Prepoznavanje znakova ograničenja brzine iz postojećih baza podataka

Matoš, Ivona

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:928216

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-30**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni preddiplomski studij Računarstva

PREPOZNAVANJE ZNAKOVA OGRANIČENJA BRZINE IZ POSTOJEĆIH BAZA PODATAKA

Završni rad

Ivona Matoš

Osijek, 2018.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 31.08.2018.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Ivona Matoš
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3807, 19.09.2017.
OIB studenta:	61233709766
Mentor:	Doc.dr.sc. Zdravko Krpić
Sumentor:	Krešimir Romić
Sumentor iz tvrtke:	
Naslov završnog rada:	Prepoznavanje znakova ograničenja brzine iz postojećih baza podataka
Znanstvena grana rada:	**Obradba informacija (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	31.08.2018.
Datum potvrde ocjene Odbora:	12.09.2018.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 15.09.2018.

Ime i prezime studenta:	Ivona Matoš
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3807, 19.09.2017.
Ephorus podudaranje [%]:	4

Ovom izjavom izjavljujem da je rad pod nazivom**: Prepoznavanje znakova ograničenja brzine iz postojećih baza podataka**

izrađen pod vodstvom mentora Doc.dr.sc. Zdravko Krpić

i sumentora Krešimir Romić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. Uvod 1
1.1. Zadatak završnog rada 3
2. Problematika detekcije i prepoznavanja prometnih znakova 4
2.1. Metode preliminarne obrade slika 4
2.1.1. Standardni medijan filter 4
2.1.2. Gaussov filtar 5
2.2. Detekcija objekta unutar slike 6
2.3. Metoda potpornih vektora 7
3. Realizacija algoritma za prepoznavanje znakova ograničenja brzine 12
3.1. Baza podataka 12
3.2. Dijagram toka algoritma 13
3.3. Funkcije za preliminarnu obradu podataka 14
3.3.1. Funkcije medijan filtra 14
3.3.2. Funkcije Gaussovog filtra 16
3.4. Primjena Houghove transformacije 18
3.5. Prepoznavanje znamenki pomoću SVM-a 21
4. Točnost i ograničenja algoritma za prepoznavanje znakova ograničenja brzine 23
4.1. Programsko rješenje za prepoznavanje znakova ograničenja brzine 29
4.1.1. Trening slika 29
4.1.2. Prepoznavanje prometnih znakova 30
5. Zaključak 33
Literatura 35
Sažetak 37
The speed limit road signs recognition using existing road sign image databases 37
Abstract 37
Životopis 38
Prilozi 39

1. UVOD

Sustav za prepoznavanje prometnih znakova jedan je od temeljnih čimbenika koji uvjetuju razvoj autonomnih vozila na svim razinama. Također, takav sustav predstavlja, između ostalog, pomoć vozaču pri vožnji, obavještavajući ga pritom o zamijećenim prometnim znakovima na kolniku, koje sam vozač možda nije primijetio. Navedeno je omogućeno prikladnom opremom, odnosno kamerama, a korištenjem sve naprednijih tehnologija u području digitalne obrade slika, omogućava se brzo i efikasno detektiranje i prepoznavanje prometnih znakova tijekom vožnje.

Postoji nekoliko ključnih značajki na koje treba obratiti pozornost prilikom izrade ovakvog sustava. Obzirom da postoji više vrsta prometnih znakova različitih oblika, te značajke razlikuju se ovisno o vrsti znaka koji se želi prepoznati. Tako na primjer, znak stop predstavlja šesterokut crvene ispune, u čijem se centru nalazi riječ. Postoje dvije mogućnosti primjećivanja ovakvog znaka: po obliku ili po boji. Opće je poznato da uvjeti vožnje nisu savršeni, odnosno postoji više mogućnosti pojave različitih stupnja osvjetljenja nad prometnim znakom ili jednostavno postoje zastarjeli znakovi izblijedjele boje, stoga detekcija praćenjem boja pojedinih znakova ne mora davati točne rezultate. Na tragu toga, preporuča se odabir praćenja samo oblika znaka ili praćenja obje karakteristike. Međutim, praćenjem obje karakteristike dolazi do gubitka vremena i povećava se tromost algoritma, zato će se za potrebe izrade ovog rada odabrati detekcija prometnih znakova po obliku. Još jedan važan problem koji se pojavljuje uslijed prepoznavanja znakova jest način na koji će se prepoznati unutrašnjost znaka. U svrhu rješavanja toga problema razvijene su mnoge tehnologije, između kojih valja spomenuti podudaranje uzoraka (eng. *Template matching*), metodu potpornih vektora, otvoreni kod za prepoznavanje teksta Tesseract (eng. *Tesseract OCR*) i neuronske mreže.

Znakovi ograničenja brzina označavaju ključan segment u cjelokupnom sustavu za prepoznavanje prometnih znakova te će izvedba i implementacija navedenoga biti primarna odrednica razvoja rada. Iako je u današnje vrijeme u modernim vozilima ovakav sustav izveden konvolucijskim neuronskim mrežama, odnosno dubokim učenjem (eng, *deep learning*), ipak će se, u ovom slučaju, težište staviti na vrste tehnologije strojnog učenja, koje će biti provedene uporabom prikladnih algoritama i funkcija odgovarajućih biblioteka sukladno odabiru programskog jezika u kojem će se izraditi aplikacija za prepoznavanje znakova ograničenja brzine.

Kako bi se navedeno ostvarilo, odabran je C#, objektno-orijentirani programski jezik, koji omogućava bržu i jednostavniju implementaciju potrebnih algoritama u odnosu na C ili C++ jer sadrži ugrađen „sakupljač smeća“ (eng. *Garbage collector*), te nije potrebno izvoditi alokaciju memorije. Sagledavajući problem iz dizajnerske perspektive, C#, svakako, osigurava pregledniju izradu same aplikacije. Budući da je potrebno realizirati algoritme za obradu slike, unutar samog projekta konfigurirana je Emgu CV platforma koja omogućava manipulaciju i obradu slike uslijed korištenja funkcija implementiranih u samu biblioteku.

Obzirom da će se ovaj rad baviti problematikom prepoznavanja prometnih znakova ograničenja brzine, detekcija samog znaka bazirat će se na prepoznavanju kružnice unutar odabrane fotografije i izdvajanju teksta unutar kružnice, a prepoznavanje brojeva odvijat će se na temelju izdvojenih značajki iz kružnice.

Ponajprije, ovaj rad obuhvatit će teme vezane za preliminarnu obradu slika na kojima se želi izdvojiti i prepoznati određeni znak. U ovu fazu uključeno je proučavanje odabrane baze fotografija rMASTIF i uporaba filtera za poboljšanje kvalitete fotografija, odnosno za uklanjanje smetnji u promatranim fotografijama, a to su medijan zaglađivanje (eng. *MedianSmooth*) i zaglađivanje slike uporabom Gaussove formule (eng. *GaussianSmooth*). Nadalje, uporabom funkcija za pronalaženje te crtanje kontura i kružnica na slici, detektirat će se sam prometni znak, koji će zatim biti izdvojen sa slike. Posljednja faza obuhvaća prosljeđivanje izdvojenih znakova algoritmu za treniranje i prepoznavanje korištenjem metode potpornih vektora.

Rad je strukturno podijeljen u četiri dijela. Prvo poglavlje označeno je predstavljanjem teme rada i tehnologija koje će se obraditi. Nadalje, drugo poglavlje obuhvaća teorijsku podlogu i detaljan opis prethodno navedenih faza razvijanja rada. Zatim, u trećem poglavlju, u središte pozornosti bit će stavljena izvedba aplikacije te će se detaljnije predstaviti rješenje izvedbe sustava za prepoznavanje znakova ograničenja brzine. Naposljetku, četvrto poglavlje govorit će o preciznosti rada aplikacije i graničnim uvjetima u kojima dolazi do smetnji prilikom rada iste te će se prikazati programsko rješenje prepoznavanja prometnih znakova ograničenja brzine.

1.1. Zadatak završnog rada

Implementacija algoritma za prepoznavanje znakova ograničenja brzine koristeći postojeće baze podataka zadatak je ovog rada. Pritom je potrebno koristiti preliminarnu obradu slika uporabom postojećih funkcija iz odgovarajućih biblioteka, kao što je Emgu CV. Broj sa slike potrebno je izdvojiti i prepoznati pomoću alata za detekciju znaka i metode potpornih vektora.

2. PROBLEMATIKA DETEKCIJE I PREPOZNAVANJA PROMETNIH ZNAKOVA

U skladu sa zahtjevima zadatka, potrebno je odabrati slike s kojih se želi razaznati traženi prometni znak ograničenja brzine. Obzirom da se slike mogu pojaviti u različitim rezolucijama te sadržavati brojne smetnje ili oštećenja, jasno je da se iste, ponajprije, moraju izgladiti otklanjanjem nepravilnosti u što većem postotku, a zatim pokušati detektirati i prepoznati traženi objekt. Kako bi se to uspješno ostvarilo, koriste se: dvije metode preliminarne obrade slika, standardni medijan i Gaussov filtar, detekcija znaka po obliku primjenom konturiranja slike i SVM algoritam za prepoznavanje brojeva unutar znaka. U ovom poglavlju bit će sagledani opći podaci i matematički izrazi svake metode ponaosob.

2.1. Metode preliminarne obrade slika

Preliminarna obrada slika podrazumijeva korištenje različitih filtera kojima se obrađuje zadana slika kako bi se postigli precizniji rezultati prilikom detekcije i prepoznavanja prometnog znaka, odnosno kako bi se postigla kvalitetnija slika uklanjanjem impulsnog šuma, iz koje je, zatim, lakše primijetiti i izdvojiti ključne značajke znaka. Impulsni šum može se razaznati u slici kao piksel koji intenzitetom vlastite svjetlosti čini kontrast susjednim pikselma. Sukladno tome, rezultat primjene filtera za zaglađivanje na neku sliku podrazumijeva promjenu intenziteta samih piksela slike, pri čemu ne dolazi do promjene položaja piksela u prostoru. Prema [1], pikselima se povećava intenzitet kako bi se naglasile rubne granice unutar slike ili, suprotno, intenzitet određenih piksela se smanjuje kako bi se rubne granice izgladile. Obzirom da postoje razne metode kojima se mogu filtrirati slike, za potrebe izvedbe aplikacije, zbog učinkovitosti i opće raširenosti uporabe, odabrani su standardni medijan filtar i Gaussov filtar. Također, kombinacija ovih dvaju filtera omogućava efektivno uklanjanje šuma unutar slike uzimajući u obzir sve piksele i to izračunavanjem medijana (Standardni medijan filtar) i srednje vrijednosti (Gaussov filtar) istih.

2.1.1. Standardni medijan filter

Standardni medijan filter koristi se za zaglađivanje slike jer omogućava otklanjanje šuma u slici analizirajući svaki piksel slike uspoređujući pritom isti sa susjednim pikselima. Ukoliko se promatrani piksel etiketira kao onaj koji je u kontrastu s okolinom, zamijenit će se medijanom svih susjednih piksela. Kontrastni pikseli se, u slici, često razaznaju kao crne točke, tako zvani papar šum, koje zapravo imaju vrijednost piksela u nuli, ili kao bijele točke, tako zvana sol u slici, koje sadrže najveću moguću vrijednost piksela, odnosno 255., što je i opisano u [3]. Kako je navedeno

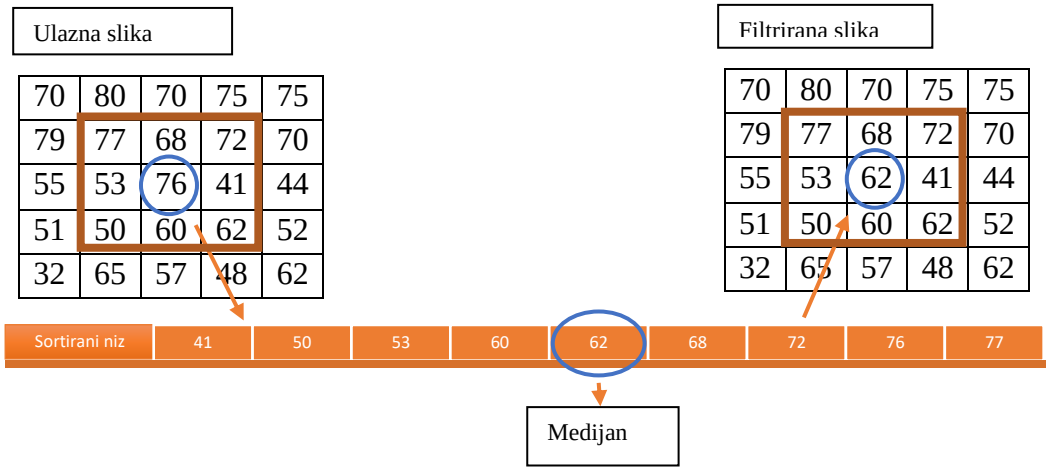
u [3], ovaj postupak primjenjuje se na cjelokupnu sliku, uključujući i one dijelove koji nisu zahvaćeni djelovanjem impulsnog šuma. Tako na primjer, može doći do zaglađivanja rubnih dijelova slike, odnosno onih dijelova koji odvajaju cjeline u slici, a koji su ključni za detekciju slike.

Slika filtrirana na ovaj način može se opisati sljedećim izrazom, navedenim u [3]:

$$F(i,j) = median_{(k,l)\in O_{d,v}}\{D(i+k, j+l)\} \qquad (2-1)$$

gdje $O_{d,v}$ predstavlja promjenjivi okvir unutar slike veličine (d*v) piksela centriran u točki s (i, j) koordinatama.

Prema [3], računanje medijana vrši se nad uzlazno sortiranim nizom vrijednosti piksela, koji se nalaze u odabranom okviru. Ukoliko se u okolini promatrane točke nalazi neparan broj susjeda, medijan se računa kao aritmetička sredina dviju srednjih vrijednosti spomenutog niza. Nasuprot navedenom, ako je riječ o parnom broju susjeda, medijan će se ostvariti kao središnja vrijednost sortiranog niza. Primjer računanja medijana nad određenim pikselom za filtar veličine 3x3, za slučaj neparnog broja piksela unutar okvira, može se vidjeti na slici 2.1.



Slika 2.1. Prikaz postupka izračunavanja medijana nad određenim pikselom slike

2.1.2. Gaussov filtar

Govoreći o odstranjivanju šuma u digitalnoj slici, svakako, treba spomenuti Gaussov filtar kao jedan od najraširenijih. Prema [4], riječ je o dvodimenzionalnom konvolucijskom niskopropusnom

filtru, koji se temelji na izračunavanju srednje vrijednosti piksela koristeći posebno određenu jezgru. U ovom slučaju, jezgra označava manju matricu brojeva, koji predstavljaju vrijednost piksela, a koristi se prilikom primjene konvolucije na sliku, što je i opisano u [4]. Izračun srednje vrijednosti unutar jezgre filtra započinje množenjem vrijednosti intenziteta svih piksela u slici. Dobivenom rezultatu se, zatim, pridodaju vrijednosti položaja svakog piksela, tako zvane težišne vrijednosti, koordinata (x, y), koje se dobivaju u odnosu na središnji piksel, čije se koordinate vode kao ishodište jezgre filtra. Izraz koji se koristi za izračun težišnih vrijednosti jest Gaussova distribucija opisana u [5]:

$$G(x,y) = \frac{1}{2*\pi*\sigma^2} * e^{-\frac{x^2+y^2}{2*\sigma^2}} \tag{2-2}$$

gdje σ predstavlja standardnu devijaciju jezgre.
Prema (2-2), što je standardna devijacija jezgre veća, to će rezultat filtriranja, odnosno zaglađivanje, biti veće.

2.2. Detekcija objekta unutar slike

Detekcija objekta označava proces pronalaska instanci stvarnih i realnih objekata unutar slike ili videa. Provođenjem postupka detekcije rješava se problematika klasifikacije u pogledu razaznavanja različitih vrsta objekta u slici, kao na primjer ljudskih obrisa, lica, bicikala, prometnih znakova, i lokalizacije samog objekta, pri čemu se saznaje točna pozicija objekta unutar slike. Ove dvije karakteristike omogućavaju izdvajanje pronađenih značajki iz slike kao područje od interesa (eng. *Region of interest*, ROI), što je spomenuto u [6], čime se omogućava prilagodba slike za provođenje preciznog prepoznavanja samog objekta. Budući da je ovo jedno od najraširenijih područja istraživanja čija rješenja ruše brojne prepreke današnjice, posebice u medicini i razvoju autonomnih sustava, razvijene su mnoge metode za rješavanje ovog problema, a mogu se podijeliti u dvije glavne skupine, detekcija izdvajanjem značajki i detekcija dubokim učenjem. Pritom, detekcija dubokim učenjem podrazumijeva uporabu dubokih neuronskih mreža, čiji je razvoj u posljednjih nekoliko godina doživio popriličan zamah.

Detekcija provedena u ovom radu koristit će metode izdvajanja značajki iz slike. Elaborirat će se značajke znakova ograničenja brzine u hrvatskom sustavu znakovlja. Takav znak sastoji od dvaju koncentričnih kružnica te crveno ispunjenog kružnog vijenca između, unutar kojeg se nalaze crnim slovima ispisane brojke, koje predstavljaju ograničenje brzine. Na tragu toga, pozornost ove faze usmjerit će se na traženje kružnog oblika unutar slike koji sadrži brojke.

Metoda za prepoznavanje kružnih kontura u slici naziva se Houghova transformacija (eng. *Hough Transform*). Kako bi bilo moguće primijeniti ovu metodu, u slici je potrebno definirati sve rubne konture. Ovaj proces izvodi se uporabom Cannyjevog detektora, koji se provodi u četiri koraka, opisanih u [7]. Inače, Cannyjev detektor rubova je višefazni algoritam za detekciju rubova na crno-bijelim slikama uz uvjet redukcije šuma u slici. Dakle, najprije se eliminiraju smetnje u slici korištenjem Gaussovog filtra opisanog u prethodnom poglavlju. Nadalje, određuje se gradijent slike kako bi se naglasile regije uporabom Sobelovog operatora, algoritma koji mjeri dvodimenzionalni prostorni gradijent u slici naglašavajući pritom područja visoke frekvencije koja se smatraju rubovima. Zatim se zanemaruju svi pikseli koji nemaju maksimalnu vrijednost, a uporabom histereze izdvajaju se svi oni pikseli kojima je zabilježena najveća moguća vrijednost. Naposljetku, koristi se metoda dvostrukog praga (eng. *double tresholding method*) nad svakim pikselom kako bi se odredilo radi li se o rubnom pikselu. U tu svrhu određuju se dva praga, T_1 i T_2, da bi se pojedini gradijenti mogli točno klasificirati. U slučaju da je gradijent veći od praga T_2, piksel se definira kao rubni. Ako je gradijent manji od praga T_1, piksel se ne smatra rubom. Konačno, kada je vrijednost gradijent između dva praga, donosi se odluka na osnovu smjera piksela i raspoređenosti postojećih rubnih točaka.

Detekcija kružnice odvija se postupkom eksplicitnog glasanja, koje je objašnjeno u [6], a ostvaruje se korištenjem parametara određenih oblikom koji se traži. Parametri potrebni za definiranje kružnice su središte iste s koordinatama (x, y) i polumjer r, a jednadžba kružnice predstavljena je sljedećim izrazom:

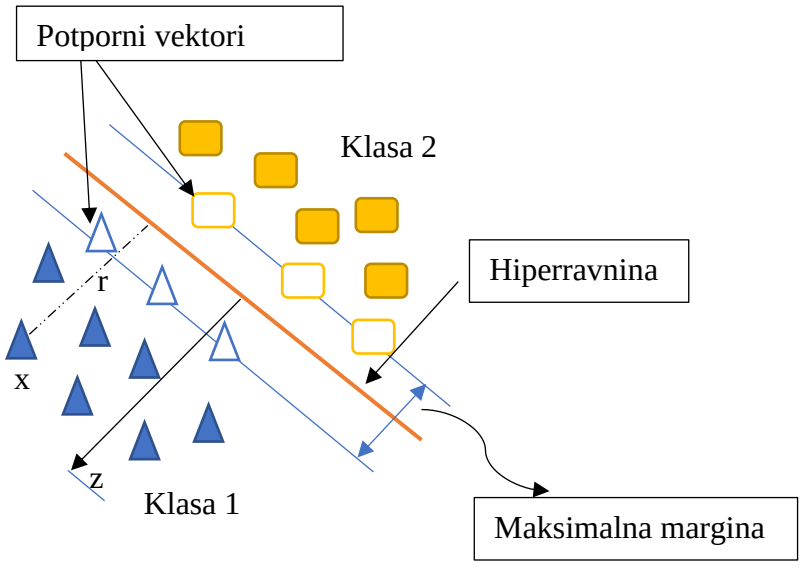
$$(p - x)^2 + (q - y)^2 = r^2 \tag{2-3}$$

Budući da je postupak orijentiran isključivo na traženje kružnice unutar zadane slike, svaki piksel glasat će za parametre kružnice kojoj može pripadati. Na temelju razrade u [6], zaključuje se da se ti glasovi međusobno pridružuju u trodimenzionalno akumulacijsko polje jer tri navedena parametra definiraju oblik objekta, a tražena kružnica ostvaruje se u maksimumima stvorenog polja.

2.3. Metoda potpornih vektora

Metoda potpornih vektora (eng. *Support Vector Machines*, SVM), naziva se još i metoda jezgrenih funkcija, a, kao što je opisano u [8], predstavlja tehnologiju strojnog učenja koja pronalazi

hiperravninu u svrhu određivanja udaljenosti između točaka najbližih istoj (eng. *decision boundary*) kako bi se mogle razdvojiti klase. Ta udaljenost naziva se margina, a pomoću SVM-a ona se pokušava maksimizirati oko plohe razdvajanja. Obzirom da je riječ o strojnom učenju, ova tehnologija ima određenu, tako zvanu, funkciju odluke, definiranu potpornim vektorima (eng. *support vectors*), koji su, zapravo, elementi podskupova skupa za učenje.



Slika 2.2. Prikaz margine i potpornih vektora oko ravnine razdvajanja

Ravnina razdvajanja opisuje se izrazom iz [8]:

$$\boldsymbol{z^T} * \boldsymbol{x} + l = 0 \tag{2-4}$$

gdje je $\boldsymbol{z^T}$ vektor normale na ravninu razdvajanja, $\boldsymbol{x}$ ulazni vektor, a l pomak.

Prilikom određivanja udaljenosti r između plohe razdvajanja i vektora $\boldsymbol{x}$ koristi se sljedeća relacija navedena u [8]:

$$r = \frac{\boldsymbol{z^T} * \boldsymbol{x} + l}{\|\boldsymbol{z}\|} \tag{2-5}$$

gdje je $\boldsymbol{z^T}$ normala na ravninu razdvajanja, $\|\boldsymbol{z^T}\|$ je euklidska norma iste, a $l/\|\boldsymbol{z^T}\|$ je okomita udaljenost ravnine razdvajanja do originala (ulaznog vektora).

Neka je skup $A=\{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_n}, y_n)\}$ skup ulaznih podataka, pri čemu vrijedi $y_i \epsilon \langle -1,1 \rangle$. Ako se pretpostavi da se svaki vektor $\boldsymbol{x_i}$ nalazi na udaljenosti 1 od plohe, vrijedit će sljedeći uvjeti, koji su opisani u [8]:

$$\boldsymbol{z^T} * \boldsymbol{x_i} + l \geq 1 \; ako \; je \; y_i = 1 \tag{2-6}$$

$$\boldsymbol{z^T} * \boldsymbol{x_i} + l \leq -1 \; ako \; je \; y_i = -1 \tag{2-7}$$

Izrazi (2-6) i (2-7) zajedno daju izraz spomenut u [9]:

$$y_i * (\boldsymbol{z^T} * \boldsymbol{x_i} + l) - 1 \geq 0, \forall i \tag{2-8}$$

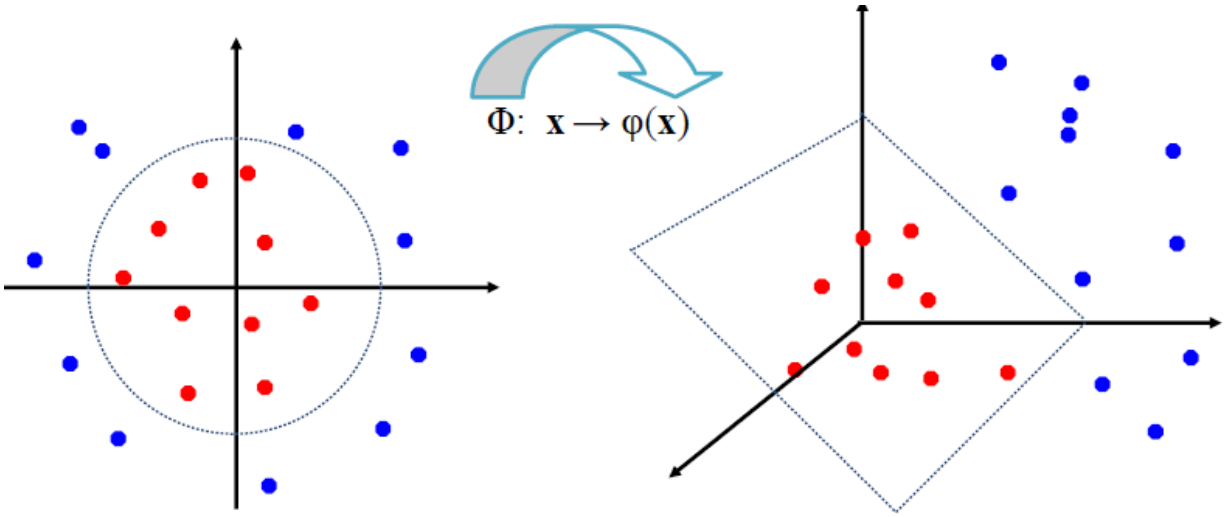
Budući da se želi pronaći ravnina razdvajanja s maksimalnom marginom, traži se, prema opisu u [9], $\min\|\boldsymbol{z}\|$ tako da vrijedi (2-8).

Ograničenja postavljena pod (2-6), (2-7) i (2-8) za ulazne podatke označavaju uvjete za ostvarivanje linearnog SVM-a sa razdvojivim klasama, što je spomenuto u [8]. U slučaju da je riječ o nerazdvojivim klasama, u navedene izraze uvodi se nenegativna konstanta ξ_i, te pronalazak ravnine razdvajanja poprima sljedeći oblik naveden u [9]:

$$\min \frac{1}{2} \|\boldsymbol{z}\|^2 + C \sum_{i=1}^{N} \xi_i \text{ tako da vrijedi } y_i * (\boldsymbol{z^T} * \boldsymbol{x_i} + l) - 1 + \xi_i \geq 0, \forall i \tag{2-9}$$

gdje je *C* faktor pogreške koje se pojavljuju prilikom treniranja ulaznih podataka.

Kada se govori o nelinearnom SVM-u, valja napomenuti da se uglavnom primjenjuje prilikom klasificiranja podataka koji su nelinearni, nerazdvojivi i koji imaju veću količinu šuma za odstraniti, čime se otežava pronalazak ravnine razdvajanja. Prema Coverovom teoremu, razjašnjenom u [10], prelazak u višedimenzionalni prostor povećava vjerojatnost pojave linearne razdvojivosti. U tom slučaju, glavni zadatak jest vrijednosti, koje se nalaze oko ravnine razdvajanja, smjestiti u višedimenzionalni prostor kako bi se stvorio razdvojiv sustav. Ovaj postupak prikazan je slikom 2.3.



Slika 2.3. Grafički prikaz nelinearnog preslikavanja, iz [8]

Rezultat prema slici 2.3. ostvaruje se primjenom tako zvanog Jezgrenog trika (eng. *Kernel Trick*). Prema [9], matematički zapis istog postiže se zamjenom $\boldsymbol{x_i}$ vektora s funkcijom $\Phi(\boldsymbol{x_i})^T$ u jednadžbi (2-9). Pritom, jednadžba ravnine razdvajanja za m-dimenzionalni prostor poprima sljedeći oblik, opisan u [10]:

$$\sum_{j=1}^{M} \boldsymbol{z_j}\, \Phi_j(\boldsymbol{x}) + l = 0 \tag{2-10}$$

Kako bi se olakšalo računanje umnoška zamijenjene funkcije i vektora normale $\boldsymbol{z}$, koji se, prema [10], u svrhu optimizacije tretira Langrangeovim multiplikatorom α_i (2-11), uspostavlja se jezgrena funkcija (eng. *Kernel function*) (2-12):

$$\boldsymbol{z} = \sum_{i=1}^{N} \alpha_i\, y_i \Phi_i(\boldsymbol{x}) \tag{2-11}$$

$$J(\boldsymbol{x_i}, \boldsymbol{x_j}) = \Phi(\boldsymbol{x_i})^T * \Phi(\boldsymbol{x_j}) \tag{2-12}$$

Prema [9], postoji nekoliko vrsta jezgrenih funkcija, od kojih je najraširenija Gaussova jezgrena funkcija (eng. *Radial Basis Function*, RBF), dana izrazom:

$$J(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{-\delta \|x_i - x_j\|^2},\ \delta > 0 \tag{2-13}$$

gdje je δ veličina koja određuje koliko će Gaussova krivulja biti široka.

Zbog prirode slika koje pripadaju setu za treniranje, za potrebe implementacije algoritma za primjenu metode potpornih vektora, koristit će se Gaussova jezgra, čija će detaljna izvedba, to jest korišteni parametri za ostvarivanje navedenih matematičkih zakona, biti opisana u narednom poglavlju.

3. REALIZACIJA ALGORITMA ZA PREPOZNAVANJE ZNAKOVA OGRANIČENJA BRZINE

Implementacija algoritma u C#-u zahtijeva uporabu biblioteka i funkcija koje će ostvariti učinak svih metoda opisanih u prethodnom poglavlju. Ponajprije, koristi se već ranije spomenuta Emgu CV biblioteka. Inače, Emgu CV je inačica Open CV biblioteke prilagođena radu u *.NET* programskom okruženju uglavnom namijenjena za primjenu računalnog vida u stvarnom vremenu te sadrži sve funkcije koje će biti potrebne za implementaciju algoritma, a to su*: MedianSmooth, GausianSmooth, Canny, HoughCircles, ROI*. Pritom, pozornost treba obratiti na parametre funkcije, te ih podesiti u skladu sa slikama kojima se želi manipulirati. Također, Emgu CV sadrži više biblioteka strojnog učenja, a jedna od navedenih je potrebna SVM biblioteka. Sukladno tome, ovo poglavlje bavit će se problematikom određivanja prikladnih uvjeta unutar algoritma za postizanje željenih rezultata u odnosu na odabranu bazu slika.

3.1. Baza podataka

Budući da priroda ovoga rada iziskuje implementaciju algoritma za prepoznavanje znakova ograničenja brzine, potrebno je odrediti bazu slika na kojima se nalaze traženi objekti. Postoji nekoliko vrsta postojećih baza posebno poznatih i korištenih u području detekcije i prepoznavanja prometnih znakova za europsko područje, između kojih se ističu: Njemački standard prepoznavanja prometnih znakova (eng. *The German Traffic Sign Recognition Benchmark*, GTSRB) s više od 40 klasa i preko 50 000 slika te Njemački standard detekcije prometnih znakova (eng. *The German Traffic Sign Detection Benchmark*, GTSDB) s preko 900 slika. Ipak, za izvedbu ove aplikacija odabrana je tako zvana rMASTIF baza. Riječ je o klasificiranoj hrvatskoj bazi prometnih znakova nastaloj tijekom istraživačkog projekta MASTIF i moguće ju je koristiti u akademske svrhe. Glavna prednost ove baze jest što sadrži realne slike prometnih znakova s hrvatskih prometnica. Uz navedeno, postoji minimalno dvanaest različitih slika pojedinog znaka u dijelu baze za trening algoritma, koje su snimljene u različitim uvjetima osvjetljenja i okoliša te pod drugačijim kutom. Uz navedeno, svaka slika zasebno dolazi u nekoliko različitih veličina i rezolucija, što na kraju čini čitavu aplikaciju vjerodostojnijom i prilagođenom različitim uvjetima i kutovima slikanja.

Za potrebe realizacije projekta koristit će se klasificirane slike o prometnim znakovima ograničenja brzine iz navedene baze. Postoje dva glavna dijela baze, dio za trening i testni dio.

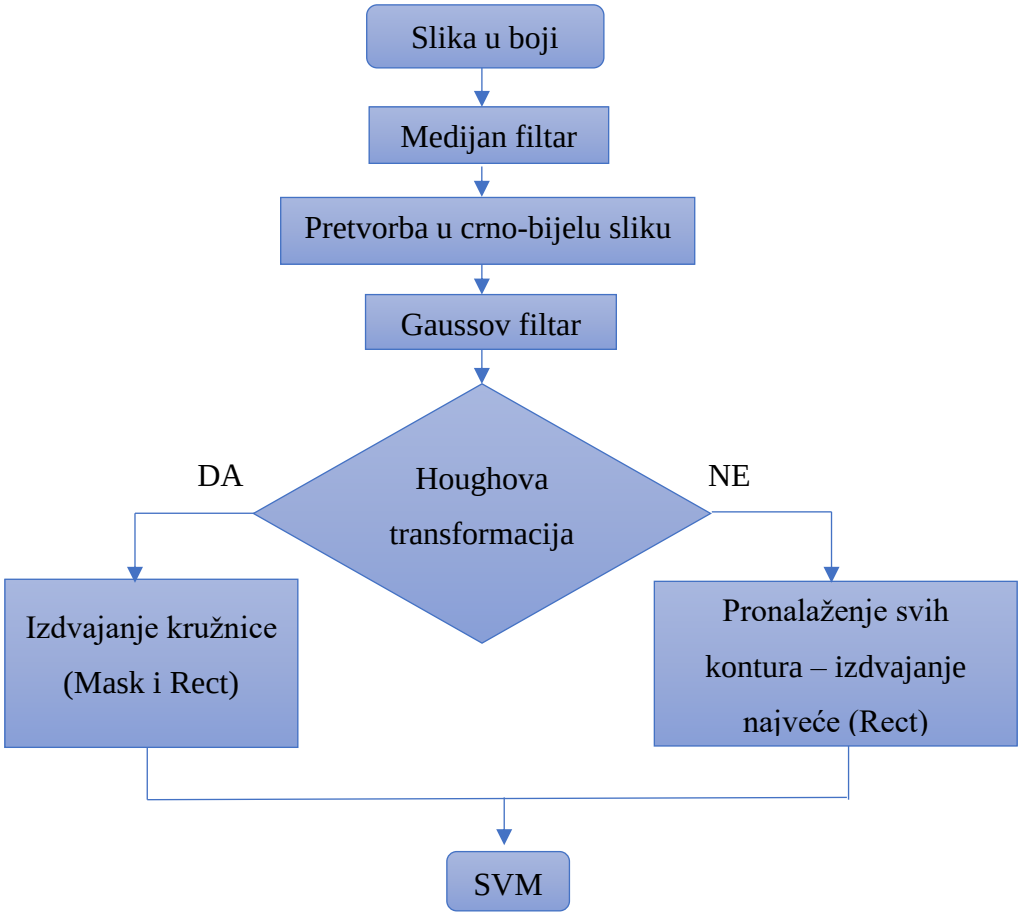
Dakako, slike u setu za trening i testiranje su različite. U svrhu uspostavljanja granica između klasa primjenom metode potpornih vektora, koristit će se set slika za trening, koji se sastoji od ukupno 280 slika. Prometni znakovi ograničenja brzine podijeljeni su na pet klasa. Svaka klasa predstavlja jednu vrstu znaka ograničenja brzine, a to su redom brzine: 30, 40, 50, 60 i 70 km/h.
Testni dio baze koncipiran je na isti način, a koristi se za provjeru algoritma za prepoznavanje te sadrži 192 slike.

Slika 3.1. Prikaz primjera unutar četiri različite klase rMASTIF baze podataka (set za testiranje)

Osim navedenog, za testiranje algoritma koristi se i pristup kameri računala kako bi se omogućilo prepoznavanje u stvarnom vremenu na ispisanim znakovima s bijelom podlogom.

3.2. Dijagram toka algoritma

Na slici 3.2. prikazan je dijagram toka izrade algoritma aplikacije za prepoznavanje znakova ograničenja brzine, na kojem se nalazi slijed svih postupaka obrade slika i detekcije kružnih objekata unutar istih kako bi ih sustav mogao uspješno naučiti i prepoznati korištenjem metode potpornih vektora.



Slika 3.2. Dijagram toka algoritma za detekciju i prepoznavanje znakova ograničenja brzine

Prema slici 3.2., nakon uporabe Gaussovog filtra, pokušat će se prepoznati kružne konture unutar slike uporabom funkcije *HoughCircles*. Međutim, ako funkcija u slici ne bude mogla razaznati kružnice, koristit će se funkcija *FindCountours* koja će pronaći sve konture unutar iste i izdvojiti najveću. Nakon izdvajanja kontura slijedi primjena SVM-a.

3.3. Funkcije za preliminarnu obradu podataka

Kao što se može vidjeti na slici 3.1., određene slike odabrane baze sadrže u sebi popriličan broj smetnji. Kako bi se te smetnje odstranile u što je moguće većem broju, koriste se tehnike filtriranja i zaglađivanja slika. Zbog toga su se, prilikom implementacije algoritma, koristile dvije funkcije, čiji rezultati odgovaraju željenim ishodima preliminarnih metoda opisanih u Poglavlju 2.

3.3.1. Funkcije medijan filtra

Ponajprije, koristi se funkcija *SmoothMedian()*, koja u fazi realizacije zadatka obnaša djelovanje medijan filtra. Funkcija *SmoothMedian()* poziva se kao filtar na odabranu sliku u boji te se

obrađena slika sprema kao novi objekt pogodan za daljnju manipulaciju. Korištena funkcija prima jedan 32-bitni neparan cijeli broj, koji predstavlja veličinu (visinu i dužinu) prozora unutar kojeg se redom, nad svakim pikselom, izvodi medijan filtar. Eksperimentalno je utvrđeno kako se generalno za rMASTIF bazu dobivaju optimalni rezultati kada taj cijeli broj ne prelazi pet. Pod pojmom optimalni rezultati, osim vidljivog izgleda slike, podrazumijeva se ispravan rad daljnjih funkcija, koje se koriste u radu.

a) b) c) d)

Slika 3.3. Usporedba različitih veličina prozora za primjenu medijan filtera (veličina svih slika 100x100 piksela): veličina prozora vrijednosti 3 (b), veličina prozora vrijednosti 5 (c), veličina prozora vrijednosti 7 (d) u odnosu na originalnu sliku (a)

Promatrajući djelovanje medijan filtra u odnosu na različite parametre, uviđa se kako se slika sa problematičnim rubnim dijelovima najbolje izgladi kada se prozor djelovanja medijan filtera postavi na vrijednost pet, a kao što je spomenuto dodatno se zamućuje uporabom veličine prozora vrijednosti sedam nadalje. Sukladno tome, preporuka je ostvarivanje medijan filtra veličinom prozora vrijednosti pet za cijelu bazu.

Pritom, svakako valja napomenuti da se sve slike prije početka obrade postavljaju na istu veličinu, to jest 100 piksela po dužini i širini. Razlog tomu je taj što su sve slike u bazi različite veličine, a postavljanjem fiksnih parametara funkcija uzrokuje drugačije djelovanje na svaku sliku ponaosob, što je vidljivo na slici 3.4. Pritom su ponajbolji rezultati zamijećeni uz veličinu slike 100x100 piksela i fiksnu veličinu prozora medijan filtera vrijednosti 5. U slučaju da se promjeni veličina slike na primjerice 50 piksela po dužini i širini, funkcija rezultira dodatnim zamućenjem slike, čime se onemogućava ispravan rad narednih funkcija. (Sl. 3.4.)

a) b) c)

Slika 3.4. Usporedba originalne slike (a) i filtriranih slika veličina: 100x100 piksela (b), 50x50 piksela(c), uz veličinu prozora medijan filtra iznosa 5

Također, postoji drugi način poziva funkcije korištenjem klase CvInvoke implementirane unutar Emgu CV-ja. Tada se poziva funkcija pod nazivom *MedianBlur()* koja prima tri parametra: ulaznu sliku, sliku na koju se ispisuje rezultat djelovanja funkcije te veličinu jezgre, zapisanu kao 32-bitni cijeli broj, koji mora biti neparan i veći od jedan.

a) b) c)

Slika 3.5. Usporedba djelovanja medijan filtra (veličina prozora medijan filtra iznosa 5) na originalnu sliku (a) različitim pozivom metoda: SmoothMedian funkcija (b) i MedianBlur funkcija (c)

Bez obzira koji poziv funkcija se koristio, rezultat provedbe filtra će za isti parametar biti jednak. Jedan od razloga zbog kojih je dobro usporediti funkcije različitom izvedbom sintakse jest provjera ispravnosti vlastitog koda. U slučaju, da su se, za medijan filtar parametara pet, dobili različiti rezultati to bi označavalo pogrešku programera prilikom implementacije algoritma. Nakon što se utvrdila ispravnost uporabe ove funkcije, moguće je filtriranu sliku proslijediti na daljnju obradu.

3.3.2. Funkcije Gaussovog filtra

Primjena Gaussovog zaglađivanja zahtijeva prethodnu pretvorbu slike u boji u crno-bijelu sliku. Na tako pripremljenu sliku poziva se *SmoothGaussian()* funkcija. Pritom, uvijek je dobro definirati novi objekt za svaku promjenu slike kako ne bi došlo do gubitka podataka izvorne slike. Ova funkcija zahtijeva poznavanje nekoliko parametara opisanih u [11, str. 112]. Ponajprije, prva dva

parametra su cijeli 32-bitni neparni brojevi veći od 1, ne nužno jednaki, koji redom predstavljaju dužinu i visinu jezgre, odnosno matrice unutar slike koja se promatra. Veličina iste ovisi o veličini same slike jer se uzimaju u obzir svi pikseli slike, te je potrebno prilagoditi parametre jezgre za dobivanje željenog učinka. Druga dva parametra koji promatrana funkcija prima su standardne devijacije jezgre u vodoravnom i okomitom smjeru, odnosno standardne devijacije u odnosu na x-os i na y-os jezgre, koje su potrebne za izračun Gaussove dvodimenzionalne distribucije jezgre.

U slučaju da se veličina jezgre postavi u nulu, ista će biti određena obzirom na standardnu devijaciju jegre. Vrijedi i obrnuto, ukoliko se odredi željena veličina jezgre i postave obje standardne devijacije u nulu, iste će biti određene u odnosu na parametre jezgre. Upravo navedeni način parametriranja ove funkcije bit će korišten u svrhu implementacije algoritma. Budući da su sve slike oblikovane na istu veličinu, bit će određeni reprezentativni primjeri usporedbe djelovanja različitih veličina jezgre na sliku. (Sl. 3.6.)

Kao što je već spomenuto, ovo filtriranje izvodi se na slici obrađenoj medijan filtrom te se bira onaj parametar koji će najbolje surađivati s prethodnom obradom, ali i narednim funkcijama. Prema [11, str. 113], izvedba ove funkcije optimizirana je unutar biblioteke za tri često korištene jezgre, a to su 3x3, 5x5 i 7x7, stoga će se njihov učinak na zadanu bazu razmatrati u daljnjem razvitku teksta.

a) b) c)

Slika 3.6. Usporedba primjene Gaussovog filtra na dva različita tipa slika po veličini jezgre: a) Jezgra veličine 3x3, b) Jezgra veličine 5x5, c) Jezgra veličine 7x7

Prema slici 3.6. vidi se da su slike najjasnije uporabom 3x3 filtera. Povećanjem filtera povećava se zamućenost slike, odnosno pojačava se djelovanje filtra, te je time dokazan zaključak donesen u drugom poglavlju na temelju relacije (2-2). Sukladno navedenom, u ovom radu upotrebljavat će se Gaussova jezgre dužine i visine tri piksela.

3.4. Primjena Houghove transformacije

Budući da se funkcija *HoughCircles()* za traženje kružnica u slici zasniva na *Canny()* funkciji, najprije će se pojasniti djelovanje te funkcije i kako je implementirana u *HoughCircles()* funkciju. Pozivom *Canny()* funkcije na filtriranu sliku treba obratiti pažnju na dva parametra o kojima funkcija ovisi. Riječ je o višem i nižem pragu (eng. *higher and upper threshold*) intenziteta crne i bijele boje u crno-bijeloj slici. Prema tome, kao što je navedeno u [11, str. 152], ako je intenzitet određenog piksela veći od gornjeg praga, taj piksel se smatra rubom. Nasuprot toga, ako je intenzitet piksela ispod donje postavljene granice Cannyjevog detektora, piksel se ne smatra rubom u slici. U slučaju da se intenzitet piksela nalazi između te dvije vrijednosti, smatrat će se rubom samo ako se nalazi pokraj piksela koji je iznad gornjeg praga po intenzitetu. Također, prema [11, str. 152], preporuča se odnos višeg i nižeg praga kao 2:1 ili 3:1.

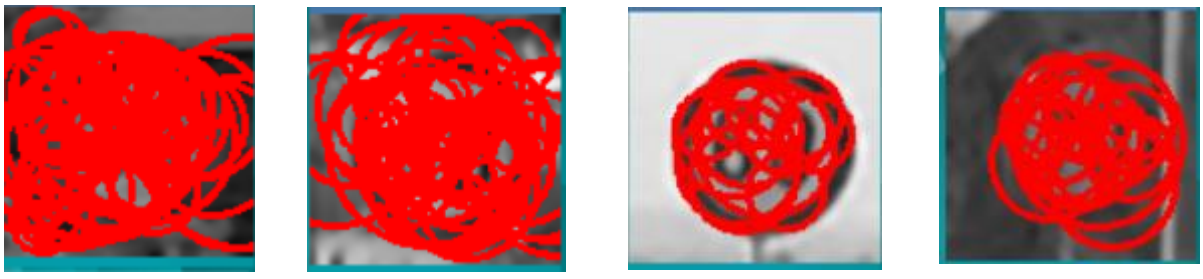
a) b) c) d)

Slika 3.7. Prikaz djelovanja Cannyjeve funkcije za različite kombinacije pragova (viši prag, niži prag): a) (50, 10), b) (100, 50), c) (150, 75), d) (200, 100)

Prema slici 3.7. zaključuje se da se konture obaju kružnica najbolje ističu prilikom uporabe kombinacije vrijednosti 100 za viši prag i vrijednosti 50 za niži prag unutar *Canny()* funkcije, stoga će se u funkciji *HoughCircles()* za viši prag uzeti upravo 100. To je ujedno prvi parametar koji navedena funkcija zahtijeva. Niži prag je definiran unutar same funkcije kao dvostruko manji od višeg. Osim toga, potrebno je unijeti vrijednosti praga akumulatora (eng. *accumulator threshold*), koji označava koliko je minimalno točaka potrebno akumulirati da bi se objekt smatrao kružnicom. Što je taj prag akumulatora manji detektirat će se više lažnih kružnica, odnosno što je veći bit će generalno detektirano manje kružnica. Općenito, za potrebe ovog zadatka postavljen je prag

akumulatora vrijednosti 35. Nadalje, određuje se rezolucija akumulatora, koja je postavljena u 1.9, što znači da će akumulator imati gotovo dvostruko (1.9 puta) manju visinu i širinu od slike. Zatim se određuje minimalna udaljenost između središta dva detektirana kruga, koja je postavljena na vrijednost 5, jer, ako je ta udaljenost premala, detektira se puno manjih lažnih kružnica, a ako je prevelika može se zaobići detekcija tražene kružnice. Naposljetku, unose se minimalan i maksimalan polumjer kružnice koje se traže. Ako se te dvije vrijednosti postave u nulu, detektirat će se sve moguće kružnice te će u konačnici rezultat prepoznavanja biti znatno manje točan. Uzimajući u obzir navedeno, u ovom slučaju, eksperimentalno je utvrđeno kako, generalno, za trening dio baze, vrijednosti 10 za minimalan i 50 za maksimalan polumjer, uz minimalan polumjer 10 i maksimalan polumjer 80 za testni dio baze, daju optimalne rezultate prilikom testiranja čitave baze jer je razlika između radijusa dovoljna za detekciju kružnica različitih veličina, što je vrlo važno obzirom da se slike međusobno razlikuju u odnosu na veličinu, kut snimanja te položaj znaka unutar slike. Također, pri uporabi navedenih vrijednosti dolazi do detekcije više kružnica, međutim, naposljetku se izdvaja samo jedna unutar koje se nalazi detektirani prometni znak. (Sl. 3.8. a) Ipak, postoje brojne slike u bazi nad kojima se može točno detektirati znak koristeći vrijednost 30 za minimalan i 35 za maksimalan polumjer kako bi se pokazala detekcija kruga koristeći samo jednu kružnicu, ali uz navedene polumjere krajnji rezultat testiranja provedenog nad svim slikama testne baze približno je 10% lošiji u odnosu na vrijednosti 10 za minimalan i 50 za maksimalan polumjer. (Sl. 3.8. b)

U izrađenoj aplikaciji, funkcija se izvodi nad kopijom slike koja je prošla kroz sve filtre te se kao rezultat vraća obrađena crno-bijela slika, na kojoj su detektirane kružnice označene crvenom bojom (Sl. 3.8.)



a)

b)

Slika 3.8. Prikaz nekoliko primjera rada funkcije HoughCircles a) pri korištenju vrijednosti 10 za minimalan polumjer i 50 za maksimalan polumjer, b) pri korištenju vrijednosti 30 za minimalan polumjer i 35 za maksimalan polumjer

Nakon što se detektira krug, potrebno ga je izdvojiti i stvoriti masku, odnosno zacrniti područje koje nije prometni znak. Za to je iz detektiranog kruga potrebno izdvojiti središte i polumjer. Nakon toga stvara se nova slika koja zacrnjuje sve osim područja u rasponu predanog joj polumjera i središta detektiranog kruga. Pomoću funkcije *And* u novu sliku preslikava se maska i sadržaj kruga iz filtrirane slike. (Sl. 3.9, a) Zatim se pomoću funkcije *Rect* označava područje interesa, koje se izrezuje i stvara se nova slika koja sadrži znamenke prometnog znaka. (Sl. 3.9, b) Znamenke se prosljeđuju algoritmu za prepoznavanje.

a)

b)

Slika 3.9. Prikaz izdvajanja prometnog znaka ograničenja brzine: a) Rezultat funkcije *And*, spoj maske i sadržaja znaka, b) Područje interesa

3.5. Prepoznavanje znamenki pomoću SVM-a

Proces prepoznavanja znamenki korištenjem SVM-a sastoji se od nekoliko faza, iako, ističu se dvije osnovne, trening i testiranje podataka. No, prije puštanja u pogon tih faza, potrebno je najprije ispravno učitati podatke. Podaci se učitavaju kao cjelovite slike u prirodnom okruženju, stoga ih je potrebno redom filtrirati i odvojiti značajke koje predstavljaju brojeve kako bi se mogla izvršiti klasifikacija. Budući da funkcija prima isključivo nizove kao parametre, potrebno je izdvojene brojke pretvoriti u nizove. Obzirom da se slike predstavljaju dvodimenzionalnim matricama, gdje svaki element matrice predstavlja intenzitet pojedinog piksela u slici, nužno ih je pretvoriti u jednodimenzionalan niz (vektor) jer je potrebno stvoriti jednu veliku matricu, u kojoj su sadržane sve slike iz seta za trening. Pritom, svaki red matrice predstavlja jednu sliku. Na taj način će se, prilikom uporabe SVM-a, svaka slika moći redom učitavati kao vektor (red iz matrice) i na osnovu istih, SVM će učiti obilježja slika svake klase. Na tragu toga, potrebno je redom bilježiti koja slika pripada kojem prometnom znaku, odnosno nužno je odrediti oznake klasa. Ti podaci se spremaju u posebnu jednodimenzionalnu matricu i, prilikom procesa učenja, isti se pridružuju odgovarajućoj slici. Nad tako pripremljenim matricama, moguće je ostvariti poziv i klasifikaciju uporabom SVM-a, za što se koristi nekoliko funkcija unutar klase SVM.

Za potrebe treniranja SVM-a, najprije se stvara novi objekt tipa *SVM* u istoimenoj klasi, a zatim se određuju pojedini parametri za novi objekt. Ti parametri ovise o podacima koji se koriste, odnosno o bazi slika. Parametar *C*, kao što je navedeno u [12, str. 40], predstavlja veličinu koja kompenzira pogreške nastale prilikom treninga slika i složenost modela (određivanje maksimalne margine). Drugim riječima, što je *C* manji bit će više pogrešaka u treningu podataka i može doći do podbačaja (eng. *underfitting*). Suprotno tome, što je *C* veći, znači da SVM uglavnom odgovara uzorcima u setu za trening. Međutim, ako je isti prevelik, teško je prilagoditi trenirani model s jako malom greškom testnim uzorcima, što na kraju uzrokuje veliku testnu pogrešku. Ova pojava naziva se prekomjerna specijalizacija problema ili prebačaj (eng. *overfitting*). Za potrebe izvedbe ovog rada izabrano jest da ovaj parametar iznosi 1000 jer se, pri odabiru vrijednosti 1000, u kombinaciji sa ostalim vrijednostima parametara dobivaju prihvatljivi rezultati, odnosno točnost SVM-a prelazi 79% za tri od ukupno pet treniranih klasa.

Isto tako, potrebno je odrediti tip SVM-a koji se želi upotrijebiti. Odabran je *C_SVC* tip koji se općenito koristi za klasifikaciju više od dviju klasa, a odlikuje se sposobnošću rada s nelinearnim i nerazdvojivim ulaznim podacima. Sljedeći važan postupak jest određivanje vrste jezgre koja se

koristi. Ovaj odabir određuje način na koji će se rasporediti ulazni podaci kako bi postali linearno razdvojivi. Drugim riječima, bira se jezgrena funkcija koja će djelovati na ulazni set podataka. Obzirom da su podaci baze blago nelinearno raspoređeni, odabrat će se *Rbf* tip SVM-a, koji predstavlja Gaussovu jezgru, objašnjenu u Poglavlju 2. Prije poziva funkcije *Train* za treniranje seta, potrebno je još odabrati kriterij prekida algoritma. Prema [13], u ovom dijelu određuje se broj dopuštenog ponavljanja izvođenja SVM-a, a to je u ovom slučaju 1000, i toleranciju pogreške kako bi algoritam mogao završiti radnju u manjem broju koraka. Tolerancija pogreške postavljena je na 1e-6. Uporabom tako određenih funkcija, set ulaznih podataka je uspješno klasificiran i istreniran.

Zaključno, koristi se funkcija *Predict,* koja predviđa odgovor na dani uzorak. Ova funkcija se rabi u svrhu testiranja SVM-a. Testiranje se provodi nad slikama iz seta za testiranje rMASTIF baze. Također, kao što je slučaj s pripremom slika za učenje, potrebno je odabranu sliku pretvoriti u vektor. U ovom radu to se ostvaruje korištenjem, funkcije *HOGDescriptor,* koja preuzima koordinate svakog piksela slike i zapisuje ih u obliku točaka kako bi ih, naposljetku, mogla pohraniti u niz. Ako se testiranje vrši nad više slika, dohvaćaju se redovi matrice jedan po jedan i nad svakim zasebno se vrši predviđanje. Pritom, u slučaju da postoji matrica oznaka testne baze, ista se uspoređuje s dobivenim rezultatom te se, na osnovu točnih predviđanja i ukupnog broja testiranih slika, može dobiti postotak točnosti algoritma. Rezultati testiranja SVM-a bit će razrađeni u sljedećem poglavlju.

4. TOČNOST I OGRANIČENJA ALGORITMA ZA PREPOZNAVANJE ZNAKOVA OGRANIČENJA BRZINE

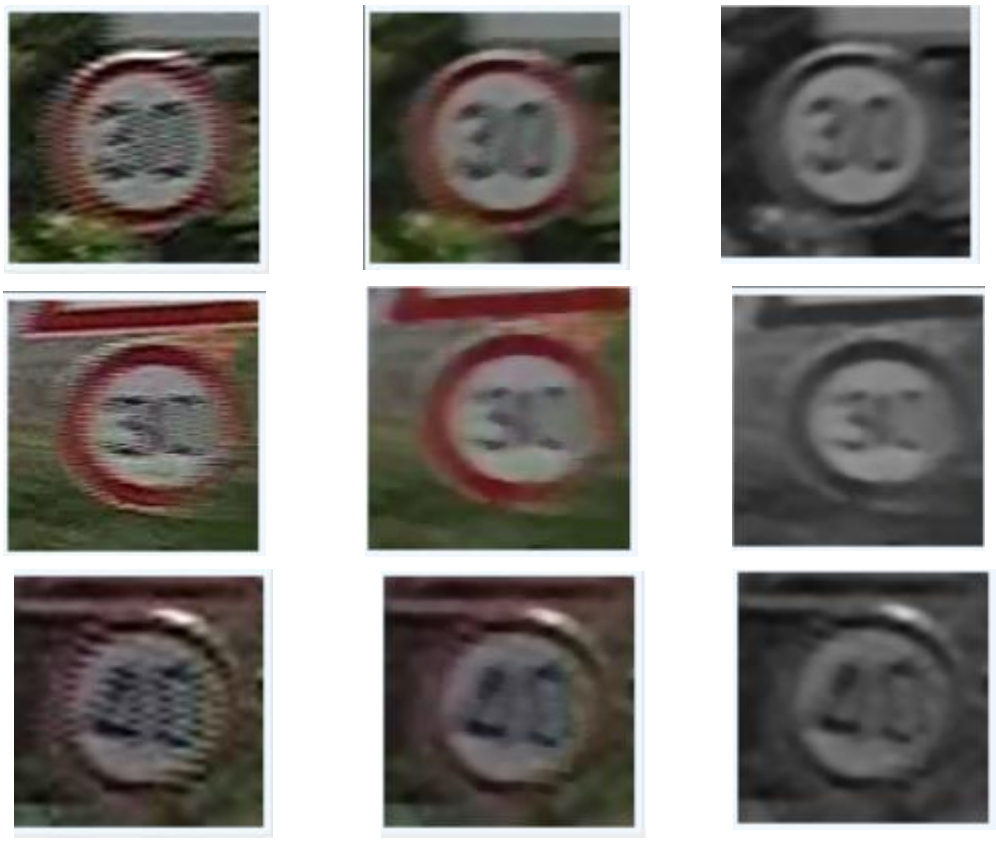
Kako bi se utvrdila ispravnost prepoznavanja znaka, potrebno je najprije ispitati krajnje granice prethodno uporabljenih funkcija. Unatoč odabiru povoljnih parametara za korištenu bazu, postoje određene situacije u kojima jedna ili više funkcija algoritma ne daje očekivani rezultat. Širok je raspon razloga zašto je tomu tako, a valja istaknuti lošu kvalitetu slike, kao i sličnosti između brojki (na primjer brojka pet i šest su dosta slične jer imaju kružni dio na istom mjestu te može doći do miješanja istih prilikom prepoznavanja). U slučaju slike lošije kvalitete, često izostaje točan rezultat prilikom obrade iste, što u konačnici može rezultirati time da više slika iz različitih klasa nose slična obilježja (kao vektori matrice za trening i testiranje) te dolazi do pogreške prilikom uporabe SVM-a za prepoznavanje znakova ograničenja brzine. Zbog navedenog će se ovo poglavlje baviti upravo tim graničnim slučajevima i utvrđivanjem izvora nastale pogreške.

a) b)

Slika 4.1. Prikaz djelovanja medijan filtera na više različitih primjera (b) u odnosu na originalne slike (a)

Kao što je vidljivo na slici 4.1., medijan filtar uspješno otklanja većinu vidljivog šuma na slikama snimljenima u normalnim uvjetima. Pod pojmom normalni uvjeti podrazumijevaju se okolnosti u kojima je nastala slika s jasno prepoznatljivim prometnim znakom, a navodeći pojam originalna slika označava se učitana slika u aplikaciju, preoblikovana u veličinu 100x100. Također, primjećuje se da je razlučivost svake slike znatno poboljšana te je su znakovi još više uočljiviji.

Međutim, u bazi postoji više primjeraka slika iz kojih se teže raspoznaje prometni znak. Takvi primjeri prikazani su slikom 4.2. te, osim djelovanja samog medijan filtera, prikazuju kako Gaussov filtar u kombinaciji s istim utječe na razlučivost takvih slika.

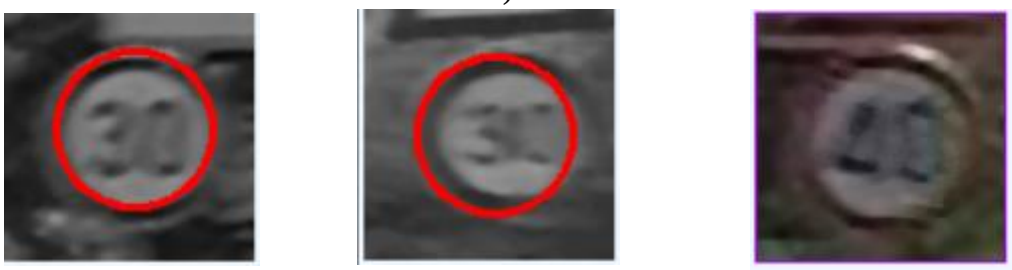


a) b) c)

Slika 4.2. Prikaz djelovanja medijan filtra (b) i Gaussovog filtra (c) na originalnim slikama niske razlučivosti (a)

Na slici 4.2. prikazani su primjeri znakova ograničenja brzine 30 i 40 na kojima je došlo do pogreške prilikom fotografiranja, odnosno čini se da su slike uhvaćene u pokretu što je dovelo do jakih smetnji te se znak vrlo teško razaznaje. Dio šuma uklonjen je medijan filtrom te su rubovi znaka postali dosta jasniji, a Gaussov filtar dodatno zaglađuje cjelokupnu sliku te krajnji rezultat prikazuje kvalitetniju sliku iz koje se može razaznati o kojem je broju riječ. Ovaj proces iznimno je važan za djelovanje Houghove transformacije jer je opće poznato da navedena funkcija ne djeluje baš dobro na slike s jakim smetnjama. Ponajprije, to je tako jer Cannyjev detektor ne može dobro definirati rubove na slikama. To će se najbolje uvidjeti ako se sagleda djelovanje tih funkcija nad istim setom slika kao i na slici 4.2.

a)



b)

Slika 4.3. Postupci konturiranja slika lošije kvalitete: a) Cannyjev detektor, b) HoughCircles funkcija

Razmatrajući identične slika kao i na slici 4.2., može se uvidjeti da Cannyjev detektor u prve dvije slike, između ostalog, razaznaje barem jednu kružnu konturu te je ona, na temelju toga, označena djelovanjem funkcije *HoughCircles*. Ipak, slika koja prikazuje broj 40 nije uspješno detektirana niti u jednoj od ponuđenih veličina. Riječ o fotografiji snimljenoj po danu, na kojoj se, u odnosu na druge dvije promatrane slike, znak ne ističe u okolini, a uz prisutnost dodatnih smetnji nastalih prilikom fotografiranja dio znaka stopio se s prirodnim okruženjem u okviru. U skladu s tim, nije bilo moguće odvojiti cjelovitu kružnicu, čak ni nakon provedbe odabranih filtera, što je i vidljivo na prikazu Cannyjevog detektora u posljednjem dijelu pod a) slike 4.3. U slučaju da funkcija za detekciju kružnice ne radi, ista se pokušava barem djelomično detektirati općom funkcijom *FindCountours,* koja pronalazi sve konture na slici. No, čak ni ta funkcija nije uspješno provela detekciju te je ljubičastom bojom označila čitavu sliku kao područje interesa. Ovakav slučaj nazvat će se graničnim slučajem detekcije kontura. Osim navedenog, postoji nekoliko zanimljivih slučajeva u kojima se detekcija ne izvršava.

a)

b)

Slika 4.4. Prikaz iznimnih slučajeva izostanka detekcije: a) Znak 60 (originalna slika, Cannyjev detektor, filtrirana slika, funkcija *HoughCircles*), b) Znak 50 (originalna slika, Cannyjev detektor, funkcija *FindCountours*)

Slike koje su očito nastale prilikom statičkog položaja kamere, a koje imaju jako malu rezoluciju također spadaju u iznimke kada detekcija po obliku ne djeluje, a to je vidljivo na b) dijelu slike 4.4. Ono što je zaista zanimljivo promatrajući navedenu sliku jest kako slika dosta visoke razlučivosti, u kojoj je Cannyjev detektor označio jednu cjelovitu kružnicu, također spada u ove iznimke. Štoviše, funkcija je detektirala kružnicu, ali ju nije smatrala relevantnom za označavanje. Naime, funkcija *HoughCircles* proslijedila je sliku na daljnju obradu te je od nje napravila samo masku, odnosno slika je zacrnjena, što je vidljivo u posljednjoj slici u a) dijelu slike 4.4. Zbog toga nad slikom su provedena određena ispitivanja i dobiveni različiti rezultati. (Sl. 4.5.)

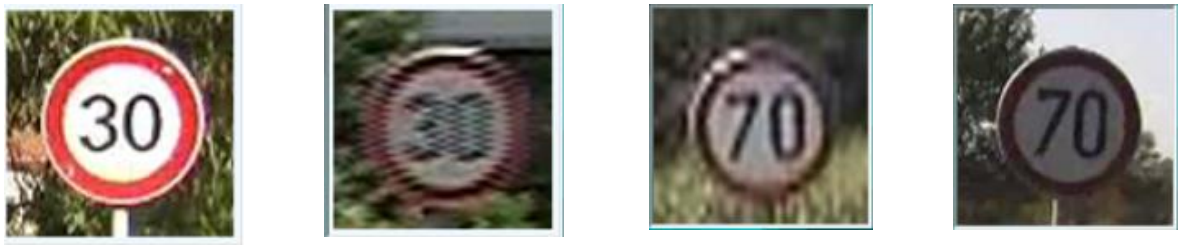
a) b)

Slika 4.5. Prikaz znaka 60 iznimnog ponašanja pod utjecajem različitih parametara i funkcija: a) Rezultat *FindCountorus* funkcije, b) Rezultat *HoughCircles* funkcije

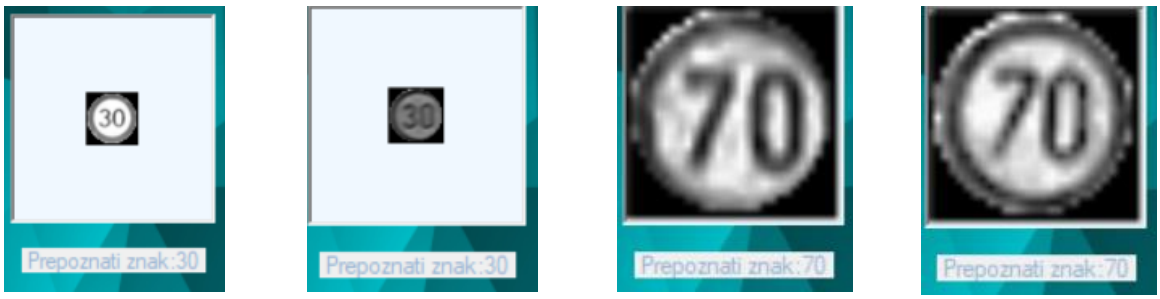
Promjenom parametra rezolucije akumulatora na vrijednost 1, funkcija nije prepoznala kružnicu te ju je proslijedila funkciji za generalno pronalaženje kontura, koja je uspjela detektirati kružnicu nakon što joj se proslijedio rezultat Canny detektora, kao što se može vidjeti u a) dijelu slike 4.5. Daljnjom izmjenom parametara unutar funkcije *HoughCircles*, točnije postavljanjem oba polumjera kružnice na nulu, kako bi se detektirali svi krugovi unutar slike, uspješno je detektirana

kružnica uporabom željene funkcije, što je vidljivo na b) dijelu slike 4.5. Naposljetku, dade se zaključiti da, iako su parametri općenito postizali optimalne rezultate za većinu slika unutar baze, postoje neke iznimke koje zahtijevaju promjenu parametara u svrhu postizanja željenih rezultata.

Govoreći o implementaciji metode potpornih vektora, treba napomenuti kako prednost svakako predstavlja spremanje rezultata istrenirane baze u tekstualnu datoteku, što uvelike umanjuje potrebno vrijeme osposobljavanja sustava za provedbu testova, odnosno za prepoznavanje znaka u željenoj slici.



a)



b)

Slika 4.6. Prikaz prepoznavanja znakova: a) Originalne slike, b) Filtrirane slike izdvojenih značajki testirane u SVM-u

Prepoznavanje znakova uspješno se odvija za slike različite rezolucije i slike snimane pod različitim uvjetima, što je pokazano primjerom različitih slika znakova ograničenja brzine na slici 4.6. Unutar aplikacije, ispod svake slike, moguće je vidjeti tekstualni ispis znaka koji je prepoznat. Istrenirano je ukupno 280 slika prometnih znakova ograničenja brzine. Uspješnost prepoznavanja testirano je nad 192 slike, a iznosi ukupno 77,604%. Prikaz točnosti prepoznavanja svakog znaka posebno, kao i broj slika nad kojima je izvršen trening i testiranje, prikazano je tablicom 4.1.

Znak ograničenja brzine	Broj istreniranih slika	Broj testiranih slika	Točnost testiranja (%)
30	93	44	84,09
40	121	56	91,07
50	56	24	79,17
60	93	40	60,00
70	60	28	64,28

Tablica 4.1. Točnost prepoznavanja znakova ograničenja brzine

Prema tablici 4.1., prometni znak ograničenja brzine 40 ima najveću točnost prepoznavanja, a najmanju točnost imaju znakovi ograničenja brzine 60 i 70. Postoji mogućnost dobivanja boljih rezultata kada bi se baza slika prometnih znakova ograničenja brzine podijelila tako da 70% ukupnog broja slika pripada bazi za trening, a 30% bazi za testiranje jer na taj način metoda potpornih vektora koristi više slika za učenje, čime se povećava vjerojatnost prepoznavanja prometnog znaka. Originalna baza podijeljena je tako da broj slika u bazi za trening sadrži približno 55% ukupnih slika. Također, razlog lošijeg prepoznavanja znakova 60 i 70 leži u tome što broj 60 ima zaobljeni dio kao i 30 i 50 , ali sadrži zatvorenu liniju kao broj 40 stoga ih SVM algoritam ponekad krivo prepoznaje. Slično se može dogoditi s brojem 70 koji se u procesu prepoznavanja češće miješa s brojevima 30 zbog otvorenosti broja (nema zatvorenih linija), te 40 i 50, zbog sličnosti u ravnim linijama koje SVM može krivo raspoznati pogotovo u slučaju lošije kvalitete slike ili uslijed pogrešne detekcije područja interesa. (Sl. 4.7.)

a) b) c) d)

Slika 4.7. Pogrešno prepoznati prometni znakovi ograničenja brzine a) znak 70 prepoznat kao 40, b) znak 70 prepoznat kao 30, c) znak 60 prepoznat kao 30 , d)¸ znak 60 prepoznat kao 40

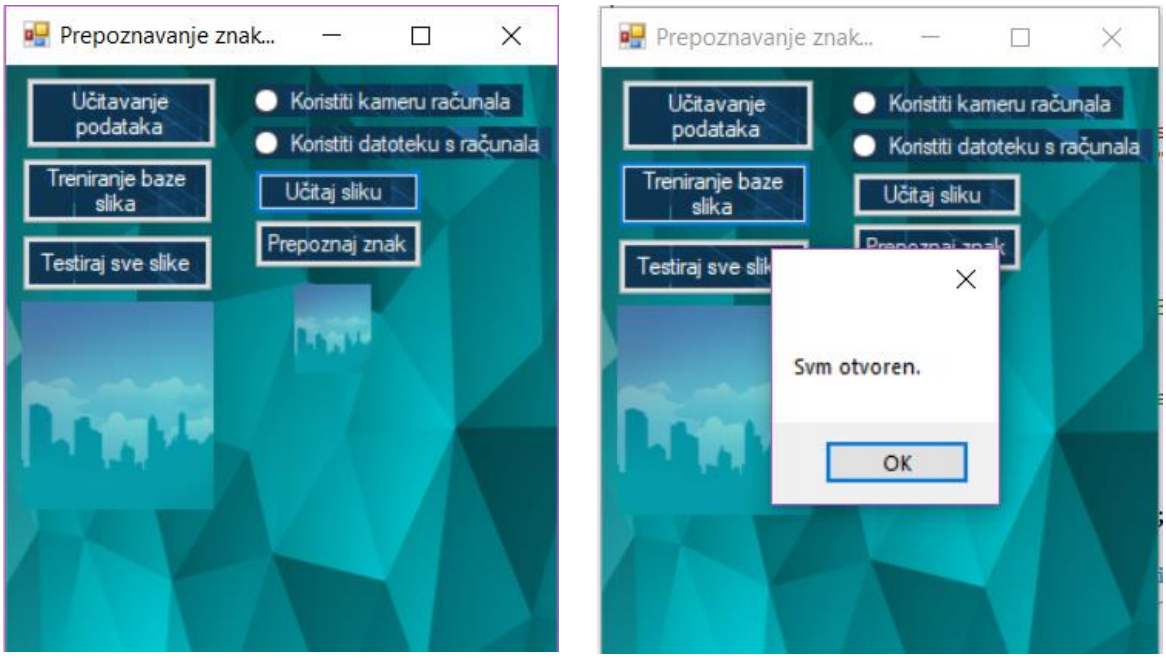
Obzirom na količinu slika dostupnih za obradu i korištenje funkcija fiksnih vrijednosti parametara za sve slike, rezultati testiranja su zadovoljavajući te za većinu slika uspješno prepoznaju prometni znak ograničenja brzine.

4.1. Programsko rješenje za prepoznavanje znakova ograničenja brzine

Aplikacija za prepoznavanje znakova ograničenja brzine pokriva funkcionalnosti koje se očekuju kako bi se stekao uvid u sve segmente korištenih funkcija obrade slike te nudi nekoliko načina prepoznavanja znakova: uporabom kamere računala, odabirom željene slike iz baze podataka za testiranje ili pokretanjem automatskog testiranja svih prometnih znakova ograničenja brzine uz ispis statistike uspješnosti istog.

4.1.1. Trening slika

Odabirom opcije „Učitavanje podataka“ dohvaćaju se slike iz mape slika za trening SVM-a te se iste obrađuju prethodno pojašnjenim filterima i stvara se jedinstveni vektor za svaku sliku. Uporaba navedenog neophodna je za rad čitave aplikacije. Korisnik zatim odabire opciju „Treniranje baze slika“, kojom SVM klasificira potporne vektore slika dobivene iz matrice vektora te ih sprema u zasebnu datoteku. Ukoliko već postoji navedena datoteka preskače se korak učitavanja podataka te se odmah odabire opcija „Treniranje baze slika“, pri čemu se pojavljuje poruka u novom prozoru *„SVM otvoren“*.

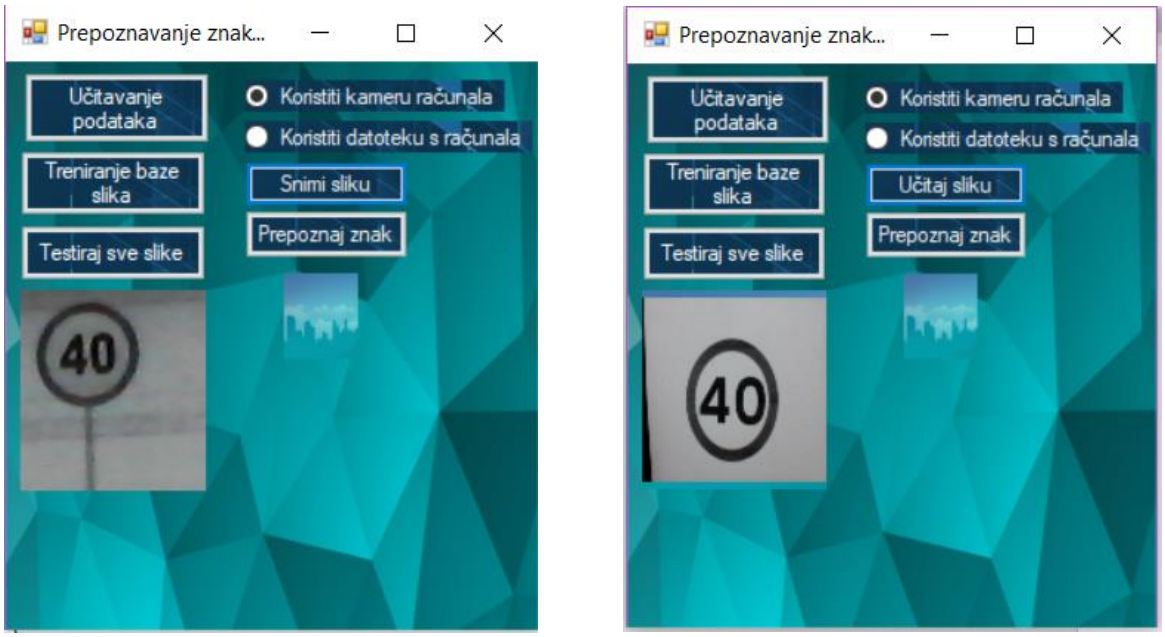


a) b)

Slika 4.8. a) Korisničko sučelje aplikacije za prepoznavanje znakova ograničenja brzine, b) Prikaz poruke prilikom otvaranja postojeće SVM datoteke

4.1.2. Prepoznavanje prometnih znakova

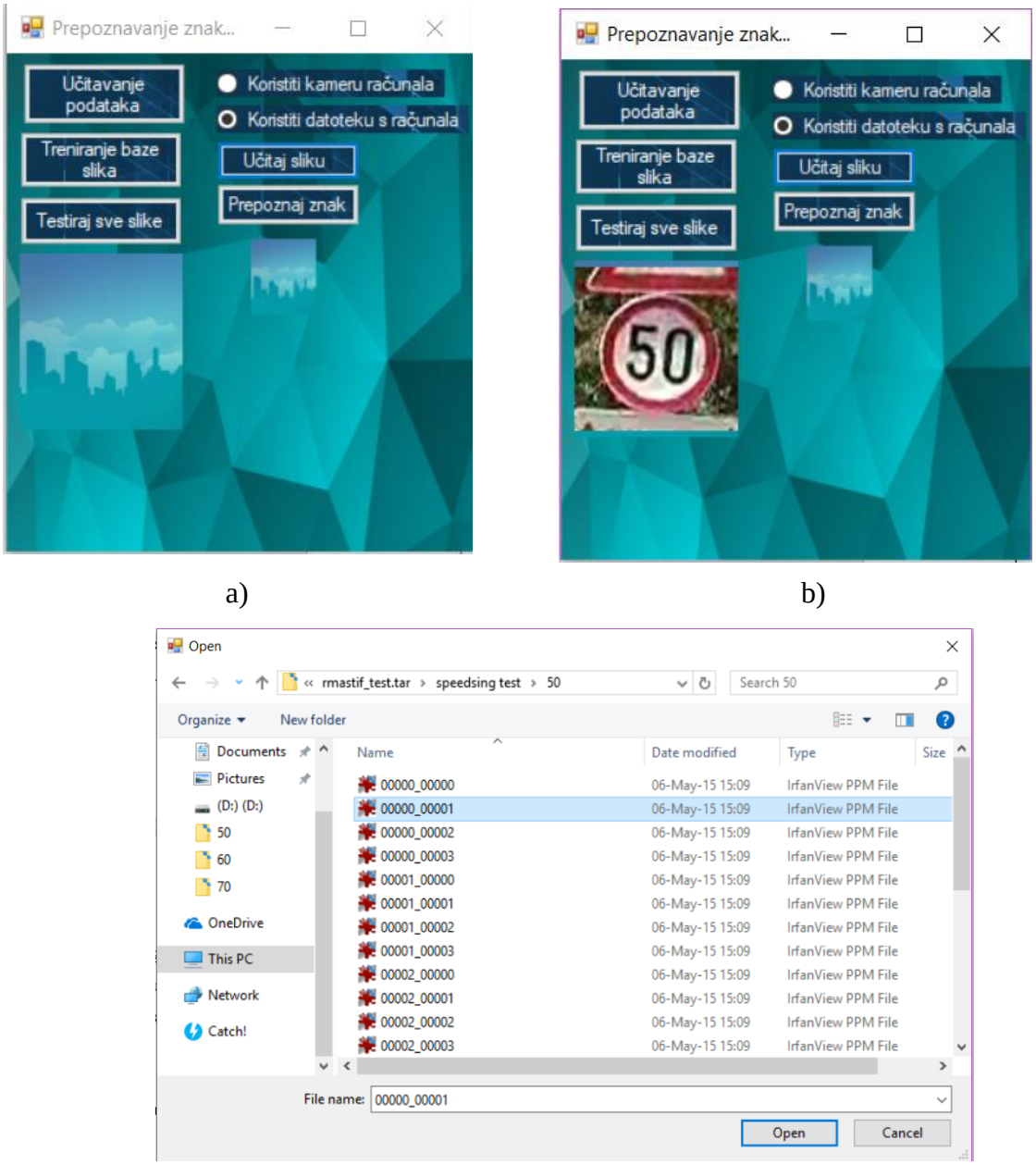
Postoje tri načina prepoznavanja prometnih znakova sa slike: odabir opcije „Testiraj sve slike“ te označavanje opcije „Koristiti kameru računala“ ili „Koristiti datoteku s računala“ uporabom opcijskih gumba, uslijed čega je potrebno odabrati opciju „Učitaj sliku“. Ukoliko se korisnik odluči za opciju „Koristiti kameru računala“, upalit će se kamera računala na kojem se izvodi aplikacija te će se pojaviti opcija „Snimi sliku“ koju je potrebno odabrati nakon što se odabere željeni položaj slike, koji se može odrediti prema želji, pomoću prozora za prikaz slika unutar aplikacije. (Sl. 4.9)



a) b)

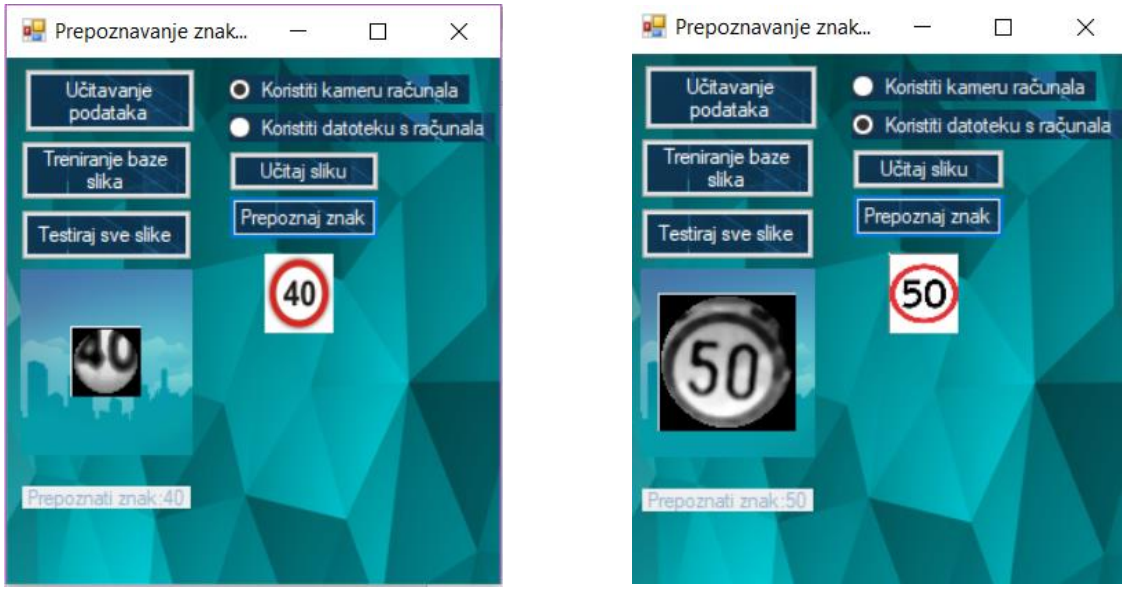
Slika 4.9. Korisničko sučelje odabirom opcije „Koristiti kameru računala“: a) Prikaz učitavanja slike, b) Prikaz snimljene slike

U slučaju odabira opcije „Koristiti datoteku s računala“ (Sl. 4.10. a), otvara se prozor s podacima računala unutar kojeg je potrebno označiti željenu datoteku za testiranje i otvoriti je odabirom opcije „Open“. (Sl. 4.10. c) Uspješnim odabirom slike, ista se pojavljuje u prozoru aplikacije predviđenim za prikaz slika. (Sl. 4.10. b)



Slika 4.10. a) Korisničko sučelje aplikacije s prikazom odabira opcije „Koristiti datoteku s računala“, b) Korisničko sučelje – prikaz odabrane slike, c) Prozor s podacima za odabir i označavanje željene datoteke

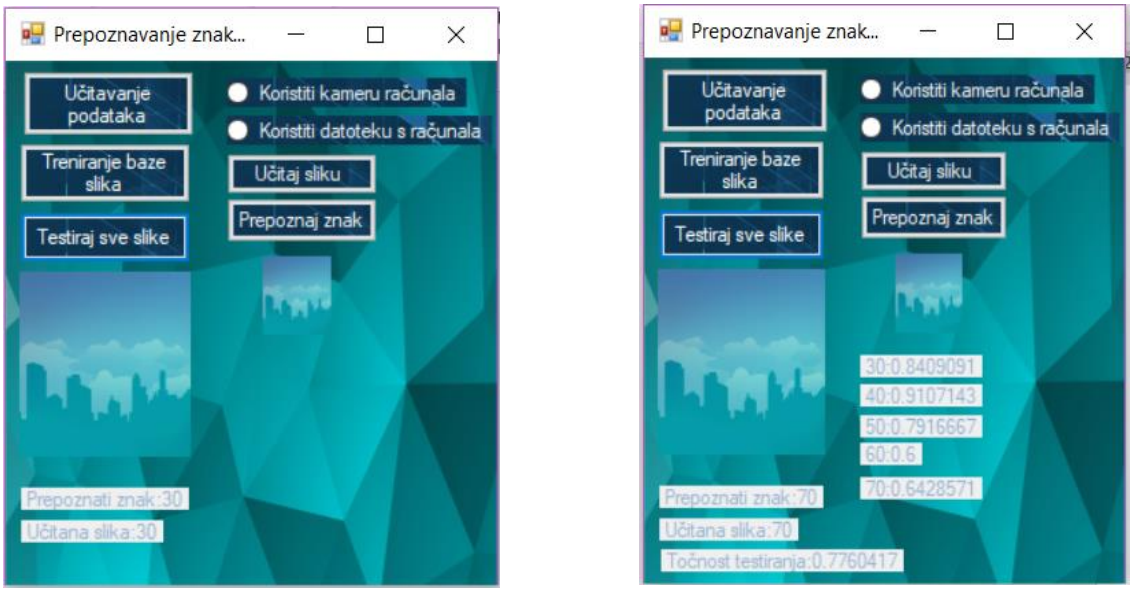
Nakon odabira jedne od opcija prikazanih na slikama 4.9. i 4.10., potrebno je izabrati opciju „Prepoznaj znak“, nakon čega se, u prozoru predviđenom za prikaz slika, mogu vidjeti rezultati provođenja svih filtera i funkcija te konačan rezultat, koji je ispisan ispod učitane slike te, ujedno, prikazan u prozoru na desnoj strani korisničkog sučelja aplikacije, koji služi za prikaz prepoznatih slika. (Sl.4.11.)



a) b)

Slika 4.11. Prikaz rezultata testiranja korištenjem opcije: a) „Koristiti kameru računala“, b) „Koristiti datoteku s računala“

Konačno, odabirom opcije „Testiraj sve slike“, iste se najprije učitavaju, obrađuju ranije spomenutim filterima i funkcijama te se, pomoću metode potpornih vektora, redom predviđa prometni znak ograničenja brzine, pri čemu se ispisuju prometni znakovi učitane i prepoznate slike. (Sl.4.12. a) Nakon što se predvidi znak na svim slikama, pojavljuje se ukupna statistika uspješnosti algoritma. (Sl. 4.12. b)



a) b)

Slika 4.12. Uporaba opcije „Testiraj sve slike“: a) Prepoznavanje prometnih znakova, b) Statistika uspješnosti algoritma

5. ZAKLJUČAK

Svakim danom raste prodor tehnologije u sve pore svakodnevnice ljudi. Prisutnost obrade slike do samog prepoznavanja objekata sigurno je jedna od najzanimljivijih, ali i najkorisnijih pojava današnjice, posebice u području razvijanja autonomnih sustava. Postavljene u prirodno okruženje, slike često mogu biti registrirane na mnogo načina, uključujući različita osvjetljenja i različite kutove. Imajući navedeno na umu, bilo je potrebno osmisliti algoritam koji bi mogao prepoznavati prometne znakove barem u većini mogućih situacija te pokazati određene situacije u kojima pojedini dijelovi sustava zakazuju u izvedbi. Ovaj rad bavi se razvitkom aplikacije za prepoznavanje znakova ograničenja brzine u C# programskom okruženju, koristeći pritom Emgu CV, inačicu najpoznatije biblioteke za obradu slike, Open CV-ja, i bazu raznovrsnih slika rMASTIF, koje oponašaju moguće realne situacije. Izrađen je sustav koji upotrebljava standardni medijan filtar, Gaussovu funkciju, Houghovu transformaciju i SVM s ciljem uspostave algoritma koji će krajnjem korisniku pružiti mogućnost predviđanja zamijećenih prometnih znakova ograničenja brzine na kolniku.

Preliminarna obrada podataka podrazumijevala je uporabu standardnog medijan i Gaussovog filtra. Rezultati eksperimenta pokazali su da čak i na slikama s većom količinom šuma uporaba ova dva filtra čini sliku dovoljno razlučivom za uspješnu daljnju obradu. Nadalje, Houghova transformacija korištena je za detekciju prometnog znaka. Pokazano je da u većini slučajeva Cannyjev detektor uspješno prepozna barem jednu kružnicu, što je uvjetovalo uspješno djelovanje Houghove transformacije. Ipak, pronađene su određene slike s prevelikim šumom i nedovoljne razlučivosti na kojima nije bilo moguće detektirati znak. Osim toga, postoje određeni primjeri nad kojima se ne odvija detekcija zbog fiksno postavljenih parametara. No, promjenom istih omogućen je pronalazak prometnog znaka u slici. Pravilna izvedba dosad navedenih faza iznimno je važna za algoritam prepoznavanja jer je omogućeno izdvajanje bitnih značajki slike i prosljeđivanje istih algoritmu za učenje, odnosno prepoznavanje. Metoda potpornih vektora korištena je za učenje na izdvojenim značajkama slike postupkom klasifikacije. Pokazano je da, na temelju naučenih slika, ova metoda uspješno predviđa o kojem je prometnom znaku ograničenja brzine riječ. Prednost je korištenje opcije spremanja naučenog sadržaja, stoga se smanjuje vrijeme osposobljavanje sustava za cjelokupnu obradu slike.

Metoda primjene obrađenog načina detekcije u kombinaciji sa strojnim učenjem na području prepoznavanja znakova kao kamen temeljac ključna je za sveukupni razvoj ove grane znanosti.

Međutim, razvitak novih tehnologija, poput neuronskih mreža, omogućava bržu i precizniju preliminarnu obradu slike kao i prepoznavanje značajki na istoj. Ipak, na posljednjem sloju neuronske mreže često se nalazi implementirana metoda poput SVM-a. Spomenuta vrsta tehnologije danas je u uporabi prilikom razvitka gotovo svih autonomnih sustava.

LITERATURA

[1] E. J. Leavline, D. A . A. G. Sing, Salt and Pepper Noise Detection and Removal in Gray Scale Images: An Experimental Analysis, International Journal of Signal Processing, Image Processing and Pattern Recognition, No. 5, Vol. 6, str. 343-344, 2013. godina

[2] G. Kaur, R. Kumar, K. Kainth, A Review Paper on Different Noise Types and Digital Image Processing, International Journal of Advanced Research in Computer Science and Software Engineering, No. 6, Vol. 6, str. 562-564, lipanj, 2016. godine

[3] S.H. Teoh, H. Ibrahim, Median Filtering Frameworks for Reducing Impulse Noise from Grayscale Digital Images: A Literature Survey, International Journal of Future Computer and Communication, No. 4, Vol. 1, str. 323-327, prosinac, 2012. godina

[4] R. Fisher, S. Perkins, A. Walker and E. Wolfart, Hypermedia image processing reference: Gaussian smoothing [online], Edinburgh, 2003. godine, dostupno na: https://homepages.inf.ed.ac.uk/rbf/HIPR2/whatis.htm [15.06.2018.]

[5] L. Kabbai, A. Sghaier, A. Douik, M, Machout, FPGA implementation of filtered image using 2D Gaussian filter, International Journal of Advanced Computer Science and Applications, No. 7, Vol. 7, str. 514-520, 2016. godina

[6] A. Adam, C. Ioannidis, Automatic Road-Sign Detection And Classification Based On Support Vector Machines And Hog Descriptors, ISPRS Annals Of The Photogrammetry, Remote Sensing And Spatial Information Sciences, No. 5, Vol. 2, lipanj, 2014. godine

[7] R. Biswas, J. Sil, An Improved Canny Edge Detection Algorithm Based on Type-2 Fuzzy Sets, Precedia Technology, Vol. 4, str. 820-824., 2012. godina

[8] T. Šmuc, Strojno učenje: Metoda potpornih vektora, Zagreb, ožujak, 2013. godine

[9] P. Petej, Klasifikacija dokumenata primjenom algoritama za strojno učenje na paralelnoj GPU arhitekturi, FESB Split, lipanj, 2014. godine

[10] S. Lončarić, M. Subašić, Neuronske mreže: Stroj s potpornim vektorima [online], FER, Zagreb, dostupno na: http://www.ieee.hr/_download/repository/08a-SVM.pdf [15.06.2018]

[11] G. Bradski, A. Kaehler, Learning OpenCV, O'Reilly Media, SAD, 2008. godina

[12] T. Joachims, Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms, Springers US, New York, SAD, 2002. godine

[13] R. Fan, P. Chen, C. Lin, Working Set Selection Using Second Order Information for Training Support Vector Machines, Journal of Machine Learning Research 6, str. 1889-1918, Taiwan, 2005. godina

SAŽETAK

Metoda potpornih vektora jedna je od najučinkovitijih za rješavanje problema prepoznavanja prometnih znakova. U ovom radu predstavljena je aplikacija za prepoznavanje znakova ograničenja brzine iz baze slike znakova. Aplikacija koristi standardni medijan i Gaussov filtar za uklanjanje šuma iz slika, Houghovu transformaciju za detekciju kružnog objekta i metodu potpornih vektora za prepoznavanje znaka. Prilikom implementacije metode potpornih vektora, bilo je potrebno klasificirati i istrenirati set slika iz baze rMASTIF. Analizom algoritma, uspostavljeni su prikladni parametri svih upotrijebljenih funkcija za postizanje optimalnih rezultata u većini slučajeva. Isto tako, prilikom detekcije pronađeni su primjeri gdje je nužno promijeniti parametre za postizanje točnih rezultata. Također, utvrđeni su primjeri iznimno loše kvalitete slike nad kojima je nemoguće obaviti detekciju. Usprkos tome, trening nad bazom uspješno je proveden te je testiranjem implementirane metode utvrđeno kako u većini slučajeva algoritam pravilno predviđa zadani prometni znak.

Ključne riječi: metoda potpornih vektora, standardni medijan filtar, Gaussov filtar, Houghova transformacija, klasifikacija

THE SPEED LIMIT ROAD SIGNS RECOGNITION USING EXISTING ROAD SIGN IMAGE DATABASES

ABSTRACT

A support vector machine is one of the most efficient methods used for traffic sign recognition. In this paper application for speed limit traffic sign recognition from a traffic sign image database is presented. The application uses standard median filter and Gaussian filter for removing noise in corrupted images, Hough transformation for detection of a circular object within an image and a support vector machine for sign recognition. During the implementation of the support vector machine, it is necessary to classify and train a set of images taken from the rMASTIF database. While analyzing the algorithm adequate parameters are established for every function used, in order to achieve as optimal results as possible. Also, a few examples are found where changing the parameters is necessary to obtain accurate results. In addition, there are a few images in which it is impossible to detect a circle due to images' bad quality. Despite identifying significant amount of problems, the support vector machine is successfully trained and tested. In most cases, the algorithm accurately predicting a speed sign in the tested images is established.

Key words: support vector machine, standard median filter, Gaussian filter, Hough transformation, classification

ŽIVOTOPIS

Ivona Matoš rođena je 27. kolovoza 1996. u Zagrebu. Osnovnu školu pohađala je u Dugom Selu, nakon koje upisuje Sedmu Gimnaziju u Zagrebu, koju završava s odličnim uspjehom. Tijekom školovanja sudjelovala je na natjecanjima iz fizike, matematike i hrvatskog jezika. U sklopu srednjoškolskog obrazovanja sudjeluje u programu za dobivanje Njemačke jezične diplome razine C1, koju, nakon uspješno položenih ispita, stječe godine 2015. Također, iste godine seli u Vukovar i upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Posljednje godine preddiplomskog studija posvećuje se proučavanju obrade slike i strojnog učenja te sudjeluje u studentskim projektima Pametni stol i Pametno raskrižje.

PRILOZI

Izvorni kod aplikacije priložen je na CD-u.