

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 859

**DETEKCIJA PROMETNIH ZNAKOVA NA
TEMELJU MODELA OBLIKA**

Marijan Fofonjka

Zagreb, lipanj 2009.

Sadržaj:

1. Uvod	1
2. Pregled ostvarenog sustava i programskih alata	3
2.1. Biblioteka za računalni vid OpenCV	3
2.2. Pregled ostvarenog sustava	3
2.3. Digitalni fotoaparat	5
2.4. Baza ulaznih slika	6
2.5. Baza kandidata	6
3. Obrada slike	7
3.1. Promjena dimenzija ulazne slike	8
3.2. Izjednačavanje histograma	9
3.3. Pretvorba sustava boja RGB→HSV	12
3.4. Izdvajanje crvene boje	14
3.5. Uklanjanje šuma	17
4. Prepoznavanje oblika	22
4.1. Pronalaženje vanjskih kontura i proračun affinih momenata	22
4.2. Izdvajanje i pohrana konačnih kandidata u bazu kandidata	25
5. Uspješnost rada i analiza rezultata neuspješne detekcije ostvarenog sustava	30
5.1. Ocjena uspješnosti rada ostvarenog sustava	30
5.2. Analiza rezultata neuspješne detekcije ostvarenog sustava	35
6. Upute za instalaciju i korištenje programski ostvarenog sustava	38
6.1. Instalacijski postupak za Windows operacijski sustav	38
6.2. Upute za korištenje programski ostvarenog sustava	41
7. Zaključak	43
8. Literatura	44

1. Uvod

Prometni znakovi su znakovi postavljeni na prometnicama, koji upozoravaju sudionike u prometu kako se sigurno ponašati, odnosno koja pravila vrijede na određenoj dionici javne prometnice. Prometni znakovi dizajnirani su tako da ih sudionici u prometu mogu na vrijeme i lako uočiti temeljem njihovog oblika i različite boje u odnosu na okolinu. Prema zakonu o sigurnosti prometa na cestama [1], „prometni znakovi jesu: znakovi opasnosti, znakovi izričitih naredbi, znakovi obavijesti i znakovi obavijesti za vođenje prometa s dopunskom pločom koja je sastavnim dijelom prometnog znaka i koja pobliže određuje značenje prometnog znaka ili bez nje, promjenjivi prometni znakovi, prometna svjetla i svjetlosne oznake te oznake na kolniku i drugim površinama“. Sigurno jednu od najvažnijih skupina prometnih znakova za sigurno i pravilno kretanje pojedinaca u prometu čine znakovi opasnosti i izričitih naredbi. Većina znakova tih skupina sadrži crveni okvir unutar kojega se nalazi ključna informacija za pojedinca uključenog u promet. Ostvareni sustav vrši detekciju isključivo takvih znakova uz uvjet da su oni okruglog, trokutastog ili osmerokutnog oblika. Prikaz skupa prometnih znakova koje ostvareni sustav detektira dan je u tablici 1. .

Rad je strukturiran kako slijedi. U poglavlju 2. dan je pregled ostvarenog sustava uz opis njegovih ulaznih i izlaznih dijelova, te pregled korištenih programskih alata za programsko ostvarenje sustava. U poglavlju 3. i 4. dan je opsežan opis rada i korištenih metoda dvaju glavnih procesa sustava: proces obrade slike i proces prepoznavanja oblika. U poglavlju 5. dana je ocjena uspješnosti i analiza rezultata neuspješne detekcije ostvarenog sustava na temelju baze ulaznih slika. U poglavlju 6. dane su upute za instalaciju i korištenje sustava. U poglavlju 7. je zaključak. U poglavlju 8. dan je pregled korištene literature.

Tablica 1. Skup prometnih znakova koje ostvareni sustav detektira

Podskup znakova opasnosti									
									
									
									
									
									
Podskup znakova izričitih naredbi									
									
									
									
									

2. Pregled ostvarenog sustava i programskih alata

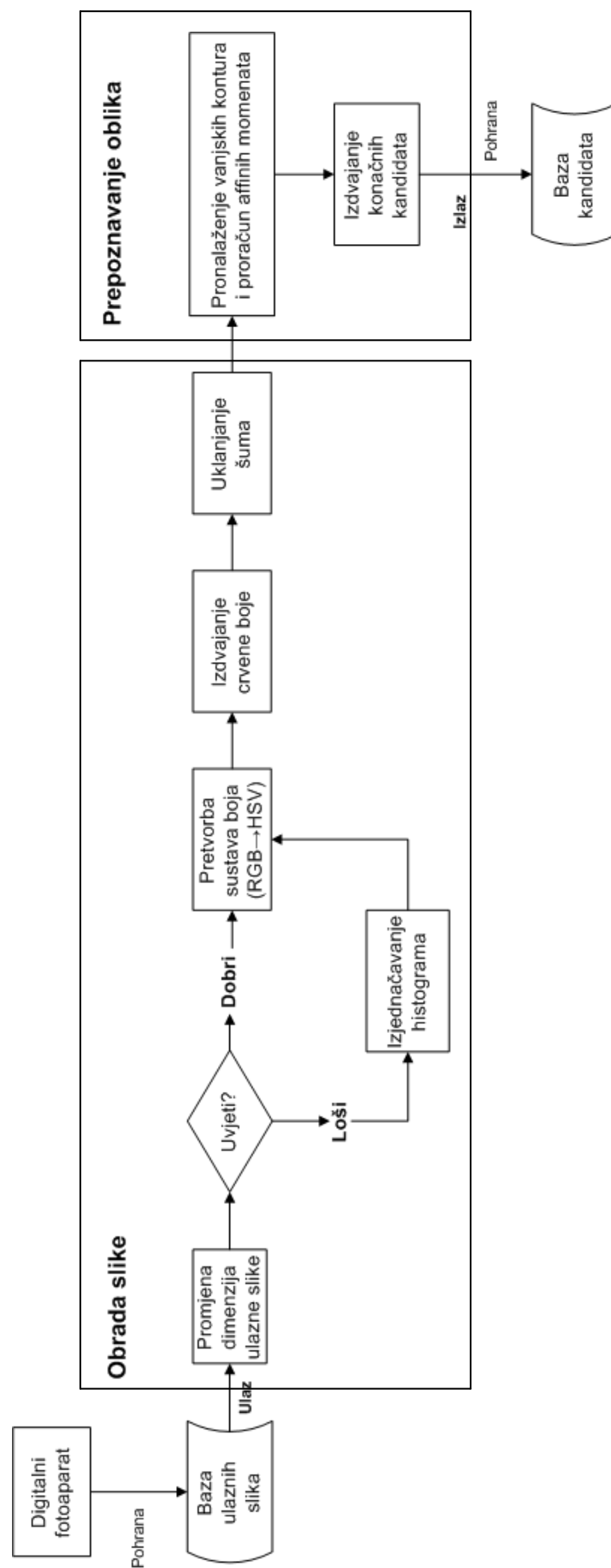
Ovo poglavlje daje pregled ostvarenog sustava uz opis njegovih ulaznih i izlaznih dijelova te pregled korištenih programskih alata za programsko ostvarenje sustava. Za programsko ostvarenje sustava korišten je razvojni programski alat tvrtke Microsoft, *Microsoft Visual Studio 2008* [2] u kombinaciji sa bibliotekom za računalni vid *OpenCV* tvrtke Intel. Podpoglavljje 2.1. daje kratak uvid u biblioteku *OpenCV*. U podpoglavljju 2.2. dana je shema ostvarenog sustava. U podpoglavljju 2.3. dane su karakteristike digitalnog fotoaparata korištenog za stvaranje baze ulaznih slika. U podpoglavljju 2.4 dan je pregled korištene baze ulaznih slika. U podpoglavljju 2.5. dan je pregled baze kandidata.

2.1. Biblioteka za računalni vid OpenCV

OpenCV (Open Computer Vision Library) je biblioteka za računalni vid otvorenog koda, dostupna na [3]. Prvu verziju biblioteke razvila je tvrtka Intel 1999. godine. Biblioteka je pisana u programskim jezicima C i C++ i moguće ju je koristiti pod Windows, Linux i Mac OS X operacijskim sustavima. Glavni cilj programera pri stvaranju biblioteke bio je omogućiti korisnicima jednostavan i brz razvoj aplikacija na području računalnog vida. Jedno od bitnih svojstava aplikacija izgrađenih na ovoj biblioteci je mogućnost korištenja višejezgrenih procesora. Biblioteka sadrži preko 500 funkcija koje obuhvaćaju skoro sva područja današnje primjene računalnog vida, primjer takvih je tvorničko ispitivanje proizvoda, obrade slika u medicini, stvaranje sigurnosih sustava, ostvarivanje stereo vida u robotici itd. . Pošto su područja računalnog vida i strojnog učenja usko povezana, OpenCV sadrži i biblioteku za strojno učenje naziva MLL (Machine learning library). MLL obuhvaća algoritme za izgradnju klasifikatora temeljenih na statističkim metodama ili postupcima grupiranja (eng. clustering). Mnogi poznati sustavi današnjice na području računalnog vida izgrađeni su na OpenCV biblioteci, primjer jednog takvog je robot „Stanley“ [4] sa sveučilišta Stanford, pobjednik pustinjske utrke robota \$2M DARPA Grand Challenge.

2.2. Pregled ostvarenog sustava

Na slici 1. dana je shema ostvarenog sustava.



Slika 1. Shema sustava za detekciju prometnih znakova na temelju modela oblika

2.3. Digitalni fotoaparati

Pri prikupljanju podataka potrebnih za izradu baze ulaznih slika, korišten je digitalni fotoaparati SONY DSC-T9 prikazan na slici 2. . Bitne specifikacije fotoaparata vezane uz rad dane su u tablici 2. .



Slika 2. Digitalni fotoaparati SONY DSC-T9

Tablica 2. Specifikacije digitalnog fotoaparata SONY DSC-T9

Svojstva	Specifikacija
Objektiv	Carl Zeiss Vario-Tessar
Format slike	JPEG
Maksimalna rezolucija	2816x2112 (6 mega piksela)
Zumiranje	3x optičko + 6x digitalno
F	3,5 – 4,3
Duljina fokusa (f = mm)	6,3 – 19,0
Postavka ISO osjetljivosti	80/100/200/400/640/Auto
Podešavanje osvjetljenja	Auto, danje svijetlo, oblačno, fluorescentno, obična žarulja, bljeskalica
Izlazni priključak	USB 2.0

2.4. Baza ulaznih slika

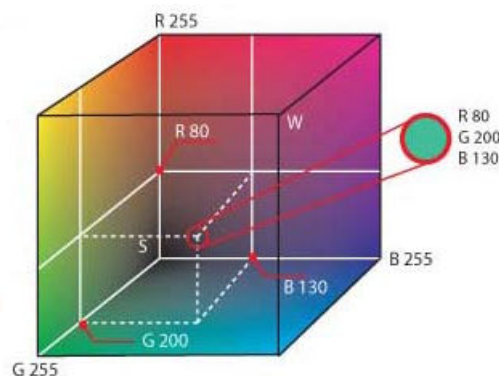
Bazu ulaznih slika čine slike znakova dobivene normalnim slikanjem u mirnom položaju i slikanjem iz vozila u pokretu na području grada Samobora i grada Zagreba. Slike su u boji i slikane su pod različitim vremenskim uvjetima; sunčano, kišno i sumrak. Baza ulaznih slika sadrži ukupno 140 slika u JPG formatu, rezolucija 2816x2112 i 2048x1536 piksela. Slike su podijeljene u dvije grupe; dobri uvjeti i loši uvjeti, ovisno o uvjetima u kojima su slikane i načinu rada ostvarenog sustava.

2.5. Baza kandidata

Bazu kandidata čine slike dimenzija 36x36 u 24bitnom JPG, RGB formatu. Slike predstavljaju kandidate pronađene na ulaznoj slici sustava i naziva su *kandidatX.jpg*, gdje *X* predstavlja redni broj pronađenog kandidata. Slike unutar baze kandidata mogu se dalje koristiti kao ulaz u klasifikator koji vrši točno prepoznavanje pojedinog prometnog znaka.

3. Obrada slike

Kvalitetno predprocesirana slika na ulazu sustava uvelike ovisi o konačnim rezultatima samog sustava, stoga ćemo se u ovome poglavlju posvetiti postupcima predprocesiranja tj. obrade ulazne slike. Proces obrade ulazne slike podijeljen je na pet glavnih podprocesa. U podpoglavlju 3.1. opisan je podproces promjene dimenzija ulazne slike. U podpoglavlju 3.2. opisan je podproces izjednačavanja histograma. U poglavlju 3.3. opisan je podproces pretvorbe sustava boja ulazne slike iz RGB sustava u HSV sustav boja. U podpoglavlju 3.4. opisan je podproces izdvajanja crvene boje. U podpoglavlju 3.5. opisan je podproces uklanjanja šuma. Ostvareni sustav fleksibilan je što se tiče formata ulaznih slika, tako su podržani danas najčešće korišteni formati slika: BMP, DIB, JPEG, JPG, JPE, PNG, PBM, PGM, PPM, SR, RAS, TIFF i TIF. Sustav podržava rad sa ulaznim slikama različitih dimenzija, više o tome dano je u podpoglavlju 3.1. Kako sustav radi detekciju znakova na temelju boje, jedino ograničenje na skupu baze ulaznih slika jest da ona sadrži slike u boji. Ulazne slike u sustav učitavaju se u 24bitnom RGB formatu, pri čemu svaki od kanala sadrži osam bita, tj. vrijednosti od 0 do 255. 24bitni RGB format možemo promatrati kao kocku prikazanu na slici 3. . Kocka je smještena u cjelobrojni trodimenzionalni sustav koordinata R (eng. red, hrv. crvena), G (eng. green, hrv. zelena) i B (eng. blue, hrv. plava). Bridovi kocke su duljine 255. Boju pojedinog piksela slike određuju njegove pripadne cjelobrojne koordinate u trodimenzionalnom RGB sustavu. Na slici 1. možemo vidjeti 24bitni trodimenzionalni RGB sustav sa primjerom dodjele boje jednom pikselu. Rubne vrijednosti boja 24bitnog RGB sustava sa pripadnim koordinatama dane su u tablici 3. .



Slika 3. 24bitni RGB sustav sa primjerom dodjele boje pikselu

Tablica 3. rubne vrijednosti boja 24bitnig RGB sustava

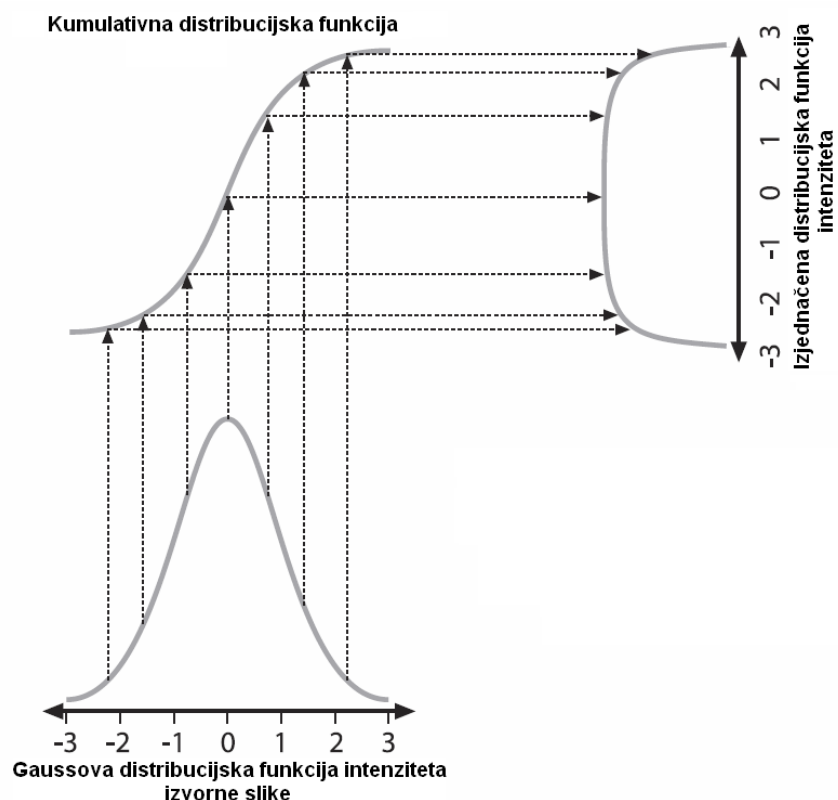
Koordinate (R, G, B)	Boja
(0, 0, 0)	crna
(255, 255, 255)	bijela
(255, 0, 0)	crvena
(0, 255, 0)	zelena
(0, 0, 255)	plava
(255, 255, 0)	žuta
(0, 255, 255)	cijan
(255, 0, 255)	ružičasta

3.1. Promjena dimenzija ulazne slike

Nakon što je učitana ulazna slika proizvoljnih dimenzija u 24bitnom RGB formatu, vrši se njeno skaliranje na fiksne dimenzije 640x480. Skaliranje je od izričitog značaja za brzinu rada sustava jer manja slika znači manji broj ulaznih podataka tj. informacija za pojedini algoritam, a time i brže izvođenje. Dimenzije 640x480 uzete su zato što predstavljaju optimalnu količinu informacija za ispravan rad i brzinu sustava. Razlog zašto se manje slike od fiksnih dimenzija skaliraju na fiksne dimenzije, leži u ispravnosti rada algoritma za otklanjanje šuma, o kojemu će više riječi biti u podpoglavlju 3.5. Skaliranje slike provodi se postupkom ponovnog uzorkovanja (eng. resampling). Ulazna RGB slika dijeli se na tri crno bijele slike, posebno za svaki kanal. Svaki tako dobiveni kanal provodi se kroz algoritam resamplinga. Ukoliko je ulazna slika većih dimenzija od zadanih 640x480, provodi postupak downsamplinga. Postupkom downsamplinga novonastali pikseli slike manjih dimenzija dobivaju se tako da se uzima prosjek vrijednosti piksela izvorne slike koju novonastali piksel prekriva. Ovim postupkom vrši se izbacivanje piksela ulazne slike. Ukoliko je ulazna slika manjih dimenzija od zadanih 640x480, provodi se postupak upsamplinga. Postupkom upsamplinga uz postojeće piksele izvorne slike dodaju se novi pikseli na temelju susjedne interpolacije piksela izvorne slike.

3.2. Izjednačavanje histograma

Postupkom promjene dimenzija ulazne slike dobivamo 24bitnu RGB sliku dimenzija 640x480. Obzirom na vremenske uvjete u kojima se vrši detekcija znakova, razlikujemo dobre i loše. Pod pojmom dobrih uvjeta smatramo uvjete u kojima je dobro izražen kontrast slike tj. ravnomjerna je distribucija intenziteta piksela na interval $[0, 255]$, pa je crvena boja znaka dovoljno dobro izražena na izvornoj slici. Tu spadaju primjerice slike snimane u sunčanim vremenskim uvjetima gdje imamo dobro osvjetljenje prometnih znakova. Pod pojmom loših uvjeta smatramo uvjete u kojima je loš kontrast slike tj. distribucija intenziteta ulazne slike nalazi se oko sredine intervala vrijednosti $[0, 255]$, pa je izvorna slika presvijetla ili pretamna. Tu spadaju primjerice slike snimane u sumrak ili u kišnim i maglovitim uvjetima gdje crvena boja znaka nije dovoljno dobro izražena. Ukoliko se radi o lošim vremenskim uvjetima, da bi se ulazna slika poboljšala provodi se postupak izjednačavanja histograma. Cijela ideja iza navedenog postupka leži u tome da se loša početna distribucija intenziteta izvorne slike raspoređuje u neku drugu širu i u idealnom slučaju uniformnu distribuciju intenziteta. Drugim riječima želimo zgusnute intenzitete izvorne slike ravnomjerno raspršiti na intervalu $[0, 255]$. To je moguće ukoliko je funkcija raspodjele kumulativna distribucijska funkcija. Radi pojašnjenja postupka uzmimo jednostavan primjer prikazan na slici 4. u kojemu se funkcija intenziteta izvorne slike ravna po Gaussovoj normalnoj razdiobi. Prikazanu kumulativnu distribucijsku funkciju u primjeru možemo iskoristiti ne samo na Gaussovoj, već i na bilo kojoj drugoj funkciji razdiobe intenziteta. Djelovanjem kumulativne distribucijske funkcije nad Gaussovom distribucijskom funkcijom intenziteta dobivamo izjednačenu distribucijsku funkciju intenziteta, tj. obavili smo izjednačavanje histograma. Izjednačavanje histograma unutar ostvarenog sustava provodi se tako da postupak primjenjujemo na svaki RGB kanal posebno, pri čemu je svaki kanal predstavljen crno bijelom slikom.



Slika 4. jednostavan primjer izjednačavanja histograma

Značaj primjene postupka izjednačavanja histograma u radu sustava možemo vidjeti na slikama 5. i 6. . Na slici 5. izvorna slika slikana je u lošim vremenskim uvjetima (kiša i sumrak) pa prometni znak nije detektiran. Nakon primjene postupka izjednačavanja histograma sustav uspješno detektira prometni znak. Na slici 6., izvorna slika slikana je u lošim vremenskim uvjetima (sumrak) pa je samo jedan prometni znak uspješno detektiran. Nakon primjene postupka izjednačavanja histograma sustav uspješno detektira sva tri prometna znaka.

ULAZ



REZULTATI DETEKCIJE NA IZVORNOJ SLICI



ULAZ



REZULTATI DETEKCIJE NA IZVORNOJ SLICI



Slika 5. rezultat detekcije prije i poslije primjene postupka izjednačavanja histograma nad slikom slikanom u sumraku i kišnim uvjetima

ULAZ



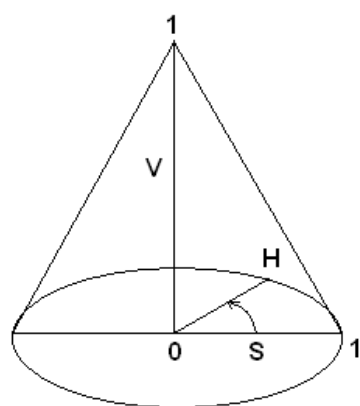
REZULTATI DETEKCIJE NA IZVORNOJ SLICI



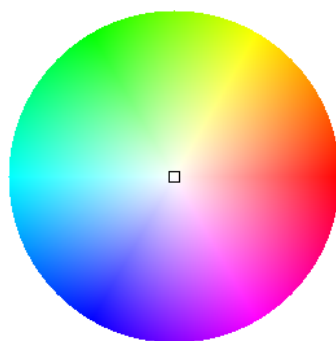
Slika 6. rezultat detekcije prije i poslije primjene postupka izjednačavanja histograma nad slikom slikanom u sumrak

3.3. Pretvorba sustava boja RGB→HSV

Jedan od velikih nedostataka RGB sustava je nemogućnost izravnog izdvajanja informacije o boji od informacije o osvjetljenju. Za razliku od RGB sustava u HSV sustavu to je puno jednostavnije. Na temelju istraživanja u radovima [5] i [6] autori su pokazali da je vrijednost komponente H (eng. Hue) u HSV sustavu manje ovisna o promjenama osvjetljenja i djelovanju sijena, nego što su to R, G i B komponente u RGB sustavu. Zbog stalne promijene uvjeta i osvjetljenja tijekom vožnje, razumno je obaviti pretvorbu ulazne slike iz RGB sustava u HSV sustav boja. HSV sustav boja čine tri glavne komponente H (eng. hue), S (eng. saturation) i V (eng. value). Možemo ga prikazati kao stožac prikazan na slici 7. kojemu baza određuje komponente H i S a visina određuje komponentu V. Komponenta H predstavlja boju kakvom je mi zamišljamo: crvena, zelena, plava, ljubičasta itd. . Na bazi stošca predstavljena je kutom od 0° do 360° od horizontalne osi crvene boje. Komponenta S u intervalu $[0, 1]$ predstavlja zasićenost, tj. dominaciju H komponente u konačnoj boji. Pri rubovima kružnice ($S = 1$) imamo potpunu dominaciju H komponente tj. čistu boju, dok pomakom prema središtu njena dominacija se smanjuje. U središtu kružnice ($S = 0$) kažemo da niti jedna boja ne dominira i ono predstavlja sivu boju (eng. gray scale). Komponenta V predstavljena u intervalu $[0, 1]$ predstavlja osvjetljenje, tj. intenzitet odabrane boje. Za vrijednost $V = 1$ boja je najsvjetlija, dok je za vrijednost $V = 0$ najtamnija.



Stožac HSV sustava boja



baza stošca



Visina stošca

Slika 7. stožac HSV sustava boja

Pretpostavimo da su sve komponente RGB sustava normalizirane unutar realnog intervala $[0, 1]$. H komponenta predstavlja vrijednost kuta od 0° do 360° od horizontalne osi crvene boje. Komponente S i V neka se nalaze unutar realnog intervala $[0, 1]$. Općenito komponente za pojedini piksel slike HSV sustava iz RGB sustava boja možemo dobiti na sljedeći način:

Komponenta V:

$$V = \max(R, G, B) \quad (1)$$

Komponenta S:

$$\text{Ako je } \max(R, G, B) \neq 0 \quad S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} \quad (2)$$

$$\text{Ako je } \max(R, G, B) = 0 \quad S = 0 \quad (3)$$

Komponenta H:

Ako je $S = 0$ H je nedefinirana

$$\text{Ako je } R = \max(R, G, B) \quad H = \frac{G - B}{\max(R, G, B) - \min(R, G, B)} \quad (4)$$

$$\text{Ako je } G = \max(R, G, B) \quad H = 2.0 + \frac{B - R}{\max(R, G, B) - \min(R, G, B)} \quad (5)$$

$$\text{Ako je } B = \max(R, G, B) \quad H = 4.0 + \frac{R - G}{\max(R, G, B) - \min(R, G, B)} \quad (6)$$

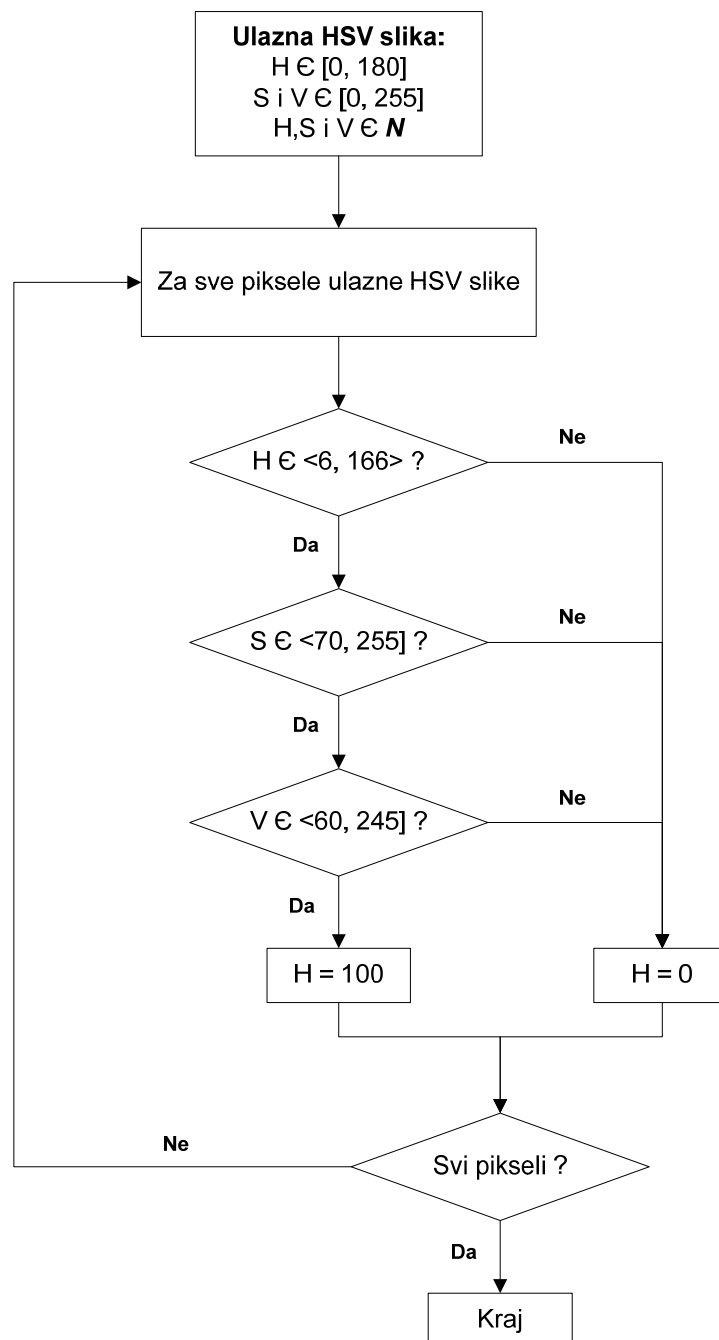
$$H = 60 \times H \quad (7)$$

$$\text{Ako je } H < 0 \quad H = H + 360^\circ \quad (8)$$

Kako u ostvarenom sustavu baratamo sa diskretnim vrijednostima komponenti HSV sustava boja. Nakon pretvorbe 24bitne ulazne RGB slike u HSV sustav, H komponenta poprima diskretne vrijednosti iz intervala $[0, 180]$, komponente S i V poprimaju diskretne vrijednosti iz intervala $[0, 255]$.

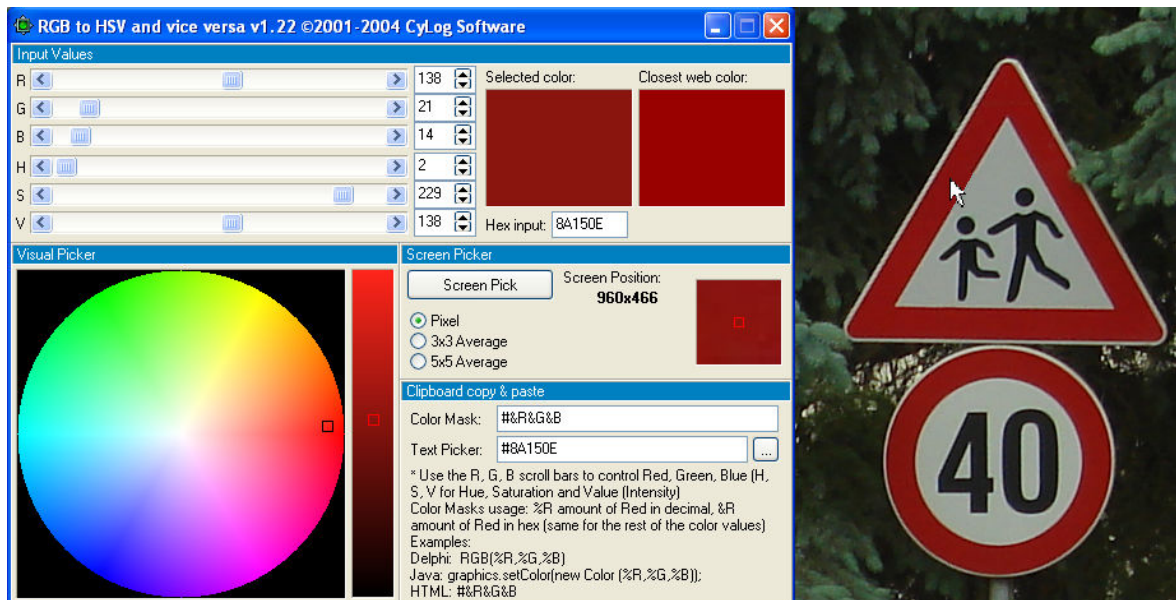
3.4. Izdvajanje crvene boje

Kako bi se omogućila lakša detekcija prometnih znakova u daljnjim koracima rada sustava, vrši se izdvajanje crvene boje iz ulazne HSV slike. Za svaki piksel ulazne slike primjenjuje se algoritam prikazan dijagramom toka na slici 8. . Rezultat primjene algoritma daje binarizirani H kanal koji sadrži korisnu informaciju i koristi se kao ulaz za daljnje postupke rada sustava. Pikseli crvene boje, koji sadrže korisnu informaciju iz ulazne slike imaju H vrijednost 100, dok ostali pikseli imaju H vrijednost 0.



Slika 8. dijagram toka algoritma za izdvajanje crvene boje

Pri određivanju graničnih vrijednosti za izdvajanje informacije pojedinog kanala korišten je program *RGB to HSV and vice versa* verzije 1.22 tvrtke CyLog Software [7]. Program sadrži opciju Screen Pick pomoću koje je direktno moguće dobiti HSV vrijednost pojedinog piksela ulazne slike. Granične vrijednosti H, S i V komponenti algoritma određene su uporabom opcije Screen Pick na bazi ulaznih slika. Prikaz programa i primjera uporabe opcije Screen Pick dan je na slici 9. . Program daje vrijednosti komponente H u diskretnom intervalu od [0, 255], kako takav raspon vrijednosti nije prikladan za algoritam, vrši se množenje sa koeficijentom $k = \frac{180}{255}$ i zaokruživanje na prvu veću cjelobrojnu vrijednost. Na slici 10. moguće je vidjeti neke izlaze algoritma za izdvajanje crvene boje primijenjenog na bazi ulaznih slika.

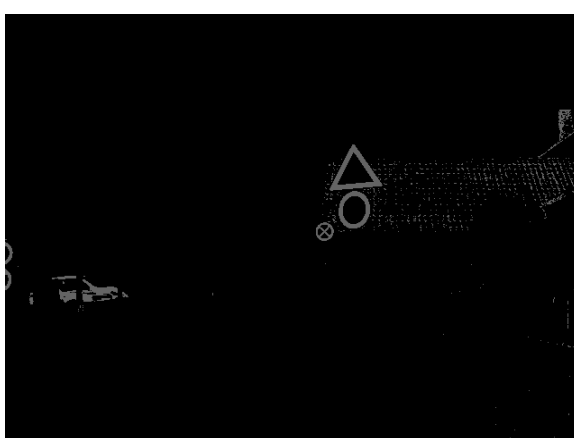
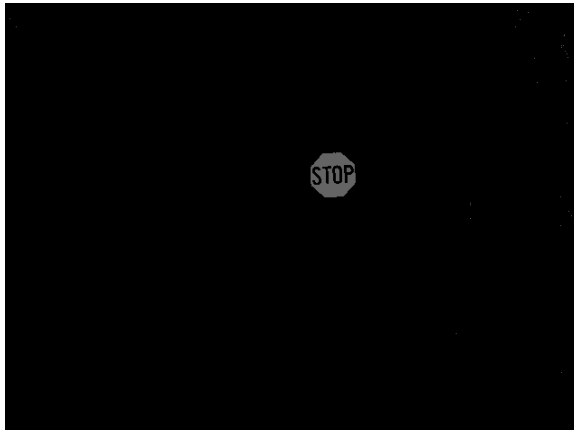


Slika 9. direktno dobivanje H, S i V komponenti traženog piksela uporabom programa *RGB to HSV and vice versa* verzije 1.22 tvrtke CyLog Software

ULAZ



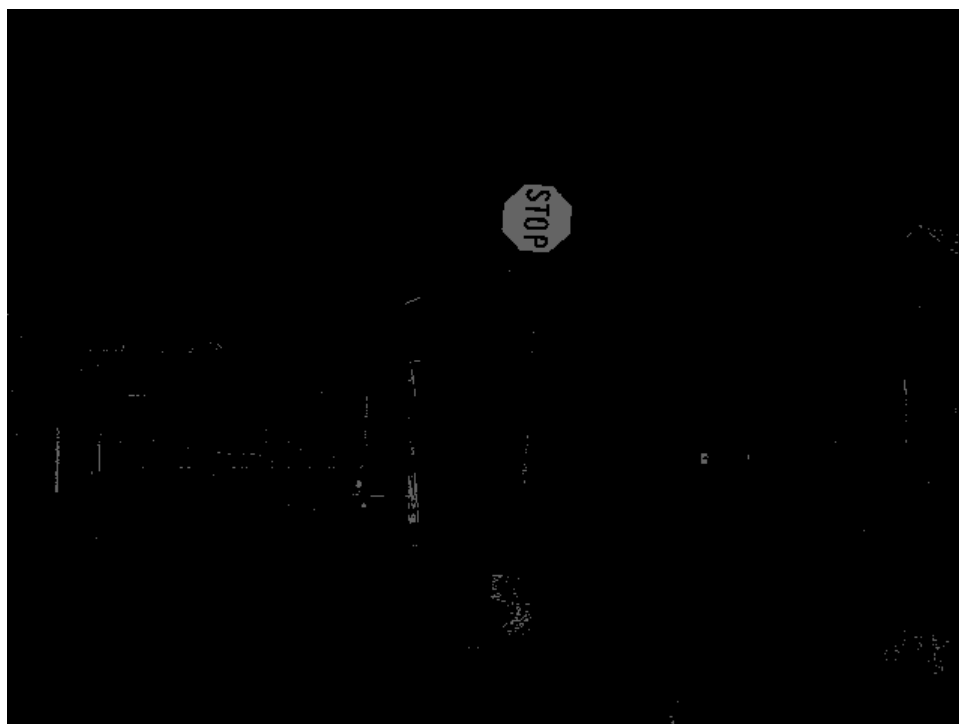
IZLAZ IZ ALGORITMA ZA IZDVAJANJE BOJE



Slika 10. primjeri izlaza algoritma za izdvajanje crvene boje

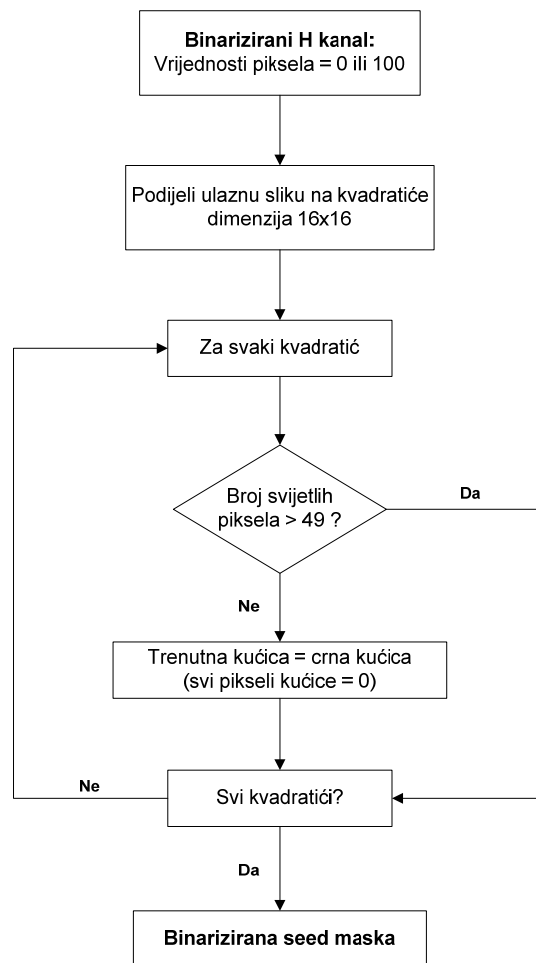
3.5. Uklanjanje šuma

Nakon primjene algoritma za izdvajanje crvene boje na ulaznoj slici u daljem postupku koristimo samo dobiveni binarizirani H kanal. Pošto algoritam izdvaja svu crvenu boju ulazne slike koja zadovoljava pragove za pojedini kanal, moguća je pojava šuma u H kanalu. Šum H kanala opisujemo kao objekte male površine koji nastaju zbog okoline u kojoj se znak nalazi. Na slici 11. možemo vidjeti primjer šuma u binariziranom H kanalu.

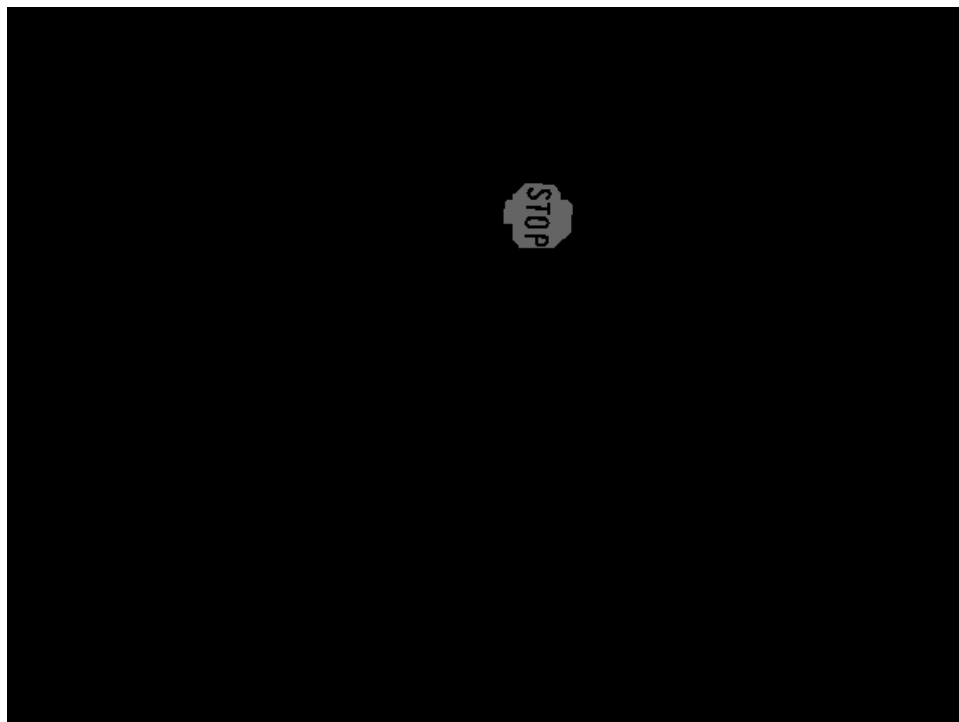


Slika 10. primjer šuma u binariziranom H kanalu

Kako bi ubrzali daljnje procese sustava, na binariziranom H kanalu primjenjujemo algoritam za otklanjanje šuma prikazan dijagramom toka na slici 11. . Algoritam dijeli ulaznu sliku na kvadratiće dimenzija 16x16 i računa ukupan broj svijetlih piksela unutar pojedinog kvadratića. Ukoliko je taj broj manji ili jednak od zadane donje granice, trenutni kvadratić zamjenjuje se crnim kvadratićem, inače se ne dira. Kao rezultat primjene algoritma dobivamo binariziranu seed masku prikazanu na slici 12. u kojoj je u idealnom slučaju sav šum otklonjen.



Slika 11. dijagram toka algoritma za otklanjanje šuma



Slika 12. binarizirana seed maska dobivena primjenom algoritma za otklanjanje šuma na sliku 10.

Djelovanjem algoritma otklanjanja šuma na binariziranom H kanalu može doći do smanjivanja korisne informacije sadržane u prometnom znaku. Kućice koje zahvaćaju rubove znaka mogu biti uklonjene iz izvorne slike zbog nedovoljnog broja svijetlih piksela. Primjer takve pojave vidljiv je na slici 12. . Iz navedenog vidimo da je potrebno obaviti obnavljanje korisne informacije na dobivenoj seed masci. Da bi se razumio daljnji postupak potrebno je dati uvid u osnovnu funkcionalnost flood fill algoritma.

Algoritam flood fill

Općenito postoji više načina rada flood fill algoritma, u nastavku kroz niz koraka objašnjena je inačica koja se koristi u ovome radu:

1. Kao ulaz algoritma zadaje se ulazna slika (nad kojom se vrše sve promjene i ispitivanja) , seed piksel, donja i gornja granica intenziteta i nova vrijednost piksela. Prototip funkcije:

FloodFill(ulazna slika, seed piksel, donja granica, gornja granica, nova vrijednost piksela)

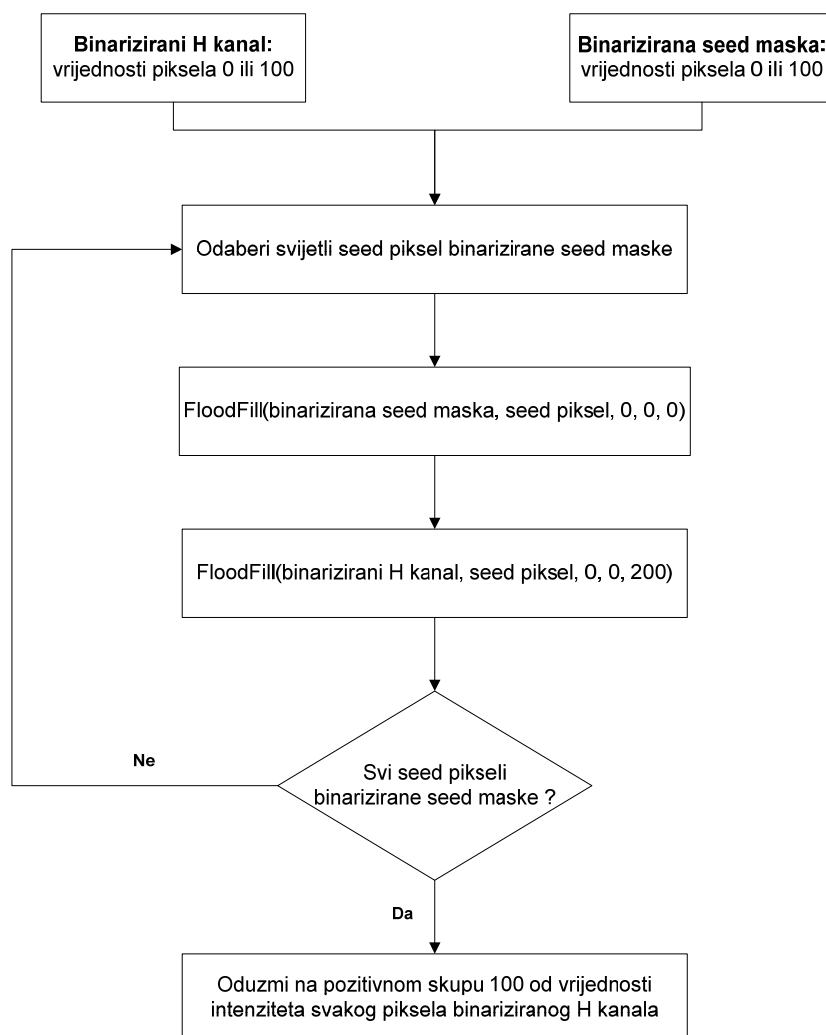
2. Traže se svi susjedni pikseli zadanog piksela, ukoliko pojedini susjedni piksel ima vrijednost intenziteta u zadanim granicama:
$$\text{seed piksel} - \text{donja granica} \leq \text{susjedni piksel} \leq \text{seed piksel} + \text{gornja granica}$$

, njegova vrijednost zamjenjuje se novom zadanom vrijednošću.
3. Ako postoje pikseli kojima je promijenjena vrijednost u prethodnom koraku, oni postaju zadani pikseli i algoritam se nastavlja od prethodnog koraka za svaki od njih.
4. Ako ne postoje pikseli kojima je promijenjena vrijednost u drugom koraku, algoritam se zaustavlja.

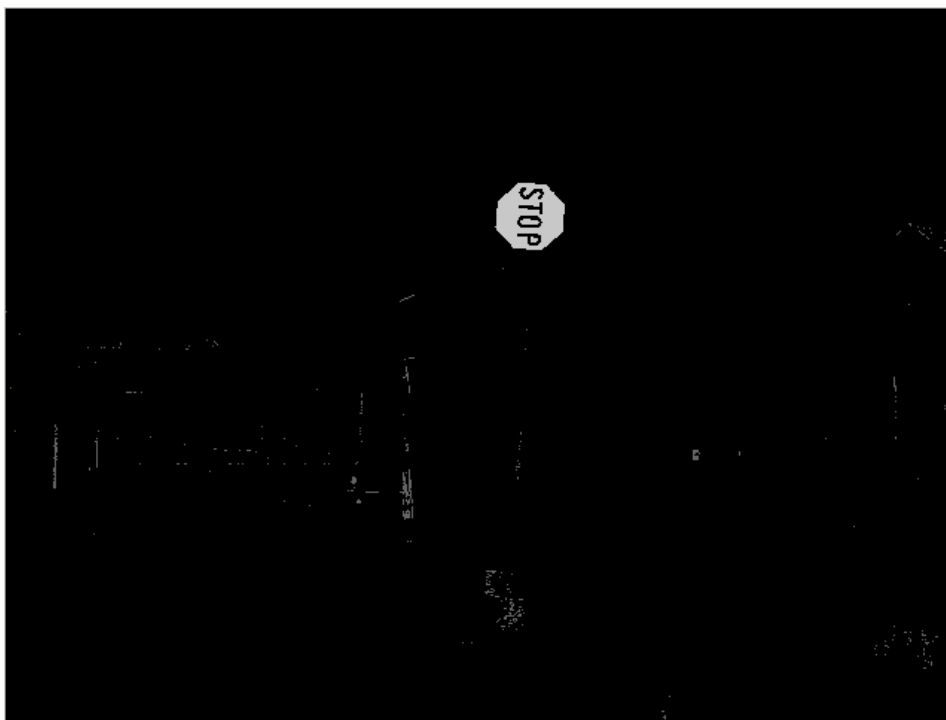
Obnova informacije

Kombiniranjem binariziranog H kanala, binarizirane seed maske i flood fill algoritma vrši se obnova korisne informacije. Algoritam za obnovu korisne informacije prikazan je dijagramom toka na slici 13. . Ulaz u algoritam čine binarizirani H kanal i binarizirana seed maska. Svijetli pikseli unutar binarizirane

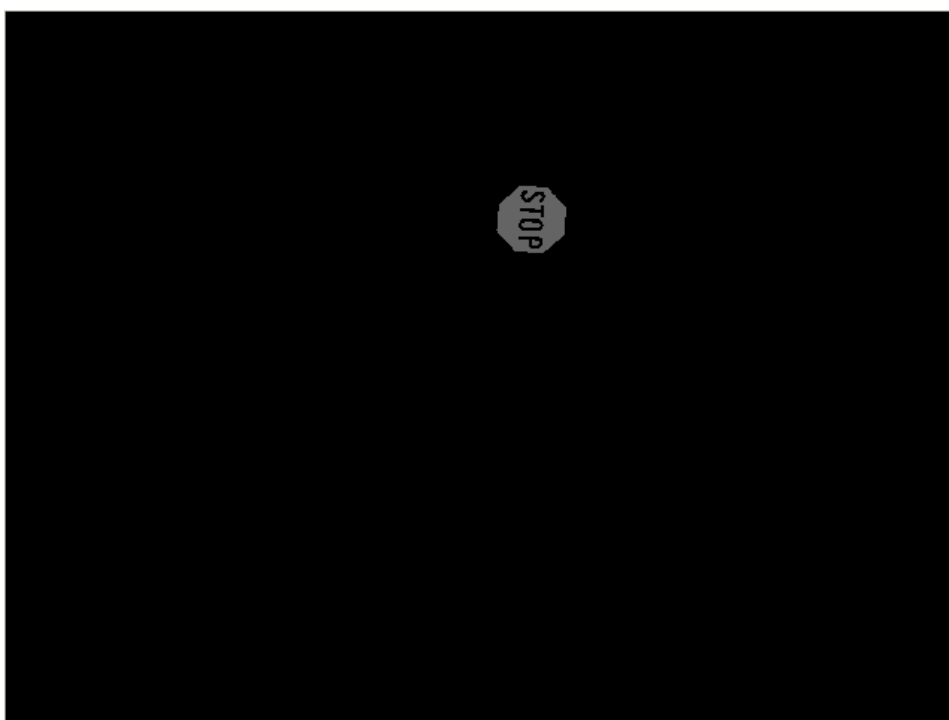
seed maske predstavljaju seed piksele za flood fill algoritam koji se poziva za binarizirani H kanal. Pošto je binarizirana seed maska izvedena iz binariziranog H kanala, pikseli koji su povezani u binariziranoj seed masci, povezani su i u binariziranom H kanalu, pa je moguće nakon odabira seed piksela ukloniti njega i sve njegove susjede u binariziranoj seed masci i time smanjiti broj budućih seed piksela a time i broj poziva flood fill algoritma za binarizirani H kanal. Nakon što binarizirana seed maska ne sadrži više niti jedan svijetli seed piksel, dobiva se konačni oblik binariziranog H kanala u kojemu pikseli koji predstavljaju korisnu informaciju imaju vrijednost intenziteta 200, a ostali pikseli vrijednost intenziteta 0 ili 100 (slika 14.). Oduzimanjem na pozitivnom skupu vrijednosti 100 od intenziteta svakog piksela binariziranog H kanala, dobivamo konačni rezultat uklanjanja šuma a time i procesa obrade slike (slika 15.) . Pod pojmom oduzimanje na pozitivnom skupu, smatra se da se sve negativne vrijednosti dobivene oduzimanjem postavljaju na 0.



Slika 13. dijagram toka algoritma za obnovu korisne informacije



Slika 14. binarizirani H kanal sa slike 12. nakon zadnje primjene flood fill algoritma



Slika 15. rezultat procesa obrade slike dobivenog primjenom algoritma za obnavljanje informacije nad slikama 10. i 12.

4. Prepoznavanje oblika

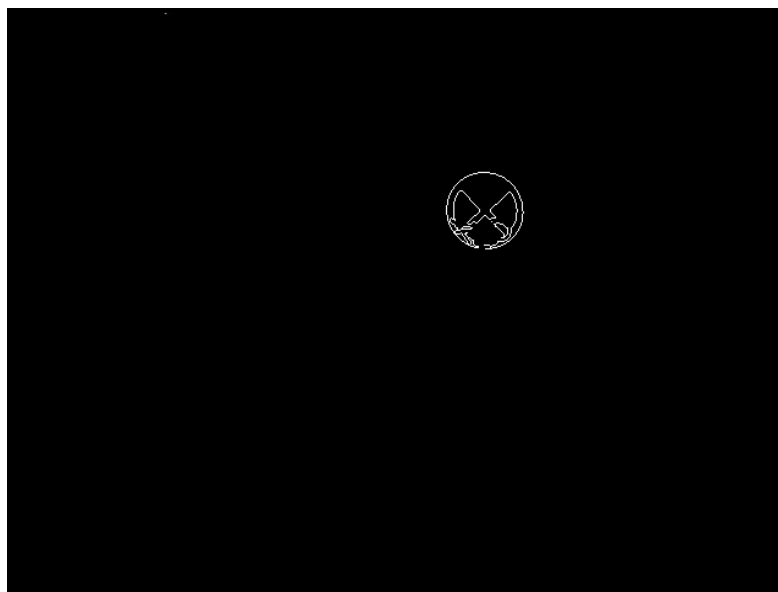
Kao rezultat procesa obrade slike dobiva se binarizirana slika koja sadrži izdvojenu korisnu informaciju uz smanjen ili potpuno uklonjen šum. Osim prometnih znakova, binarizirana slika većinom sadržava i brojne druge objekte koji su zadovoljili algoritme procesa obrade. Kako bi se iz skupa objekata razlučili samo prometni znakovi, nad binariziranom slikom vrši se proces prepoznavanja oblika. Proces prepoznavanja oblika podijeljen je na dva glavna podprocesa. U podpoglavlju 4.1. opisuje se podproces pronalaženja vanjskih kontura i proračuna affinih momenata. U podpoglavlju 4.2. opisuje se podproces izdvajanja i pohrane konačnih kandidata u bazu kandidata.

4.1. Pronalaženje vanjskih kontura i proračun affinih momenata

Binarizirana slika sadrži skupine piksela na temelju kojih ništa nije moguće zaključiti bez promatranja oblika kojega one čine. Kako bi dobili informaciju o obliku, na binariziranoj slici vrši se postupak pronalaženja vanjskih kontura. Vanjsku konturu definiramo kao konturu koja nije sadržana unutar niti jedne druge konture. Primjer rezultata postupka pronalaženja vanjskih kontura na binariziranoj slici (slika 16.) prikazan je na slici 17. .

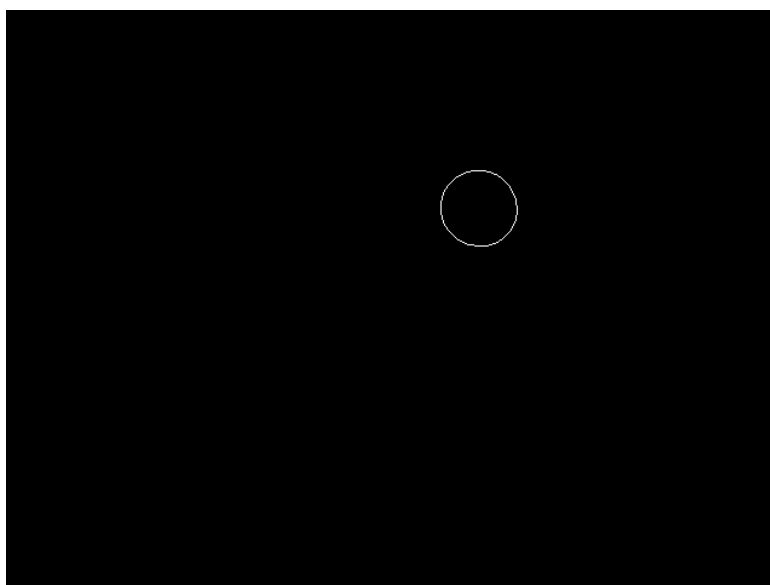


Slika 16. primjer binarizirane slike dobivene na temelju ulazne slike oštećenog prometnog znaka



Slika 17. pronađene vanjske konture na binariziranoj ulaznoj slici sa slike 16. .

Prilikom izdvajanja crvene boje na ulaznoj slici zbog oštećenja, izbljedjele boje ili prisutnosti objekata na slici koji zaklanjaju prometni znak, može se dogoditi da izdvojeni rubovi prometnih znakova nisu kontinuirani. Vanjske konture pronađene na takvoj slici, nisu nam pogodne za daljnji rad algoritma. Primjer takve pojave prikazan je na slici 17.. Zbog toga sve pronađene vanjske konture pretvaramo u konveksne konture [8]. Kontura je konveksna ukoliko ne postoji spojница dviju proizvoljnih točaka konture koja bi prolazila izvan konture. Rezultat pretvorbe vanjske konture sa slike 16. u konveksnu konturu, prikazan je na slici 17..



Slika 17. primjer pretvorbe vanjske konture sa slike 16. u konveksnu konturu

Na binariziranoj slici nakon procesa obrade često puta nije uklonjen sav šum. Traženjem vanjskih kontura na takvoj slici, tražimo i konture šuma. Ukoliko je površina konveksne konture manja od 225 piksela, smatramo je konturom šuma i uklanjamo je iz binarizirane slike. Svaka preostala konveksna kontura u binariziranoj slici predstavlja mogući prometni znak. Kako bi odvojili konveksne konture koje predstavljaju prometne znakove od ostalih konveksnih kontura, provodimo postupak proračuna invarijantnih affinih momenata kontura.

Invarijantni affini momenti

Na ulaznim slikama isti tipovi prometnih znakova mogu biti zakrenuti u prostoru i različitih veličina zbog različite udaljenosti i položaja u odnosu na objektiv digitalnog fotoaparata prilikom slikanja. Osnovni problem predstavlja pronaći tip mjere koji će omogućiti ispravnu detekciju prometnih znakova nezavisno o njihovom položaju u prostoru. Takva svojstva upravo posjeduju invarijantni affini momenti. Djelovanjem affine transformacije na krug, dobivamo najjednostavniji invarijantni affin moment:

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \quad (9)$$

U formuli (9) μ_{20} , μ_{02} i μ_{11} predstavljaju centralne momente drugog reda, dok μ_{00} predstavlja centralni moment nultog reda. Iz rada [9], uzete su jednačbe za mjerenje elipsnosti (E) i trokutastosti (T) konveksne konture. Eksperimentalnim istraživanjem, dobivena je jednačba za mjerenje osmerokutnosti (O) konveksne konture.

$$E = \begin{cases} 16\pi^2 I_1 & \text{ako je } I_1 \leq \frac{1}{16\pi^2 I_1} \\ \frac{1}{16\pi^2 I_1} & \text{inače} \end{cases} \quad (10)$$

Mjera elipsnosti (E) je iz realnog intervala [0, 1]. Za savršenu elipsu vrijedi $E = 1$.

$$T = \begin{cases} 108 I_1 & \text{ako je } I_1 \leq \frac{1}{108 I_1} \\ \frac{1}{108 I_1} & \text{inače} \end{cases} \quad (11)$$

Mjera trokutastosti (T) je iz realnog intervala [0, 1]. Za savršeni trokut vrijedi $T = 1$.

$$O = \begin{cases} 15.932\pi^2 I_1 & \text{ako je } I_1 \leq \frac{1}{15.932\pi^2} \\ \frac{1}{15.932\pi^2 I_1} & \text{inače} \end{cases} \quad (12)$$

Mjera osmerokutnosti (O) je iz realnog intervala [0, 1]. Za savršeni osmerokut vrijedi $O = 1$.

Za svaku konveksnu konturu na temelju jednadžbi (9), (10), (11) i (12) određujemo vrijednosti elipsnosti (E), trokutastosti (T) i osmerokutnosti (O). Temeljem tri uvjeta i eksperimentalno određenih pragova određujemo da li pojedina konveksna kontura predstavlja pojedini prometni znak:

1. Ako je $E > 0.9979$ i $T \leq 0.89$ i $O \leq 0.9979$ tada
(Okrugli prometni znak)
2. Inače ako je $E \leq 0.9979$ i $T > 0.89$ i $O \leq 0.9979$ tada
(Trokutasti prometni znak)
3. Inače ako je $E \leq 0.9979$ i $T \leq 0.89$ i $O > 0.9979$ tada
(Osmerokutni STOP prometni znak)
4. Inače odbaci konveksnu konturu

4.2. Izdvajanje i pohrana konačnih kandidata u bazu kandidata

Pronalaženjem vanjskih kontura na binariziranoj slici, ukoliko se radi o oštećenom, izbljedjelom ili djelomično zakrivenom prometnom znaku često puta zna se dogoditi da pretvorbom pronađenih vanjskih kontura u konveksne konture, dobijemo i unutarnje konture pojedinog prometnog znaka. Primjer takve pojave možemo vidjeti na slici 18., gdje su zbog izbljedjele rubne crvene boje okruglog prometnog znaka pronađene dvije konveksne konture nakon pretvorbe vanjskih kontura.

VANJSKE KONTURE

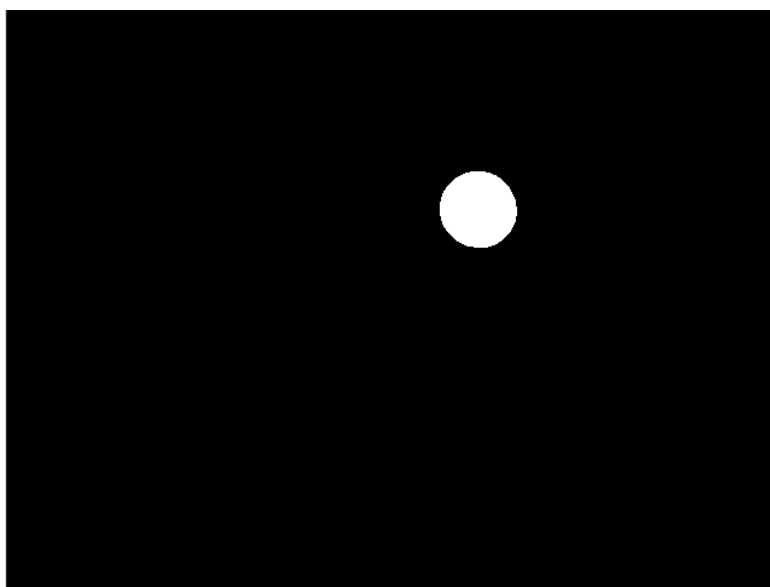


KONVEKSNE KONTURE



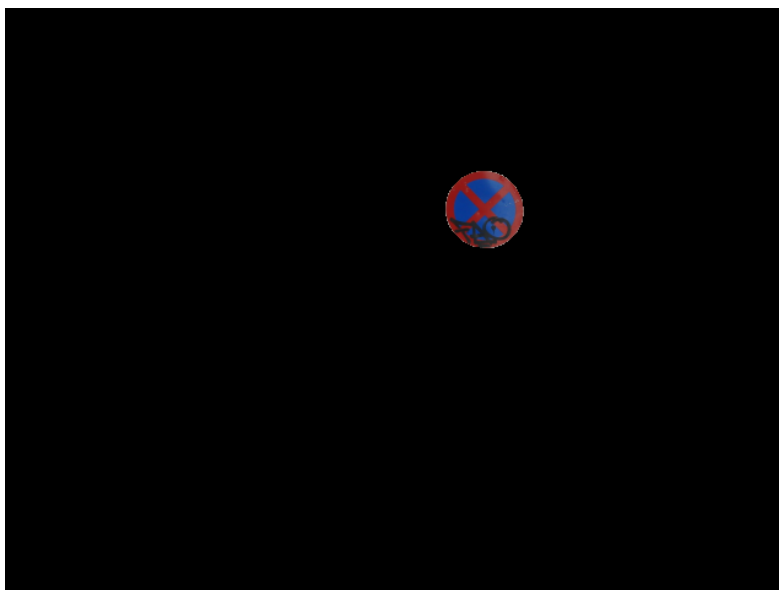
Slika 18. primjer pojave unutarnje konveksne konture nakon pretvorbe vanjskih kontura u konveksne konture

Problem pojave unutarnjih konveksnih kontura rješavamo stvaranjem maske za otklanjanje kontura. Maska za otklanjanje kontura predstavlja binarnu sliku u koju su ucrtane sve konveksne konture koje predstavljaju prometne znakove iz prethodnog podprocesa. Na dobivenoj masci obavlja se ponovni proces pronalaženja vanjskih kontura, time se eliminiraju unutarnje konveksne konture i izdvajaju se konačni kandidati. U masci za otklanjanje ispuni se površina svih konačnih kandidata, prikazano na slici 19. .



Slika 19. konačni kandidati u masci za otklanjanje sa ispunjenom unutarnjom površinom na temelju slike 17.

Provodi se logička operacija OR nad ulaznom slikom sustava iz baze slika i praznom slikom naziva ulaz za izrezivanje na područjima svijetlih piksela maske za otklanjanje. Rezultat operacije pohranjuje se u sliku ulaz za izrezivanje i prikazan je na slici 20. .



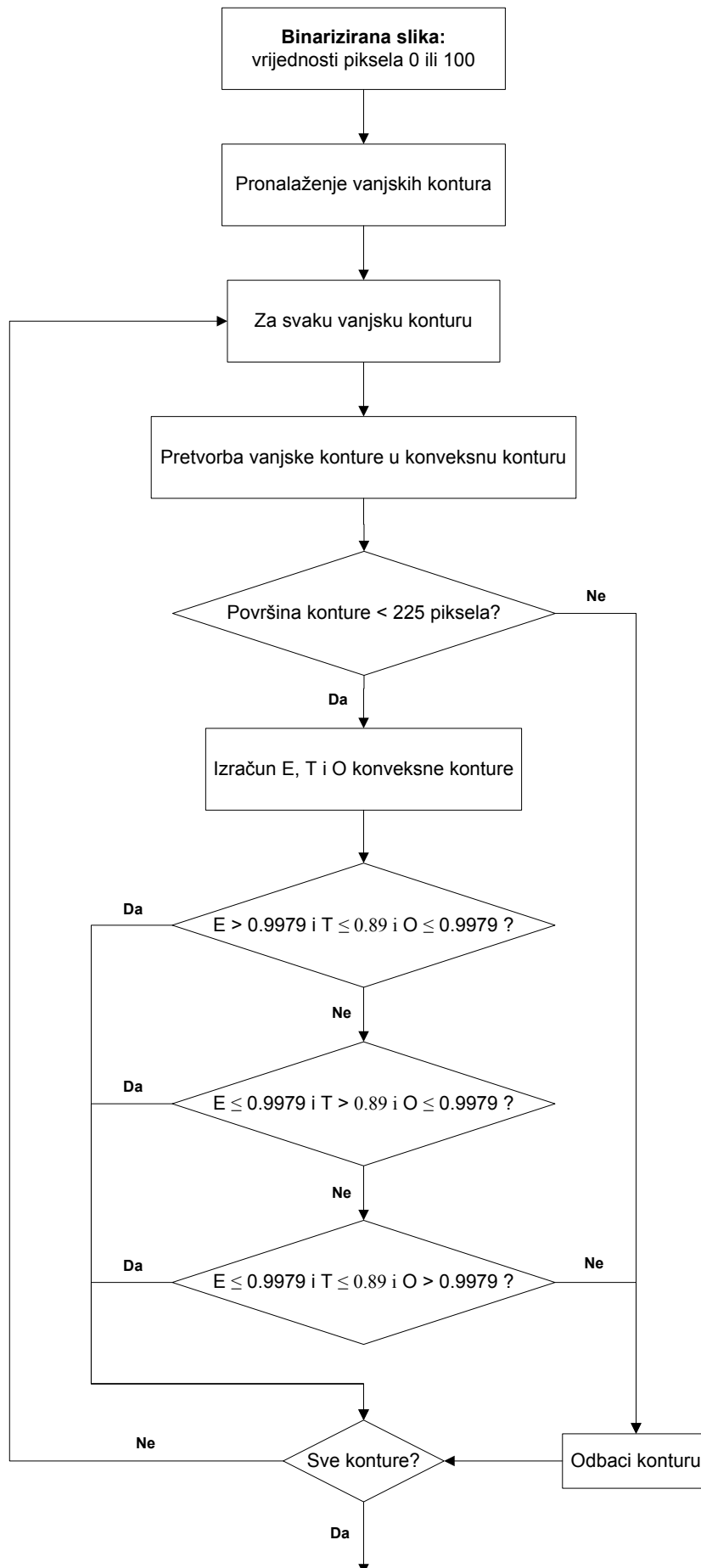
Slika 20. ulaz za izrezivanje dobiven logičkom operacijom OR nad ulaznom slikom 16. i početno praznom slikom ulaz za izrezivanje na područjima svijetlih piksela maske za otklanjanje sa slike 19.

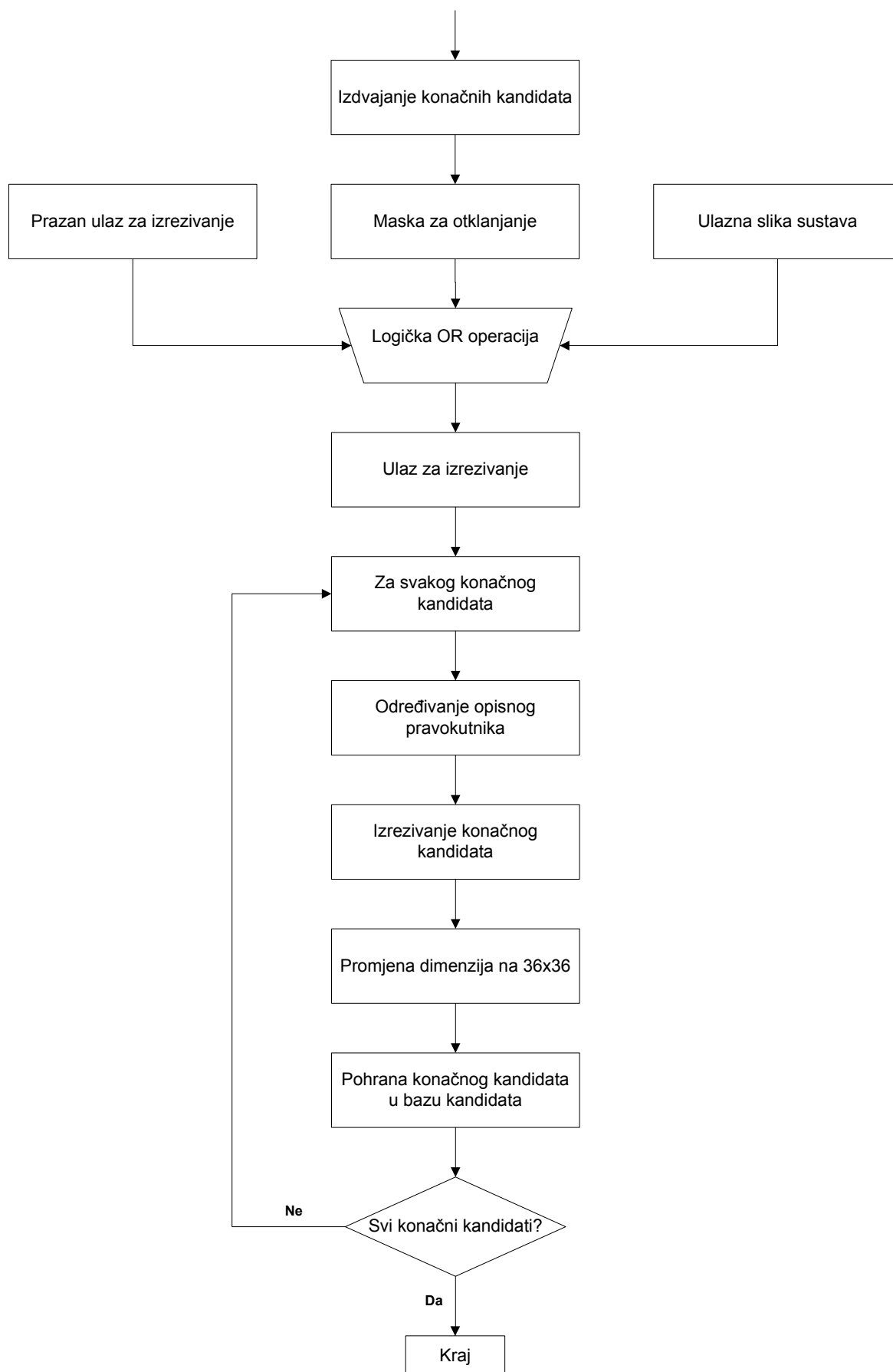
Na temelju sačuvanih informacija o konturama konačnih kandidata vrši se pronalaženje minimalnog opisanog pravokutnika oko pojedine konture konačnog kandidata. Iz ulaza za izrezivanje izrezuju se dijelovi slike koje određuju koordinate opisanog pravokutnika svakog kandidata. Dobiveni dijelovi slike skaliraju se na dimenzije 36x36 postupkom opisanim u podpoglavlju 3.1. i pohranjuju se u bazu kandidata u 24bitnom JPG, RGB formatu. Pojedini dijelovi slike mogu se dalje koristiti kao ulaz za klasifikator koji vrši prepoznavanje prometnih znakova. Primjer izdvojenog kandidata iz slike 20. prikazan je na slici 21. .



Slika 21. kandidat dimenzija 36x36 izdvojen iz slike 20. spreman za proces klasifikacije tj. prepoznavanja prometnog znaka

Kao završetak ovom poglavlju, na slici 22. dan je dijagram toka algoritma za prepoznavanje oblika objašnjenog u ovom poglavlju.





Slika 13. dijagram toka algoritma za prepoznavanje oblika

5. Uspješnost rada i analiza rezultata neuspješne detekcije ostvarenog sustava

U podpoglavlju 5.1. dana je ocjena uspješnosti rada ostvarenog sustava. U podpoglavlju 5.2. dana je analiza rezultata pogrešne detekcije ostvarenog sustava.

5.1. Ocjena uspješnosti rada ostvarenog sustava

Ocjena uspješnosti rada ostvarenog sustava temeljena je testiranjem na slikama iz baze ulaznih slika. Testiranje se provodi tako da se za svaku sliku početno postave dobri uvjeti te se ona šalje na ulaz ostvarenog sustava. Ako sustav detektira sve prometne znakove na ulaznoj slici, rezultat detekcije unosi se u tablicu testiranja. U slučaju da sustav ne detektira sve prometne znakove, provodi se postupak detekcije uz loše uvjete. U tom slučaju ukoliko se dobiju bolji rezultati, slika se proglašava slikom loših uvjeta i rezultat detekcije unosi se u tablicu testiranja. Ukupan broj prometnih znakova u tablici testiranja čine samo potpuno vidljivi prometni znakovi na ulaznoj slici površine vanjske konture veće od 225 piksela. Na temelju rezultata detekcije zapisanih u tablici testiranja (tablica 4.), formulama (13) i (14) ocijenjena je uspješnost rada ostvarenog sustava.

Tablica 4. tablica testiranja ostvarenog sustava

DOBRI UVJETI					
Redni broj slike	Broj detektiranih prometnih znakova	Broj ne detektiranih prometnih znakova	Ukupan broj prometnih znakova	Broj detekcija ne postojećih prometnih znakova	Ukupno (suma 4. i 5. stupca)
1.	2	0	2	0	2
2.	1	0	1	0	1
3.	1	0	1	0	1
4.	0	2	2	0	2
5.	1	1	2	0	2
6.	2	1	3	0	3
7.	2	0	2	0	2
8.	1	0	1	0	1
9.	1	0	1	0	1
10.	1	0	1	0	1

Redni broj slike	Broj detektiranih prometnih znakova	Broj ne detektiranih prometnih znakova	Ukupan broj prometnih znakova	Broj detekcija ne postojećih prometnih znakova	Ukupno (suma 4. i 5. stupca)
11.	1	0	1	0	1
12.	1	0	1	0	1
13.	1	0	1	0	1
14.	1	0	1	0	1
15.	2	0	2	0	2
16.	1	0	1	0	1
17.	1	0	1	0	1
18.	1	0	1	0	1
19.	1	0	1	0	1
20.	2	0	2	0	2
21.	1	0	1	0	1
22.	1	0	1	0	1
23.	3	0	3	0	3
24.	1	0	1	0	1
25.	1	0	1	0	1
26.	1	0	1	0	1
27.	1	0	1	0	1
28.	2	0	2	0	2
29.	1	0	1	0	1
30.	1	1	2	0	2
31.	2	0	2	0	2
32.	4	0	4	0	4
33.	1	0	1	0	1
34.	1	0	1	0	1
35.	1	0	1	0	1
36.	1	0	1	1	2
37.	1	0	1	0	1
38.	1	0	1	0	1
39.	1	0	1	1	2
40.	1	0	1	1	2
41.	1	0	1	0	1
42.	1	0	1	1	2
43.	2	0	2	0	2
44.	1	0	1	0	1
45.	1	0	1	0	1

Redni broj slike	Broj detektiranih prometnih znakova	Broj ne detektiranih prometnih znakova	Ukupan broj prometnih znakova	Broj detekcija ne postojećih prometnih znakova	Ukupno (suma 4. i 5. stupca)
46.	2	0	2	0	2
47.	2	0	2	0	2
48.	1	0	1	0	1
49.	1	0	1	0	1
50.	2	0	2	1	3
51.	2	0	2	0	2
52.	1	0	1	0	1
53.	3	0	3	0	3
54.	3	0	3	0	3
55.	1	0	1	0	1
56.	2	0	2	0	2
57.	1	0	1	0	1
58.	1	0	1	0	1
59.	1	0	1	0	1
60.	3	0	3	0	3
61.	2	0	2	0	2
62.	2	0	2	1	3
63.	1	0	1	0	1
64.	2	0	2	0	2
65.	1	0	1	0	1
66.	1	0	1	0	1
67.	1	0	1	0	1
68.	1	0	1	1	2
69.	1	0	1	0	1
70.	0	1	1	0	1
71.	1	0	1	0	1
72.	1	1	2	0	2
73.	1	1	2	0	2
74.	1	0	1	0	1
75.	1	0	1	1	2
76.	1	0	1	0	1
77.	1	0	1	0	1
78.	1	0	1	1	2
79.	1	0	1	0	1
80.	1	0	1	0	1

Redni broj slike	Broj detektiranih prometnih znakova	Broj ne detektiranih prometnih znakova	Ukupan broj prometnih znakova	Broj detekcija ne postojećih prometnih znakova	Ukupno (suma 4. i 5. stupca)
81.	1	0	1	0	1
82.	1	0	1	0	1
83.	1	0	1	0	1
84.	0	1	1	0	1
85.	0	1	1	0	1
86.	1	0	1	1	2
87.	1	0	1	0	1
88.	1	0	1	0	1
89.	1	0	1	0	1
90.	0	2	2	1	3
91.	1	0	1	0	1
92.	2	0	2	0	2
93.	1	0	1	0	1
94.	1	0	1	0	1
95.	1	0	1	0	1
96.	2	0	2	0	2
97.	1	0	1	0	1
98.	2	0	2	0	2
99.	1	0	1	0	1
100.	2	0	2	0	2
101.	1	0	1	0	1
102.	2	0	2	0	2
103.	1	0	1	0	1
104.	2	0	2	0	2
105.	1	0	1	0	1
106.	1	0	1	0	1
107.	1	0	1	0	1
108.	1	0	1	0	1
109.	1	0	1	0	1
110.	1	0	1	0	1
111.	1	0	1	0	1
112.	1	0	1	0	1
113.	0	1	1	0	1
114.	2	0	2	0	2
115.	1	0	1	0	1

Redni broj slike	Broj detektiranih prometnih znakova	Broj ne detektiranih prometnih znakova	Ukupan broj prometnih znakova	Broj detekcija ne postojećih prometnih znakova	Ukupno (suma 4. i 5. stupca)
116.	1	1	2	0	2
117.	1	0	1	0	1
118.	1	0	1	0	1
119.	1	0	1	0	1
120.	2	0	2	0	2
121.	1	0	1	0	1
122.	1	0	1	0	1
123.	1	0	1	0	1
124.	1	0	1	0	1
125.	2	0	2	0	2
126.	2	0	2	0	2
LOŠI UVJETI					
127.	1	0	1	0	1
128.	4	0	4	0	4
129.	1	0	1	0	1
130.	3	0	3	0	3
131.	2	0	2	0	2
132.	1	0	1	0	1
133.	1	0	1	0	1
134.	1	0	1	0	1
135.	1	1	2	0	2
136.	2	1	3	0	3
137.	2	0	2	0	2
138.	1	0	1	0	1
139.	1	0	1	0	1
140.	1	0	1	0	1
Σ	179	16	195	11	206

$$Uspješnost rada ostvarenog sustava = \frac{\sum \text{Detektiranih prometnih znakova}}{\sum \text{Ukupno}} \times 100\% \quad (13)$$

$$= \frac{179}{206} \times 100\% \approx 86,89\%$$

$$Pogreška rada ostvarenog sustava = 100\% - Uspješnost rada ostvarenog sustava \quad (14)$$

$$\approx 100\% - 86,89\% \approx 13,11\%$$

5.2. Analiza rezultata neuspješne detekcije ostvarenog sustava

Testiranjem temeljenim na bazi ulaznih slika utvrđena su tri glavna uzorka neuspješne detekcije ostvarenog sustava:

1. Skupno postavljene prometne znakove

Na pojedinom području prometnice, kada istovremeno počinje vrijediti više prometnih pravila propisanih prometnim znakovima, prometni znakovi koji ih definiraju, najčešće se postavljaju skupno na jednome stupu. Ukoliko su tako postavljene prometne znakove dovoljno daleko od objektiva digitalnog fotoaparata, prilikom resempliranja tako slikane ulazne slike zna se dogoditi da zbog premalog bijelog obruba znaka, dolazi do međusobnog spajanja vanjskih crvenih površina prometnih znakova. U tom slučaju ostvareni sustav jednom konturom opisuje više prometnih znakova, pa njihova detekcija nije moguća. Primjer takve pojave prikazan je na slici 14., gdje je sustav na ulaznoj slici detektirao samo jedan od tri prometna znaka.



Slika 14. primjer neuspješne detekcije skupno postavljenih prometnih znakova

2. Oštećenje vanjske crvene površine prometnog znaka

Ostvareni sustav ne može vršiti detekciju prometnih znakova značajno oštećene vanjske crvene površine. U takva oštećenja ubrajamo: oštećenje uslijed šaranja, preljepljivanja, zaklanjanja ili otkidanja značajnog dijela vanjske površine prometnog znaka i oštećenje uslijed značajne izbljedjelosti vanjske crvene

površine prometnog znaka. Na slici 15. prikazan je primjer neuspješne detekcije zbog preljepljivanja značajnog dijela vanjske crvene površine prometnog znaka. Na slici 16. prikazan je primjer neuspješne detekcije zbog značajne izbljedjelosti vanjske crvene površine prometnog znaka, prepoznat je jedan od dva prometna znaka.



Slika 15. primjer neuspješne detekcije zbog značajnog preljepljivanja vanjske crvene površine prometnog znaka



Slika 16. primjer neuspješne detekcije zbog izbljedjele vanjske crvene površine prometnog znaka

3. Stapanje prometnog znaka s okolinom

Okolina u pozadini prometnog znaka ponekad može poprimiti intenzitete boje crvenog spektra koje se koriste za isticanje prometnih znakova koje ostvareni sustav detektira. U takvim slučajevima prometni znakovi nisu dovoljno istaknuti s obzirom na okolinu i govorimo o pojmu stapanja prometnog znaka s okolinom. Kako je jedan od ključnih podprocesa za detekciju prometnih znakova podproces izdvajanja crvene boje, u takvim slučajevima nije moguće ostvariti izdvajanje prometnog znaka i prometni znak ne biva detektiran. Na slici 17. prikazan je primjer neuspješne detekcije dva prometna znaka zbog njihovog stapanja s okolinom. Na slici 18. prikazan je izlaz iz podprocesa izdvajanja crvene boje za navedeni primjer.



Slika 17. primjer neuspješne detekcije zbog stapanja prometnih znakova sa okolinom



Slika 18. izlaz iz podprocesa izdvajanja crvene boje za sliku 17.

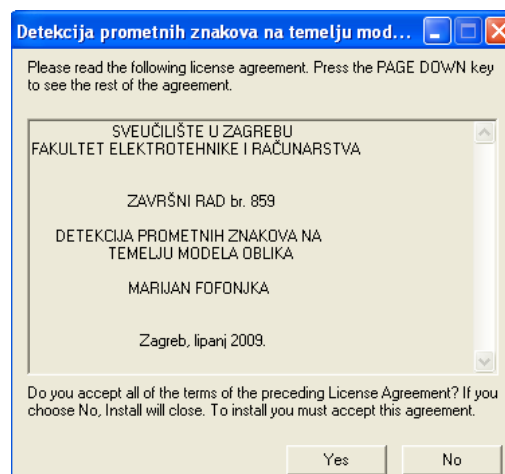
6. Upute za instalaciju i korištenje programski ostvarenog sustava

Programski ostvareni sustav kompatibilan je sa Windows, Linux i Mac OS operacijskim sustavima. U podpoglavlju 6.1. objašnjen je instalacijski postupak za Windows operacijski sustav. Instalacija programski ostvarenog sustava pod Linux i Mac OS operacijskim sustavima zasniva se na kompajliranju source koda projekta Program.cpp povezanog sa OpenCV bibliotekom dostupnom na [3]. Više informacija o samom postupku dostupno je na [11]. U podpoglavlju 6.2. dane su upute za korištenje programski ostvarenog sustava.

6.1. Instalacijski postupak za Windows operacijski sustav

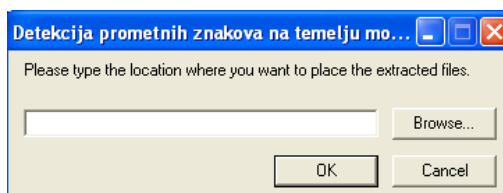
Postupak instalacije programski ostvarenog sustava pod Windows operacijskim sustavom dan je u slijedećih pet koraka:

1. Sa CD-a priloženog uz ovaj rad potrebno je pokrenuti datoteku *Setup.exe*
2. U otvorenom prozoru prikazanom na slici 19. potrebno je prihvatiti licencni ugovor pritiskom na tipku *Yes*, u suprotnome nije moguć nastavak instalacije.



Slika 19. licencni ugovor programski ostvarenog sustava

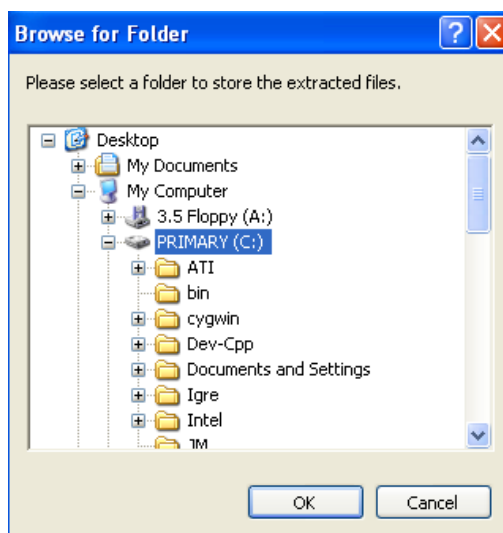
3. U otvorenom prozoru prikazanom na slici 20. odabire se direktorij za instalaciju programski ostvarenog sustava:



Slika 20. odabir direktorija za instalaciju programski ostvarenog sustava

Korištenje postojećeg direktorija na disku

Ukoliko se želi koristiti već postojeći prethodno stvoreni direktorij za instalaciju, potrebno je pritisnuti tipku *Browse* i u novo otvorenom prozoru prikazanom na slici 21. odabrati put do direktorija na disku te potvrditi odabir pritiskom na tipku *OK*. Kako bi započeo postupak instalacije u glavnom prozoru potrebno je pritisnuti tipku *OK*.

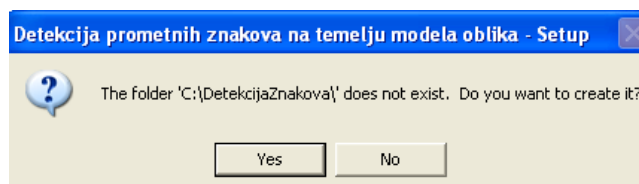


Slika 20. odabir puta do postojećeg direktorija na disku

Kreiranje novog direktorija na disku

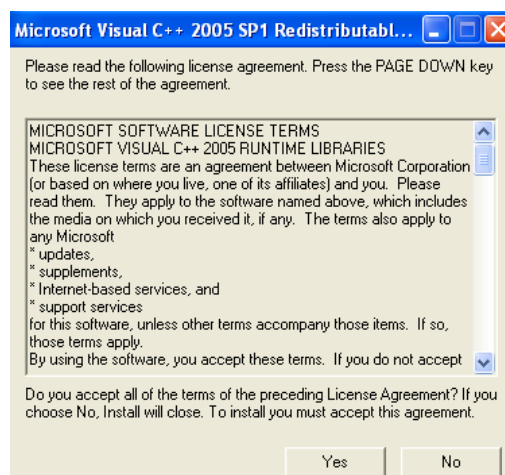
Ukoliko se želi kreirati novi direktorij za instalaciju, u kućicu prozora prikazanog na slici 20., potrebno je upisati put do novoga direktorija uključujući i njegov naziv. Primjerice ukoliko se želi kreirati novi direktorij imena *DetekcijaZnakova* na C disku, u kućicu se upisuje *C:\DetekcijaZnakova*. Pritiskom na tipku *OK* otvara se novi prozor prikazan na slici 21. u kojemu je potrebno potvrditi

stvaranje novog direktorija i započeti postupak instalacije pritiskom na tipku Yes.



Slika 21. kreiranje novog instalacijskog direktorija

4. Za korištenje programski ostvarenog sustava potrebno je instalirati programski paket *Microsoft Visual C++ 2005 SP1 Redistributable Package*. Ukoliko je prethodno instaliran navedeni programski paket u novo otvorenom prozoru prikazanom na slici 21., potrebno je pritisnuti tipku *No*. U suprotnome potrebno je prihvatiti licencni ugovor pritiskom na tipku *Yes* i pričekati završetak instalacije.



Slika 22. licencni ugovor programskog paketa

Microsoft Visual C++ 2005 SP1 Redistributable Package

5. Programski sustav uspješno je instaliran i nalazi se u odabranom instalacijskom direktoriju iz koraka 3. .

6.2. Upute za korištenje programski ostvarenog sustava

Instalacijom programski ostvarenog sustava, u instalacijskom direktoriju stvorena je datoteka naziva *DetekcijaZnakova.exe* . Navedena datoteka predstavlja programski ostvaren sustav. Za ispravno korištenje programski ostvarenog sustava prilikom poziva datoteke, potrebno je predati tri argumenta:

1. ulazna slika nad kojom se provodi detekcija

Ulazna slika nad kojom se provodi detekcija mora biti u boji, proizvoljnih dimenzija i u BMP, DIB, JPEG, JPG, JPE, PNG, PBM, PGM, PPM, SR, RAS, TIFF ili TIF formatu. Ulazna slika predaje se kao prvi argument programski ostvarenog sustava u obliku puta do odgovarajuće slike. Primjerice ukoliko se na ulaz programski ostvarenog sustava šalje ulazna slika u boji i u JPG formatu naziva Slika001 koja se nalazi na C disku u direktoriju Baza_ulaznih_slika, tada će prvi predani argument biti: C:\Baza_ulaznih_slika\Slika001.jpg .

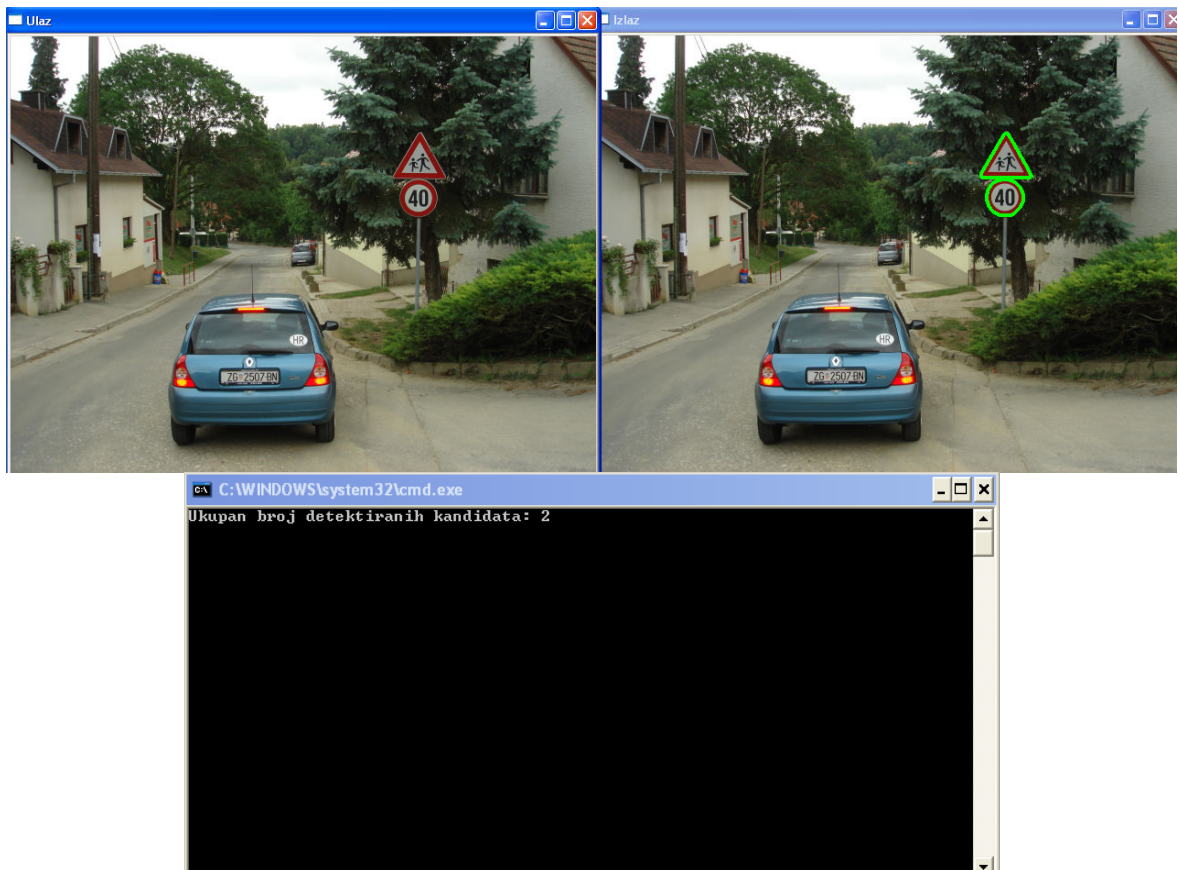
2. tip uvjeta ulazne slike

Drugi argument programski ostvarenog sustava predstavlja tip uvjeta u kojima je slikana ulazna slika. Uvjeti mogu biti dobri ili loši, više o značenju pojmova dobri uvjeti i loši uvjeti dano je u podpoglavlju 3.2. . Ako se radi o dobrim uvjetima kao drugi argument predaje se *d* . Ako se radi o lošim uvjetima kao drugi argument predaje se *l* .

3. izlazni direktorij kandidata

Treći argument programski ostvarenog sustava predstavlja put do prethodno stvorenog direktorija baze kandidata. Više o bazi kandidata dano je u podpoglavlju 2.5. . Primjerice ukoliko se baza kandidata nalazi na C disku u direktoriju Baza_kandidata, tada će treći predani argument biti: C:\Baza_kandidata\ . Poziv programski ostvarenog sustava za prethodne argumente obavlja se iz konzole operacijskog sustava. U konzoli se je prethodno potrebno pozicionirati u instalacijski direktorij programski ostvarenog sustava, a zatim se pokreće naredba: *DetekcijaZnakova.exe C:\Baza_ulaznih_slika\Slika001.jpg d C:\Baza_kandidata* Pokretanjem naredbe započinje rad programski ostvarenog sustava i kao rezultat procesa detekcije (slika 23.) dobivaju se dva prozora: *ulaz* i *izlaz* i zapis ukupnog

broj pronađenih kandidata u konzoli. Prozor *ulaz* predstavlja ulaznu sliku skaliranu na dimenzije 640x480 prije procesa izjednačavanja histograma ukoliko se radi o lošim uvjetima. Prozor *izlaz* predstavlja rezultat detekcije sustava, gdje su na ulaznoj slici zelenom bojom označeni rubovi pronađenih kandidata. Pronađeni kandidati pohranjeni su u direktoriju predanom kao treći argument sustava u formatu opisanom u podpoglavlju 2.5. . Za kraj rada sustava potrebno je pritisnuti bilo koju tipku na tipkovnici.



Slika 23. izlaz programski ostvarenog sustava

7. Zaključak

Zbog povećanja sigurnosti i obaviještenosti pojedinca koji upravlja vozilom i drugih sudionika u prometu, sustavi za raspoznavanje prometnih znakova prijeko su potreban alat u prometnim vozilima današnjice. Sustav za detekciju prometnih znakova ostvaren u ovome radu zasniva se na metodi detekcije prometnih znakova na temelju modela oblika. Temeljem testiranja sustava pokazana je dosta visoka pouzdanost ove metode detekcije. Ostvareni sustav na temelju 140 slika, slikanih pod različitim vremenskim uvjetima postiže uspješnost detekcije od 86,89% . Ključni dio u kojemu leži uspješnost detekcije sustava, zasigurno su granične vrijednosti parametara selekcije za pojedini algoritam. Kako sustav sadrži dosta veliki broj selekcijskih parametara, poboljšanje detekcije moguće je uz iscrpno testiranje na većoj bazi ulaznih slika i korigiranje istih. Još jedna od mogućnosti za bolji rad sustava zasigurno je i korištenje različitih vrijednosti graničnih parametara selekcije algoritama za različite vremenske uvjete. Obzirom na dosta brzo izvođenje i slabu osjetljivost na promjene vremenskih uvjeta, kutove snimanja i manja oštećenja površine prometnih znakova, uz manje preinake ostvareni sustav moguće je iskoristiti kao dio autonomnog sustava koji vrši prepoznavanje prometnih znakova u realnom vremenu. Ključne preinake u sustavu koje je time potrebno izvršiti su nadopuna algoritama za izdvajanje boje i prepoznavanje oblika. Algoritam za izdvajanje boje potrebno je prilagoditi za izdvajanje boja svih skupina prometnih znakova. U algoritam za prepoznavanje oblika potrebno je ugraditi nove mjere za detekciju svih oblika prometnih znakova. Najčešći uzroci neuspješnosti detekcije ostvarenog sustava su stapanje boje prometnog znaka s okolinom, značajna oštećenja površine ili spajanje površina skupno postavljenih prometnih znakova. Budući da navedene uzroke nije moguće riješiti postojećim algoritmima, potrebno je uvesti dopunske metode u rad sustava.

Vlastoručni potpis:

Marijan Tefenjka

8. Literatura

- [1] Hrvatski sabor: *Zakon o sigurnosti prometa na cestama*, 30. svibnja 2008. ,
<http://narodne-novine.nn.hr/clanci/sluzbeni/339713.html>, ožujak 2009.
- [2] Microsoft: *Microsoft Visual Studio 2008*,
<http://www.microsoft.com/visualstudio/en-us/default.mspx>, ožujak 2009.
- [3] Intel: *OpenCV*, <http://sourceforge.net/projects/opencvlibrary/>, ožujak 2009.
- [4] The Stanford Racing Team: *Robot car Stanley*,
<http://www.cs.stanford.edu/group/roadrunner/stanley.html>, ožujak 2009.
- [5] Fleyeh H. : *Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach*, Machine Vision Applications (MVA2005 IAPR), Tsukuba Japan, 2005.
- [6] Perez H. i Koach C. : *Toward color image segmentation in analog VLSI: Algorithm and hardware*, Int. J. of Computer Vision, vol. 12, str. 17-42, 1994.
- [7] CyLog Software: *RGB to HSV and vice versa v.1.22*,
<http://www.cylog.org/graphics/rgb2hsv.jsp>
- [8] Graham R. i Yao F. : *Finding the convex hull of a simple polygon*, Journal of Algorithms, vol. 4, str. 324-331, 1983.
- [9] Rosin P. : *Measuring shape: ellipticity, rectangularity, and triangularity*, Machine Vision and Applications, vol. 14, str. 172-184, 2003.
- [10] Bradski G. i Kaehler A. : *Learning OpenCV*, The O'Reilly Media, Sebastopol, 2008.
- [11] OpenCV Wiki: *Installing OpenCV*,
<http://opencv.willowgarage.com/wiki/Welcome#Welcome.2BAC8-Introduction.InstallingOpenCV>, ožujak 2009.

Detekcija prometnih znakova na temelju modela oblika

Sažetak

U ovome radu razmatramo detekciju okruglih, trokutastih i osmerokutnih prometnih znakova opasnosti i izričitih naredbi sa crvenim okvirom u slikama slikanim u mirnom položaju i iz automobila u pokretu. Predstavljamo implementaciju metode za detekciju temeljenu na modelu oblika. Rad ostvarenog sustava temelji se na dvije glavne faze: obrada slike i prepoznavanje oblika. U fazi obrade slike, ulazna RGB slika pretvorena je u HSV sustav boja i segmentirana temeljem skupa neizrazitih pravila. Neizrazita pravila korištena su za izdvajanje boje prometnog znaka. Faza prepoznavanja oblika prometnog znaka temeljena je na tri mjere: elipsnost, trokutastost i osmerokutnost, razvijena su neizrazita pravila za raspoznavanje oblika prometnog znaka. Konačna odluka sustava temeljena je na kombinaciji boje i oblika prometnog znaka. Sustav je testiran u različitim testnim uvjetima sa konačnom performansama od oko 87%.

Ključne riječi

detekcija na temelju modela oblika, prometni znakovi, segmentacija boje, računalni vid, neizrazita pravila, detekcija objekata

Model based traffic signs detection

Abstract

In this work we consider detection of round, triangular and octangular traffic signs with a red frame in images taken on foot and from a moving vehicle. We present an implementation of a model based detection. The implemented system is based on two major stages: image processing and shape recognition. In image processing stage the input RGB image was converted into HSV color space, and segmented using a set of fuzzy rules. The fuzzy rules are used to extract the color of a traffic signs. Shape recognition stage is based on three shape measures: ellipticity, triangularity and octangonality, fuzzy rules were developed to determine the shape of the sign. The final decision of the system is based on combination of both the color and shape of a sign. The system was tested in variety of testing conditions giving an overall performance of approximately 87%.

Key words

model based detection, object detection, traffic signs, color segmentation, fuzzy rules, computer vision