SVM		
Anotacije	93,8972	877/934
Detekcije	91,4347	854/934

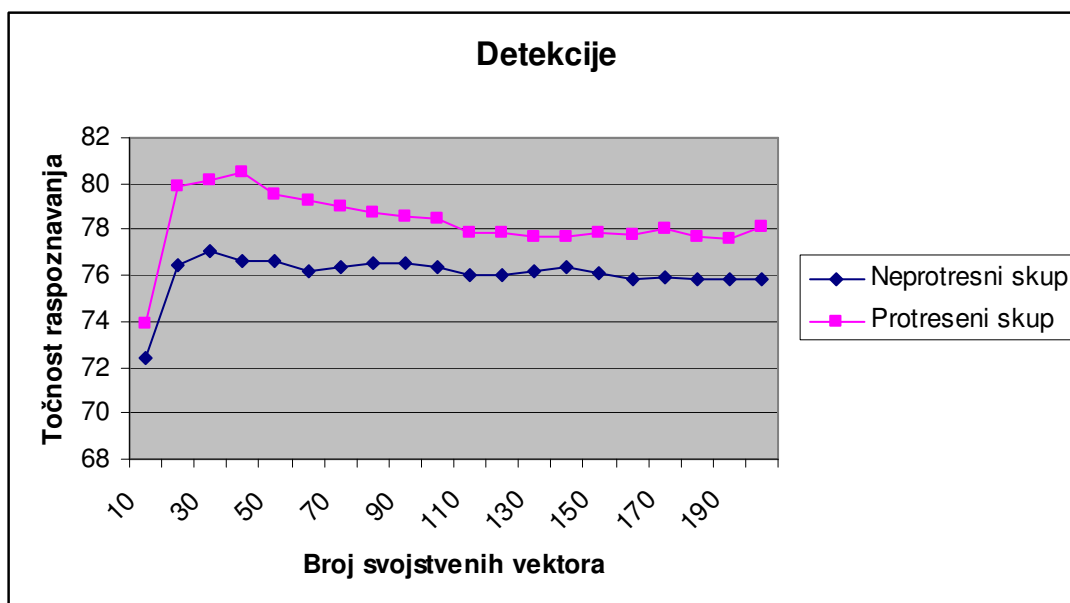
Tablica 7 Klasifikacija SVM s novim, proširenim skupom za učenje.

Rezultati tablice 7 pokazuju veliko poboljšanje pri klasifikaciji detekcija, s 83,6188% na 91,4347%. Točnost klasifikacije anotacija se smanjila.



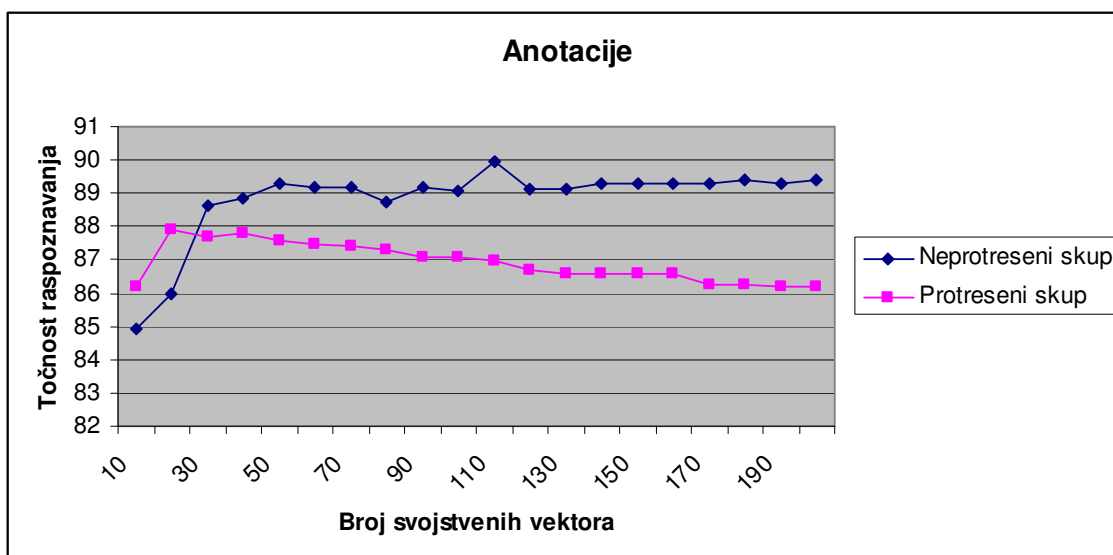
Slika 33 Usporedba SVM klasifikacije nad različito treniranim skupovima za učenje.

Na slici 33 grafički je predložena razlika rezultata klasifikacije anotacija i detekcija nad različito treniranim skupovima za učenje (nad neprotresenim skupom i nad protresenim skupom). U protresenom skupu dodani su lošije pozicionirani uzorci generirani na već opisan način (Dodatak A), dok neprotreseni skup podrazumijeva originalni skup za učenje u kojem su uzorci (znakovi) precizno locirani.



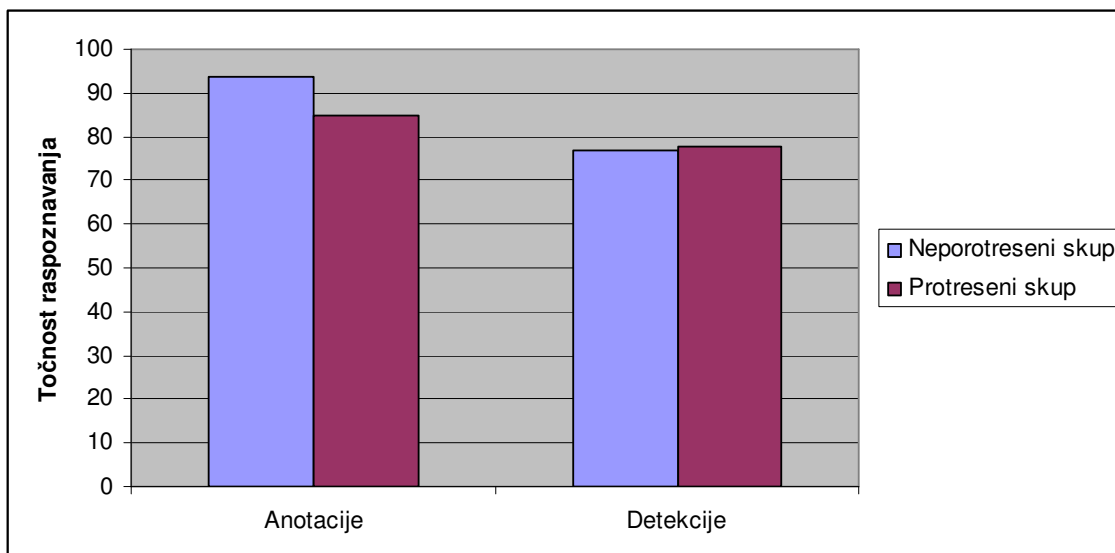
Slika 34 Usporedba klasifikacije detekcija nad različito treniranim skupovima za učenje PCA metodom.

Na slici 34 možemo vidjeti usporedbu rezultata klasifikacije detekcija za PCA metodu nad protresenim i neprotresenim skupom za učenje. Vidimo da neprotreseni skup daje bolje rezultate, najveća razlika je pri 40 svojstvenih vektora (sa 76,695% na 80,5247%).



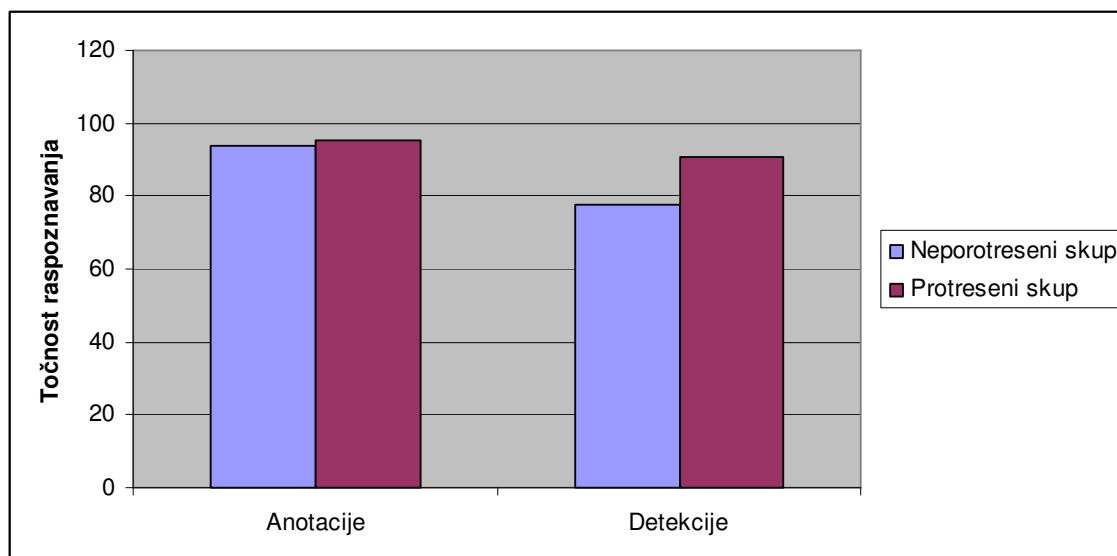
Slika 35 Usporedba klasifikacije anotacija nad različito treniranim skupovima za učenje PCA metodom.

Slika 35 prikazuje usporedbu rezultata klasifikacije anotacija PCA metodom nad različitim skupovima za učenje. Točnost raspoznavanja anotacija pri učenju nad protresenim skupom se smanjuje nakon 30 svojstvenih vektora. Vidimo da protresanje skupa ne poboljšava klasifikaciju ručno označenih znakova (anotacija).



Slika 36 Usporedba klasifikacije metodom FLD nad različito treniranim skupovima za učenje

Slika 36 prikazuje usporedbu klasifikacije anotacija i detekcija nad različito treniranim skupovima za učenje FLD metodom. Vidimo da se klasifikacija anotacija pogoršala pri učenju nad protresenim skupom, dok se klasifikacija detekcija poboljšala za 1%.



Slika 37 FLD nad neprotresenim skupom za učenje, klasifikacija 1-NN nad protresenim skupom za učenje

Rezultati dani slikom 37 dobiveni su tako da je FLD naučen nad neprotresenim skupom, a klasifikator 1-NN klasificira nad protresenim skupom. Time su se rezultati znatno poboljšali. U slučaju detekcija poboljšanje je 13 %, a u slučaju anotacija 2%. Može se zaključiti da kondicioniranje skupa za učenje loše utječe na kvalitetu transformacije u nižedimenzionalni prostor.

6.3.7. Klasifikacija SVM-om za više razreda

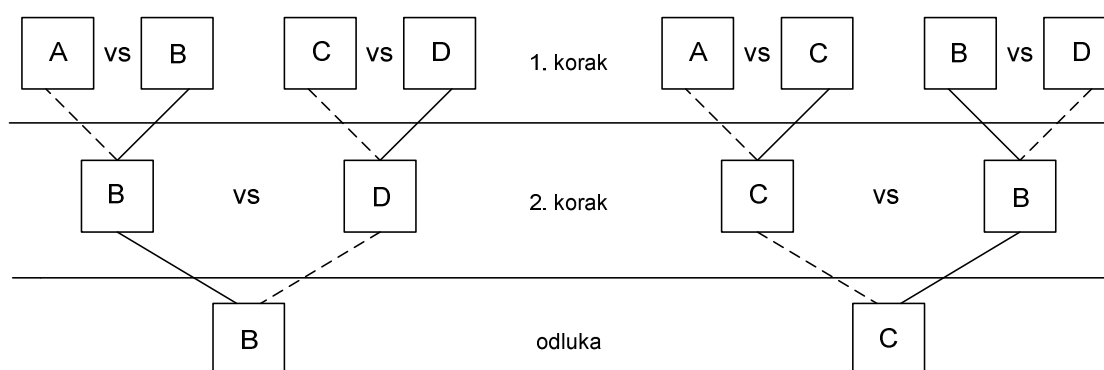
Stroj s potpunim vektorima je originalno binarni klasifikator. U ovom radu korišten je za klasifikaciju 25 razreda. Postoji više načina kojim se pristupa prilagodbi SVM-a kao klasifikatora za više razreda [18]. Tri najčešća su: jedan protiv jednoga, jedan protiv svih i DAGSVM (eng. *Directed Acyclic Graph Support Vector Machines*) [18], [19].

Pri klasifikaciji jedan protiv jednoga za sve moguće parove klasa grade se binarni klasifikatori. Takvih klasifikatora ima $n(n-1)/2$ gdje je n broj klasa. Svaka klasifikacija glasa za jednu od dvije klase te se na kraju uzorak pridjeljuje klasi sa najviše glasova.

Strategija jedan protiv svih konstruira n klasifikatora, gdje svaki klasifikator ima dvije klase: klasu tog znaka i klasu u koju spadaju svi ostali znakovi. Svaki klasifikator osim izlaza daje i mjeru sigurnosti u svoj izbor. Od svih klasifikatora koji

su izabrali klasu znaka, a ne klasu svi ostali, izabire se onaj koji ima najveću mjeru sigurnosti.

DAGSVM je pristup u kojem se gradi usmjereni aciklički graf koji izgleda kao obrnuto binarno stablo gdje su listovi sve raspoložive klase. U svakom koraku, klase koje nisu eliminirane upare se i onda se nad njima vrši binarna klasifikacija. Na taj način, svakim korakom se prepolovi broj klasa, dok ne preostane jedna koja je ujedno i odgovor klasifikatora. Primjer je dan slikom 34 [16].



Slika 38 SVM za više razreda [3]

Slika 38 prikazuje dva primjera DAGSVM stabala. Klase su podijeljene u parove. Na lijevoj strani su parovi (A;B) i (C,D), a na desnoj parovi (A,C) i (B,D). U svakom koraku se radi binarna klasifikacija. U slijedeći korak idu samo pobjednici (puna linija), (A protiv B, B pobjeđuje, C protiv D, D pobjeđuje, A protiv C, C pobjeđuje, B protiv D, B pobjeđuje). Gubitnici se isključuju (iscrtana linija). Proces se izvodi dok ne preostane samo jedna klasa. Različito uparivanje može uzrokovati različite rezultate kao što se i vidi u primjeru sa slike.

Tokom evaluacije naših eksperimenata korištena je biblioteka LIBSVM koja inicijalno koristi strategiju jedan protiv jednoga. U ovom testiranju koristimo DAGSVM implementiran u radu [16] . Rezultati su dani tablicom 8.

SVM		
Anotacije	89,677%	837/934
Detekcije	80,0556%	752/934

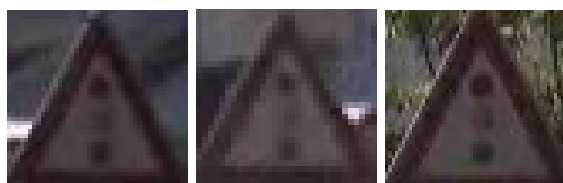
Tablica 8 Rezultati SVM klasifikacije korištenje strategije DAGSVM

Kritične klase znakova zbog kojih se točnost smanjuje su klase A03, A04, A05 i A23. U slučaju detekcija radi se i o klasi A33. Klase A03, A04 i A05 imaju premalo uzoraka u skupu za učenje naspram broju uzoraka u skupu za testiranje. Omjeri su dani tablicom 9.

	Broj uzoraka u skupu za učenje	Broj uzoraka u skupu za testiranje	Broj pogrešno klasificiranih uzoraka	
			Anotacije	Detekcije
A03 	42	128	25	33
A04 	26	165	18	48
A05 	30	156	28	42
A23 	11	16	15	15
A33 	38	206	1	26

Tablica 9 Statistika loše klasificiranih klasa

Uzorci iz klase A23 osim male brojnosti ujedno su i loše kvalitete. Neke od primjera uzoraka iz skupa za učenje dani su slikom 39.



Slika 39 Uzorci iz klase A23

Za klasu A33 velik je broj krivo klasificiranih uzoraka pri testiranju detekcija. Neki od znakova koji se loše prepoznaju dani su slikom 40. Radi se o neprecizno lociranim znakovima koji su velikim dijelom odsječeni iz slike.



Slika 40 Uzorci iz klase A33

Za sve znakove koji su krivo klasificirani, najčešće se radi o slikama malih dimenzija.

Iz priloženih rezultat vidimo da strategija DAGSVM nije poboljšala klasifikaciju. Kako je korišten implementirani SVM [16], brzina izvođenja je bolja, ali su rezultati lošiji. Možemo zaključiti da je korištenje biblioteke LIBSVM po uputama iz [19] najbolja opcija za postizanje zadovoljavajućih rezultata.

7. ZAKLJUČAK

U sklopu ovog rada proučeni su različiti pristupi iz literature za raspoznavanje prometnih znakova. Uhodani su i ispitani postupci klasifikacije k-NN i SVM te pristupi za smanjenje dimenzionalnosti uzoraka (PCA, FLD).

Svi algoritmi trenirani su nad istim skupom za učenje, a potom testirani nad ručno označenim znakovima (anotacije) i automatski detektiranim znakovima (detekcije). Priprema slika i odabir vektora značajki ima važan utjecaj na rezultate raspoznavanja. Korištenje histograma orijentacije gradijenta pokazala se kao uspješna metoda za izlučivanje vektora značajki. Značajno je povećana učinkovitost klasifikacije metodom SVM. Pri korištenju za FLD metodu radi se o poboljšanju od 5% za detekcije.

Točnost raspoznavanja ovisi o veličini skupa za učenje. Znakovi onih klasa koje imaju više slika u skupu za učenje biti će bolje klasificirani od klasa s malim brojem slika u skupu za učenje.

Uspješnost klasifikacije PCA metodom ovisi o broju svojstvenih vektora. Optimalni rezultati postignuti za 40 svojstvenih vektora.

Pomicanje slika u skupu za učenje poboljšalo je rezultate klasifikacije detekcije. Za SVM točnost raspoznavanja je 91% (poboljšanje od 7%), za PCA 81 % (poboljšanje od 4%) te za FLD je 90 % (poboljšanje od 13%).

8. LITERATURA

- [1] K. Brkić, A. Pinzl, S. Šegvić, Traffic sign detection as a component of an automated traffic infrastructure inventory system, Proceedings of the annual Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR), Stainz, Austria, 2009.
- [2] A.Žabčić, Eksperimentalna evaluacija metoda za prepoznavanje prometnih znakova, Diplomski rad, FER, Zagreb, veljača 2011.
- [3] I.Bonači, I.Kovaček, I.Kusalić, Z.Kalafatić, S. Šegvić, Addressing false alarms and localization inaccuracy in traffic sign detection and recognition, The Computer Vision Winter Workshop, Graz, Austria, 2011.
- [4] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Los Alamitos, CA, USA, 2005, vol. 1, pp. 886-893.
- [5] C. L. Novak and S. A. Shafer, Anatomy of a color histogram, in *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, IL, USA, pp. 599-605.
- [6] Duda, R., Hart, P., Stork, D., Pattern Classification, Second edition, John Wiley and Sons, Inc., 2001, str. 114-124.
- [7] S. Ribarić, Raspoznavanje uzoraka, predavanje 5 (SVM), FER, rujan 2009.
- [8] Turk, M., Pentland, A. Eigenfaces for Recognition, Journal of Cognitive Neuroscience, pp. 71-86.
- [9] I.Sučić, Primjena metode PCA nad skupom slika znakova, Završni rad, FER, Zagreb, lipanj 2009.
- [10] N.Elezović, Linearna algebra, Svojstveni vektori, Element, Zagreb, 1995.
- [11] Smith, L. I, A tutorial of Principal Component Analysis, Journal Article, Cornell University, USA, 2002.
- [12] S. Theodoridis, K. Kountroumbas, Pattern Recognition, Academic press, 2006.
- [13] S. Ribarić, Raspoznavanje uzoraka, predavanje 3 (FLD), FER, Zagreb, rujan 2009.
- [14] I. Gokcen and J. Peng, Comparing Linear Discriminant Analysis and Support Vector Machines, in *Advances in Information Systems*, vol. 2457, T. Yakhno, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 104-113.
- [15] A combined SVM and LDA approach for classification, vol. 3, pp. 1455- 1459 vol. 3, Aug. 2005.
- [16] I.Bonači, I.Kusalić, I. Kovaček, Detekcija i raspoznavanje prometnih znakova u video snimci, FER, Zagreb, 2010.
- [17] Linear discriminant analysis, Wikipedia, the free encyclopedia, 13.03.2011, http://en.wikipedia.org/wiki/Linear_discriminant_analysis.
- [18] Chang, Chih-Chung; Lin, Chih-Jen, A Comparison of methods for Multi Class Support Vector Machines, 2010, <http://www.csie.ntu.edu.tw/~cjlin/papers/multisvm.pdf>.
- [19] Chang, Chih-Chung; Lin, Chih-Jen: A practical guide to Support Vector Classification, 2010, <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.
- [20] M. Ellis and F. Lundh, "Python Imaging Library Overview." 12-Mar-2002.
- [21] ImageMagick Users Guide, E. I. du Pont de Nemours and Company, 1999, <http://www.imagefolio.com/ImageMagick/ImageMagick.pdf>.
- [22] Open Source Computer Vision Library, Reference Manual, http://hawaiiilibrary.net/eBooks/Giveway/Technical_eBooks/OpenCVReferenceManual.pdf.
- [23] Chang, Chih-Chung; Lin, Chih-Jen: LIBSVM: a library for support vector machines, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [24] Jacobi eigenvalue algorithm, Wikipedia, the free encyclopedia, 06.03.2011, http://en.wikipedia.org/wiki/Jacobi_eigenvalue_algorithm.

Dodatak A: Opis implementacije

Označavanje prometnih znakova

Programom *Marker* [1] ručno su označeni znakovi na video sekvencama te su stvorene datoteke koje opisuju položaj znakova u pojedinim okvirima video sekvence. Skripta za izrezivanje znakova napisana je u programskom jeziku *Python* (*izrezi.py*) u radu [16]. Korištena je biblioteka PIL (eng. *Python Imaging Library*) koja omogućuje manipulaciju slikama [20]. Napisana skripta omogućuje izdvajanje znaka na dva načina: normalno i protreseno. Normalno izdvajanje podrazumijeva da se znak iz slike izreže po koordinatama definiranim datotekom generiranom programom *Marker* (*sekvence.seq*). Protreseno izdvajanje podrazumijeva da se iz svake slike izreže 10 slika istog znaka pomaknutih na različite načine od originalnog položaja. Ideja je da se korištenjem normalne (Gaussove) razdiobe promijeni pozicija i dimenzija znaka. Normalna distribucija se označava kao :

$$N(\mu, \sigma^2)$$

gdje je μ srednja vrijednost oko koje su raspršene vrijednosti, a σ^2 varijacija. Znakovi su različitih veličina ovisno o udaljenosti od kamere. Pozicija i dimenzija znaka se mijenjaju relativno u odnosu na širinu znaka, što znači da se uži znakovi (ujedno i manji) manje pomiču nego širi. Pomicanje se radi tako da se generiraju za svaki od deset parametara, koordinate x i y opisanog pravokutnika te širina i visina pravokutnika. Potom se generira slučajan broj po zadanoj normalnoj razdiobi koji će se množiti sa širinom slike te dodati zadanom parametru. Množenje sa širinom slike osigurava da pomaci budu veći za veće slike, a manji za manje slike.

Obrada slike

Za obradu slika korišten je programski alat *ImageMagic* [21]. ImageMagic je alat za manipulaciju slikama u različitim formatima. Korištena verzija pokreće se .iz komandne linije. Za rješavanje problema preplitanja napisana je skripta *deinterlace.sh* u diplomskom radu [2]. Odbacivanje polovice linija i interpolacija linija koje nedostaju (eng. deinterlacing) ostvareno je korištenjem naredbe *convert* na sljedeći način:

```
Convert -sample 100%x50% -resize 100%x200% srcImage destImage
```

```
Convert -roll +0-1 -sample 100%x50% -resize 100%x200% srcImage destImage
```

Opcija `-sample 100%x50%` definira da će se iz originalne slike uzeti sve vrijednosti po x osi (100%) i svaka druga po y osi (50%). Opcija `-resize` ovako izdvojenu sliku uvećava po y komponenti na dvostruko veću sliku, na dimenzije koje je slika imala na početku. Prilikom stvaranja druge slike koristi se opcija `-roll`, čija je sintaksa `-roll{+-}x{+-}y`, koja će sliku pomaknuti za jedan redak prema gore kako bi se izdvojili parni redovi.

Za algoritme PCA i FLD, slike je potrebno posiviti te skalirati na dimenzije 64x64 (48x48). Za te potrebe napisana je skripta *resizegray.sh* u diplomskom radu [2]. Željene promjene ostvarene su korištenjem slijedećih naredbi:

```
convert -crop widthxheight+xoffset+yoffset -resize +48x+48
```

```
colorspace Gray -depth 8 srcImage destImage
```

Opcija `-crop` će izrezati pravokutni dio slike širine `width` i visine `height`, pomaknuti od početka slike za `xoffset` u smjeru x osi i `yoffset` u smjeru y osi. Opcija `-resize` će slici promijeniti dimenziju na zadanu, dok će opcija `-colorspace Gray` sliku pretvoriti u sivu sliku.

Histogram orijentacije gradijenta

Histogram orijentacije gradijenta implementiran je u programskom jeziku C++ u radu [16]. Korištena je besplatna biblioteka OpenCV [22] za prikaz, obradu i učitavanje slike. Implementacija je izvedena po postupku objašnjenom u poglavlju 6.2.1.

Prije računanja histograma orijentacije gradijenta potrebno je izračunati gradijente u oba smjera [16]. Gradijent funkcije dvije varijable:

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \vec{i} + \frac{\partial f(x, y)}{\partial y} \vec{j}$$

Kontinuirana varijanta gradijenta ne može se koristiti jer je slika diskretna pa se koriste aproksimacije derivacija:

$$dx = \frac{\partial f(x, y)}{\partial x} = \frac{f(x+1, y) - f(x-1, y)}{2}$$

$$dy = \frac{\partial f(x, y)}{\partial y} = \frac{f(x, y+1) - f(x, y-1)}{2}$$

Iznos gradijenta se računa kao:

$$g = \sqrt{dx^2 + dy^2}$$

Smjer gradijenta računa se kao:

$$\phi = \arctg\left(\frac{dy}{dx}\right)$$

Gradijenti u oba smjera izračunati su korištenjem funkcije OpenCV-a `cvSobel`:

```
void cvSobel( const CvArr* I, CvArr* J, int dx, int dy, int apertureSize=3 );
```

Zadana funkcija je pozivana na način:

```
cvSobel(img,dx,1,0,CV_SCHARR);
```

```
cvSobel(img,dy,0,1,CV_SCHARR);
```

Parametar `CV_SCHARR` označava da se umjesto Sobela koristi 3x3 Scharrov filtar. Scharrov filtar daje precizniji gradijent jer uzima u obzir i dijagonalne vrijednosti. Scharrov operator je definiran maskama:

$$G_x = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \quad G_y = \begin{bmatrix} 3 & 0 & 3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix}$$

Histogrami nad blokovima veličine 6x6 se rade za četiri kuta i 180 stupnjeva, a histogrami nad blokovima veličine 4x4 se rade za sedam kutova i 360 stupnjeva. U obzir su uzeti samo relevantni dijelovi znaka (slika 19). Ukupni vektor značajki je 174.

Stroj s potpornim vektorima

U sklopu ovog rada korištena je implementacija stroja s potpornim vektorima: besplatna biblioteka LIBSVM [23]. Glavna prednost biblioteke LIBSVM je u brzini, a nedostatak je to što ju je teže modificirati i eksperimentirati s djelovanjem pojedinih elemenata stroja s potpornim vektorima. Biblioteka je implementirana u programskom jeziku C++ u radu [16].

Za pripremu slika se koristi biblioteka OpenCV [22]. Prije računanja vektora značajki potrebno je promijeniti dimenzije slike na 48x48, pretvoriti sliku u sivu i normalizirati vrijednosti slike.

Biblioteka LIBSVM osim samog treniranja i testiranja uzoraka omogućava skaliranje ulaznih podataka na vrijednosti u rasponu $[-1,1]$. To se obavlja pozivom programa:

```
svm-scale ulaznaDatoteka > izlaznaDatoteka
```

Parametare C i γ za Gaussovu jezgru moguće je optimirati korištenjem skripte grid.py koja to čini postupkom unakrsne validacije. Stroj se trenira nad skaliranim podacima s odabranim optimalnim parametrima:

```
svm-train -c 32 -g 0.0078125 ulaznaDatoteka
```

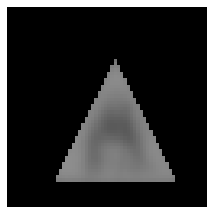
Program će u slučaju uspješnog izvršavanja u istome direktoriju stvoriti datoteku *ulaznaDatoteka.model* koja sadrži parametre modela. Klasifikacija uzoraka se vrši pozivom programa:

```
svm-predict testDatoteka model izlaznaDatoteka
```

U izlaznoj datoteci se nalaze indeksi klase u koje su klasificirani uzorci, redom za sve testne uzorke.

Analiza svojstvenih komponenti

Analiza svojstvenih komponenti (PCA) implementirana je u programskom jeziku C++ u diplomskom radu [2]. Prije same implementacije algoritma (poglavlje 4.1) slike su dodatno obrađene kako bi se poboljšala točnost klasifikacije. Pozadina znaka nije bitna za raspoznavanje. Da bi se izdvojilo područje interesa, korištena je maska (slika 41).



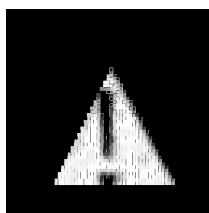
Slika 41 Maska korištena za uklanjanje pozadine

Rezultat njene primjene dan je slikom 42.



Slika 42 Primjena maske na ulaznoj slici

U korištenoj bazi slika različite slike imaju različita osvjetljenja. Da bi se normalizirala, izjednačeni su histogrami. Histogram slike je grafički prikaz distribucije vrijednosti slike. On pokazuje broj slikovnih elemenata za svaku vrijednost. Korištene slike su sive pa postoji samo jedna vrijednost za svaki slikovni element – svjetlina. Ona se nalazi u intervalu [0,255]. Izjednačavanje histograma služi za promjenu osvjetljenja i kontrasta slike na željeni način. Ideja je da se promjenom srednje vrijednosti svjetline slike mijenja osvjetljenje, a promjenom standardne devijacije svjetline mijenja kontrast. Na slici 43 dan je rezultat izjednačavanja histograma ulazne slike.



Slika 43 Slika znaka sa izjednačenim histogramom

Nakon navedene pripreme slika proveden je algoritam metode PCA opisan u poglavlju 4.1. Ulazni podatci organizirani su u kovarijacijsku matricu C , odnosno C' . te je potrebno izračunati svojstvene vektore i svojstvene vrijednosti pripadajuće matrice. Kako se radi o matrici velikih dimenzija, nepraktično je te vrijednosti računati na osnovni način. U tu svrhu razvijeni su razni algoritmi za jednostavnije i brže računanje. Jedan od njih je i Jacobijev rotacijski algoritam [2], [24]. Postupak počinje od pretpostavke da se svaka kvadratna matrica A može dijagonalizirati množenjem sa matricom svojstvenih vektora V na slijedeći način:

$$Q = V^T A V$$

Jacobijev algoritam se bazira na nizu rotacija koje su osmišljene tako da pri svakoj iteraciji podatci izvan glavne dijagonale postaju sve manji dok matrica ne postane dijagonalna. Postupak je slijedeći:

Inicijalizira se matrica $Q(1) = A$, te $V(1) = I$

1. U svakoj iteraciji h se odabire element $Q_{pq}(h)$ izvan glavne dijagonale matrice $Q(h)$ koji će se postaviti na nulu. U cikličkoj verziji algoritma uzimaju se redom svi elementi iznad glavne dijagonale
2. Izračunava se kut iteracije

$$\theta(h) = \frac{1}{2} \tan^{-1} \left(\frac{2Q_{pq}(h)}{Q_{qq}(h) - Q_{pp}(h)} \right) \text{ ako je } Q_{qq}(h) \neq Q_{pp}(h) \text{ inače } \theta(h) = 90^\circ$$

3. Računa se Jacobijeva rotacijska matrica:

$$P(h) = \begin{bmatrix} 1 & \dots & \cdot & \dots & \dots & \dots & \cdot \\ \vdots & \ddots & \cdot & \cdot & \cdot & \cdot & \vdots \\ \cdot & \cdot & \cos(\theta(h)) & \cdot & -\sin(\theta(h)) & \cdot & \cdot \\ \vdots & \cdot & \cdot & 1 & \cdot & \cdot & \vdots \\ \cdot & \cdot & \sin(\theta(h)) & \cdot & \cos(\theta(h)) & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot & \ddots & \vdots \\ \cdot & \dots & \dots & \dots & \cdot & \dots & 1 \end{bmatrix} \begin{matrix} \leftarrow p \\ \leftarrow q \end{matrix}$$

$\uparrow \qquad \qquad \uparrow$
 $p \qquad \qquad q$

4. Računa se matrica $Q(h+1)$: $Q(h+1) = P(h)^T Q(h) P(h)$
5. Računa se procjena matrice V :
6. Ponavljati korake 2 do 5 dok se ne zadovolji zadani kriterij

Vrijednosti na dijagonali matrice Q su svojstvene vrijednosti, a svojstveni vektori se nalaze u matrici V .

Zbog složenosti opisanog algoritma, za implementaciju se koristi procjena sinusa i kosinusa, te umjesto računanja matrice Q preko množenja matrica (korak 4), definirat će se jednostavniji izračun. Sama implementacija Jacobijeve rotacijske metode preuzeta je iz biblioteke OpenCV. Na početku se inicijalizira matrica V na jediničnu matricu jednakih dimenzija kao i matrica C , a početna matrica Q jednaka je ulaznoj matrici C . Zatim se mijenjaju matrice Q i V određeni broj puta na slijedeći način: uzimaju se redom sve vrijednosti p i q :

$$p = 1, \dots, n-1$$

$$q = p+1, \dots, n$$

Računaju se sinus i kosinus

$$\sin \theta = \frac{x}{\sqrt{2(1 + \sqrt{1-x^2})}}$$

$$x = \frac{-Q_{pq}}{\sqrt{Q_{pq}^2 + y^2}} \text{ ako je } y > 0:$$

$$x = \frac{Q_{pq}}{\sqrt{Q_{pq}^2 + y^2}} \text{ ako je } y < 0$$

$$y = \frac{1}{2}(Q_{pp} - Q_{qq})$$

$$\cos \theta = \sqrt{1 - \sin^2 \theta}$$

Nove vrijednosti matrice Q računaju se na slijedeći način:

$$i = 1, \dots, p-1$$

$$Q_{ip} = Q_{ip} \cos \theta - Q_{iq} \sin \theta$$

$$Q_{iq} = Q_{iq} \cos \theta + Q_{ip} \sin \theta$$

$$i = p, \dots, q-1$$

$$Q_{pi} = Q_{pi} \cos \theta - Q_{iq} \sin \theta$$

$$Q_{iq} = Q_{iq} \cos \theta + Q_{pi} \sin \theta$$

$$i = q, \dots, n$$

$$Q_{pi} = Q_{pi} \cos \theta - Q_{qi} \sin \theta$$

$$Q_{iq} = Q_{qi} \cos \theta + Q_{pi} \sin \theta$$

$$Q_{pp} = Q_{pp} \cos^2 \theta - Q_{qq} \sin^2 \theta - a$$

$$Q_{iq} = Q_{pp} \sin^2 \theta + Q_{pi} \cos^2 \theta + a$$

$$a = 2Q_{pq} \cos \theta \sin \theta$$

$$Q_{pq} = Q_{qp} = 0$$

Prethodno opisani algoritam ponavlja se dok matrica Q ne bude dovoljno dijagonalizirana. Uvjet zaustavljanja je definiran korištenjem parametra

ε prethodno postavljenog na neku malu vrijednost (u ovom slučaju e^{-15}). Definira se:

$$Q_{norm} = \sqrt{\sum_{i=0}^{i-1} \sum_{j=0}^{j-1} Q_{ij}^2}$$

$$Q_x = \frac{Q_{norm}}{n} \varepsilon$$

$$Q_{max} = Q_{norm}$$

Postupak se ponavlja dok nije zadovoljen uvjet

$$Q_{max} > Q_x$$

U svakom koraku se vrijednost Q_{max} mijenja na slijedeći na način:

$$Q_{max} = \frac{Q_{max}}{n}$$

Na kraju se sortiraju vektori po pripadajućim svojstvenim vrijednostima jer vektori s većim svojstvenim vrijednostima nose više informacija.

Linerana diskriminatna analiza

Linearna diskriminantna analiza implementirana je u programskom jeziku C++ u diplomskom radu [2] po algoritmu objašnjenom u poglavlju 5. Prethodno su slike pripremljene na jednak način kao i slike za metodu PCA: izdvojeno je područje interesa i izjednačeni su histogrami. FLD se može direktno koristiti na skupu za učenje. U ovoj implementaciji su smanjene dimenzije metodom PCA sa prostora 4096 (64x64) na 75 (40 ili 110), a potom je primijenjen postupak FLD.

Naslov, sažetak i ključne riječi

Naslov

Eksperimentalna evaluacija utjecaja odabira značajki na rezultate raspoznavanja prometnih znakova

Sažetak

U ovom radu opisani su postupci klasifikacije k-NN i SVM te pristupi za smanjenje dimenzionalnosti uzoraka (PCA, FLD). Postupci su primijenjeni za raspoznavanje trokutastih prometnih znakova iz video sekvenci dobivenih kamerom postavljenom na krov pokretnog vozila. Slike znakova su obrađene te su vektori značajki izlučeni korištenjem histograma orijentacije gradijenta. Testiranje je izvršeno nad automatski označenim slikama (detekcije) i ručno označenim slikama (anotacije). Problem nepreciznog lociranja znaka (kod detekcija) koji je uzrokovao lošije rezultate raspoznavanja, poboljšan je dodavanjem lošije pozicioniranih znakova u skup za učenje.

Ključne riječ

Raspoznavanje prometnih znakova, stroj s potpornim vektorima, linearna diskriminantna analiza, analiza svojstvenih komponenti, k najbližih susjeda

Abstract, keywords, title

Title

Experimental evaluation of features selection impact on traffic signs classification results

Abstract

In this paper, classification techniques (k-NN, SVM) and techniques for dimensionality reduction (PCA, FLD) are presented. These techniques are used for the classification of triangular traffic signs from the video sequences captured by a camera mounted on top of a moving vehicle. The images are processed and the histograms of oriented gradients are used as feature descriptors. The experiments are evaluated on automatically detected images (detections) and manually detected images (annotations). On the automatically detected images, the traffic signs are imprecisely located. In order to improve classification results, decentralized images are added in the training set.

Keywords

Traffic signs recognition, support vector machines, linear discriminant analysis, principal component analysis, k nearest neighbors