

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1048

**Detekcija prometnih znakova na temelju
boje i oblika**

Marijana Milas

Zagreb, siječanj 2010.

Zahvala :

Zahvaljujem mentoru prof.dr.sc. Zoranu Kalafatiću na uloženom trudu, pomoći, velikom zanimanju, i susretljivosti prilikom izrade ovoga rada.

Zahvaljujem mojim roditeljima i sestri na popriličnom razumijevanju i potpori tijekom cijelog postupka. Veliko hvala Marku, Matiji i Ana-Mariji, bez kojih ne bih uspjela. Također bih se htjela zahvaliti Danijelu, Tinu, Ines i Martinu, te Zvonimiru, Lauri i Niki što su uvijek bili tu negdje da mi pomognu kad je bilo najteže.

Posvećujem ovaj rad jednoj posebnoj osobi, mojoj majci Snježani...

Sadržaj

1.	Uvod	1
1.1	Skup znakova obavijesti za vođenje prometa:	3
1.2	Načini detekcije prometnih znakova opisani u literaturi	6
2.	Priprema za implementaciju programskog sustava	8
2.1	Početak rada sa OpenCV bibliotekom u Visual Studiu 2005	9
2.2	Baza ulaznih slika za učenje i testiranje sustava	11
3.	Obrada slike	12
3.1	Izdvajanje žute boje sa ulazne slike	13
3.1.1	Usporedba sustava boja RGB - HSV	16
3.1.2	Uvođenje histograma	18
3.1.3	Prepoznavanje žute boje na ulaznoj slici	22
3.1.4	Uklanjanje šuma sa slike	23
3.2	Prepoznavanje pravokutnog oblika prometnog znaka	25
4.	Rezultati i uspješnost dobivenog sustava	27
4.1	Testiranje.....	27
4.2	Analiza dobivenih rezultata.....	31
4.2.1	Stapanje okoline sa prometnim znakom.....	31
4.2.2	Loša kvaliteta prometnog znaka na ulaznoj slici	32
4.2.3	Prepoznavanje više od jednog znaka na slici	33
5.	Upute za korištenje ostvarenog sustava	34
6.	Zaključak.....	35
7.	Literatura.....	36

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.

Prema Zakonu o sigurnosti prometa na cestama (ZOSPC-u) od 14. ožujka 2005. godine, prometni znakovi su: znakovi opasnosti, znakovi izričitih naredbi, znakovi obavijesti, i znakovi obavijesti za vođenje prometa s dopunskom pločom (koja je sastavni dio prometnog znaka, i koja pobliže određuje značenje prometnog znaka), ili bez nje. To su još i promjenjivi prometni znakovi, prometna svjetla i svjetlosne oznake, te oznake na kolniku i drugim površinama.

Potrebno je oblikovati programski sustav za detekciju određenih prometnih znakova - znakova obavijesti za vođenje prometa. To su zelene, žute i plave pravokutne ploče, koje označavaju:

- raskrižje;
- raskrižje kružnog oblika;
- predputokaz;
- putokaznu ploču;
- najavu naseljenog mjesta.

Znakovi obavijesti za vođenje prometa obavještavaju sudionike u prometu o pružanju cestovnih smjerova, rasporedu odredišta, i vođenju prometa prema njima, križanjima i čvorištima na određenom smjeru ceste i udaljenostima do odredišta.

Osnovna boja znakova obavijesti za vođenje prometa je:

1. na autocestama zelena sa simbolima i natpisima bijele boje;
2. na brzim cestama plava sa simbolima i natpisima bijele boje;
3. na državnim i ostalim cestama žuta sa simbolima i natpisima crne boje;
4. za dijelove gradova, naselja i značajne objekte bijela sa simbolima i natpisima crne boje.

Na istom znaku (ploči) mogu se na osnovnu podlogu umetnuti podloge odgovarajućih boja ovisno o vrsti ceste koja vodi do naznačena odredišta.

U okviru ovoga rada želimo dobiti sustav koji će na ulaznoj slici pronaći, odnosno detektirati žuti pravokutni prometni znak. To su znakovi D skupine prometnih znakova, a ulazne slike su snimljene iz automobila u pokretu. Metode uz pomoć kojih ćemo to ostvariti su ujedno i dva glavna procesa implementacije sustava za detekciju - obrada slike i prepoznavanje boje znaka na njoj, te prepoznavanje oblika znaka. Prikaz skupa prometnih znakova koje ostvareni sustav detektira dan je u poglavlju 1.1.

Rad je strukturiran kako slijedi. U poglavlju 2. dan je pregled ostvarenja programskog sustava, uz opis njegovih ulaznih i izlaznih dijelova, te pregled korištenih programskih alata za programsko ostvarenje. U poglavlju 3. dan je opsežan opis rada i korištenih metoda implementacije. U poglavlju 4. dani su rezultati uspješnosti i analiza neuspješnih detekcija ostvarenog sustava na temelju baze ulaznih slika. U poglavlju 5. dane su upute za instalaciju i korištenje sustava. U poglavlju 6. je zaključak. U poglavlju 7. dan je pregled korištene literature.

1.1 Skup znakova obavijesti za vođenje prometa:

Predputokaz za izlaz s autoceste ili brze ceste s oznakom izlaza (D01):



Predputokaz za čvorište autocesta s oznakom čvorišta (D02):



Raskrižje (D03):



Raskrižje (D04):



Raskrižje kružnog oblika (D05):



Predputokazna ploča (D06):



Predputokazna ploča (D07):



Predputokazna ploča (D08):



Predputokaz za izlaz (D09):



Predputokaz za izlaz (D10):



Predputokaz za izlaz (D11):



Putokazna ploča (D12):



Putokaz na portalu iznad jedne prometne trake (D13):



Putokaz na portalu iznad jedne prometne trake (D14):



Putokaz na portalu iznad dvije prometne trake (D15):



Putokaz na portalu iznad dvije prometne trake (D16):



Potvrda smjera (D17):



1.2 Načini detekcije prometnih znakova opisani u literaturi

Gradijentni postupci detekcije rubova

Jedan od najuspješnijih i najkorištenijih algoritama gradijentnih postupaka je Cannyjev algoritam [13]. Detekcija rubova ovim algoritmom odvija se u više koraka. Prvo se obavlja konvolucija slike s Gaussovom maskom kako bi se slika usrednjila i uklonio šum, a zatim se određuju područja maksimuma prve derivacije. Pri tome se u svakoj točki slike označava u kojem smjeru je najveći iznos gradijenta, i ta se informacija koristi u kasnijim fazama algoritma. Zatim se obavlja postupak *non-maximal suppression*, koji svaku vrijednost gradijenta koja nije lokalni maksimum postavlja na 0, i pri tome koristi informaciju o smjeru gradijenta. Pronalaze se spojeni skupovi rubnih točaka. U posljednjem koraku se svaka točka koje je u listi rubnih točaka, i ima iznos gradijenta veći od većeg praga proglašava valjanom točkom ruba. Zatim se i sve točke koje su spojene na valjane rubne točke, a imaju vrijednost gradijenta veću od nižeg praga također proglašavaju valjanim rubnim točkama.

Križna korelacija za prepoznavanje objekata na slici

Korelacija [14] (lat. con = sa, relatio = odnos) predstavlja odnos ili međusobnu povezanost između različitih pojava predstavljenih vrijednostima dviju varijabli. Pri tome povezanost znači da je vrijednost jedne varijable moguće sa određenom vjerojatnošću predvidjeti na osnovu saznanja o vrijednosti druge varijable. Ova je metoda zato dobro iskorištena za prepoznavanje objekata, točnije prometnih znakova, i to na način da se uspoređuju sličnosti dvije slike. Jedna slika (realna) dobiva se iz ulazne, a druga iz skupa idealnih slika znakova. Kao izlaz dobivamo koeficijent koji opisuje koliko su te dvije slike slične. Rezultat je uspješno ili neuspješno prepoznavanje određenog znaka na slici.

Neuronske mreže

Raspoznavanje prometnih znakova se može učiniti i primjenom neuronskih mreža [15]. Pod pretpostavkom da su položaji znakova i njihove veličine već određeni nekom drugom metodom, neuronsku mrežu se uči da prepozna te prometne znakove na ulaznim slikama. Osim unaprijed pripremljenih prometnih znakova, neuronsku mrežu se uči da prepozna i nove, dotad neviđene znakove.

Houghova transformacija za kružnice

Houghova transformacija [16] također omogućuje detekciju rubova objekata koji se nalaze na slici. Pretpostavka je da su prije same transformacije detektirani rubovi. Kombiniranjem rubnih piksela detektiraju se linije, a kombiniranjem linija i složenije konture. Ukoliko tražimo objekte kružnog oblika, i neke određene boje, svaki piksel te dotične boje glasuje za sve kružnice kojima bi mogao pripadati. Ukoliko na slici imamo kružnicu, za nju će glasati svaki njezin piksel, i tako će nam je označiti u akumulatorskom polju. Na ovaj ćemo način, uz pomoć kružnica od kojih se sastoji obrub znaka, pokušati strojno pronaći takve znakove.

Strojno učenje

Teško je zamisliti nešto fascinantnije od automatskog sustava koji unaprijeđuje svoje vlastite performanse kroz iskustvo. Proučavanje i izgradnja sustava koji mogu učiti iz podataka važno je znanstveno i komercijalno. Stoga je i strojno učenje [17] jedna od metoda dobro iskorištena za detekciju prometnih znakova. I to na način da se sustavu predaju različite slike znakova, znakovi koji se razlikuju po boji, obliku, sadržaju. Sustav tako uči što točno mora prepoznati na ulaznoj slici, pa to na kraju i čini.

2. Priprema za implementaciju programskog sustava

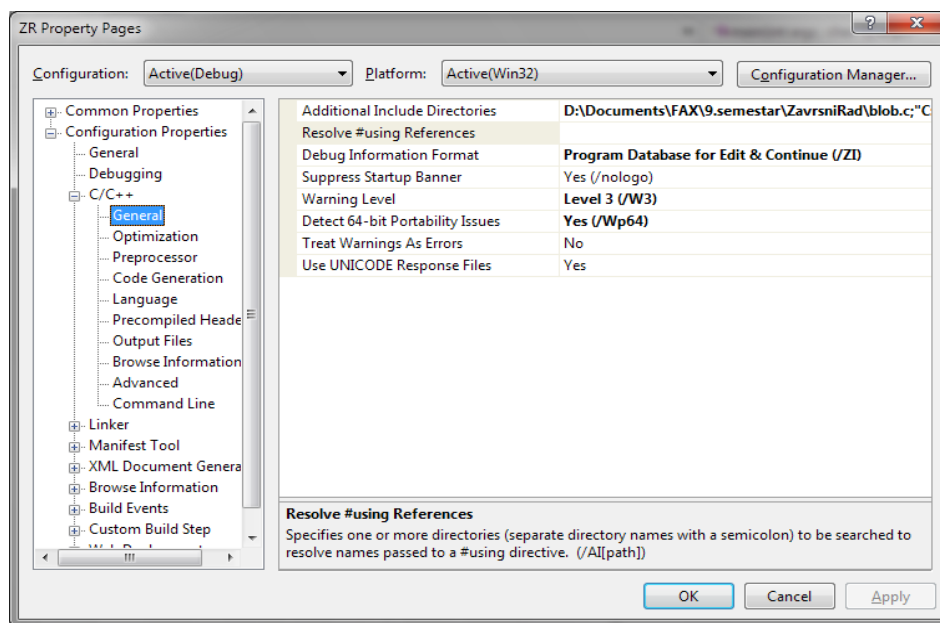
Programski sustav za detekciju prometnih znakova ostvaren je pomoću razvojnog programskog alata Microsoft Visual Studio 2005, i biblioteke za računalni vid OpenCV [6]. Po samom početku rada bilo je potrebno upoznati se sa bibliotekom OpenCV. Završni rad znači učenje, proučavanje, snalaženje i istraživanje jednog određenog područja rada, a to područje ovoga rada je detekcija prometnih znakova na slici.

OpenCV (Open Source Computer Vision) je biblioteka programskih funkcija za računalni vid. Službeno ju je pokrenula 1999. godine tvrtka Intel, besplatna je za komercijalni i istraživački rad, ali pod *open source* BSD (Berkeley Software Distribution) licencom. Uglavnom je napisana u programskom jeziku C, a radi na operacijskim sustavima Linux, Mac OS i Windows.

Glavni cilj programera pri stvaranju biblioteke OpenCV 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 višezgrednih 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 sigurnosnih sustava, ostvarivanje stereo vida u robotici. Pošto su područja računalnog vida i strojnog učenja usko povezana, OpenCV sadrži i biblioteku za strojno učenje MLL (Machine learning library), koja obuhvaća algoritme za izgradnju klasifikatora temeljenih na statističkim metodama ili postupcima grupiranja (engl. clustering). Mnogi poznati sustavi današnjice na području računalnog vida izgrađeni su na OpenCV biblioteci. Primjer jednog takvog je robot „Stanley“ sa sveučilišta Stanford, koji je 2005. godine pobijedio na pustinjskoj off-road utrci DARPA (Defense Advanced Research Projects Agency) Grand Challenge, i čiji je tim nagrađen sa dva milijuna dolara.

2.1 Početak rada sa OpenCV bibliotekom u Visual Studiu 2005

Prije samog početka programiranja bilo je potrebno namjestiti program Visual Studio 2005 kako bi ispravno radio s bibliotekom OpenCV [7]. Po završetku stvaranja novog projekta, trebalo je modificirati opcije istog projekta. To je učinjeno na sljedeći način:

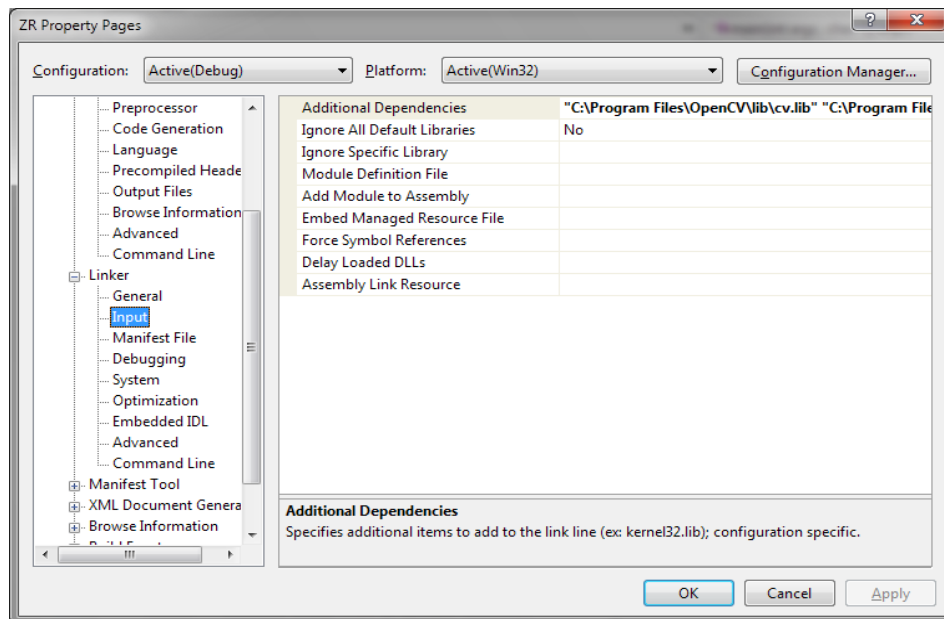


Slika 1 : Dodavanje direktorija sa datotekama koje su potrebne za ispravno korištenje biblioteke OpenCV

Ulaskom u General Configuration Properties, morali smo dodati put do direktorija gdje se nalaze datoteke koje smo koristili. Additional Include Directories su sada :

```
„C:\Program Files\OpenCV\cv\include\“;  
„C:\Program Files\OpenCV\cvaux\include\“;  
„C:\Program Files\OpenCV\cxcore\include\“;  
„C:\Program Files\OpenCV\otherlibs\cvcam\include\“;  
„C:\Program Files\OpenCV\otherlibs\highgui\“
```

Nakon toga je bilo potrebno dodati put do svih datoteka cv libs.



Slika 2 : Dodavanje .lib datoteka

Additional Dependencies su sada :

„C:\Program Files\OpenCV\lib\cv.lib“

„C:\Program Files\OpenCV\lib\cvaux.lib“

„C:\Program Files\OpenCV\lib\cxcore.lib“

„C:\Program Files\OpenCV\lib\highgui.lib“

2.2 Baza ulaznih slika za učenje i testiranje sustava

Bazu ulaznih slika potrebnih za implementaciju sustava za detekciju prometnih znakova čine fotografije izrezane iz video materijala onih kadrova gdje se pojavljuju prometni znakovi koji su potrebni. Snimke su napravljene iz vozila u pokretu na području Zagreba i okolice, te na području slavonskih cesta. Fotografije su u boji, i napravljene su u različitim vremenskim uvjetima i različitim dijelovima dana. Upravo zbog toga su i fotografije različite kvalitete, neke lošije, neke bolje, pa je stoga bilo potrebno prije svega odrediti interval žute boje, s obzirom da žuta boja nije idealna. Kad kažemo idealna boja, mislimo na vrijednosti (255, 255, 0) koje označavaju žutu boju u RGB sustavu boja. Baza ulaznih slika sastoji se od 245 slika u .bmp formatu, rezolucije 720x576 piksela.

Izdvajanje prometnih znakova se odvijalo u dva koraka:

- a) prepoznavanje boje - žuta;
- b) prepoznavanje oblika - pravokutna ploča.

CyLog software RGB to HSV and vice versa v1.22 [11] je program koji sadrži opciju Screen Pick, pomoću koje je na izravan način moguće dobiti vrijednost pojedinog piksela neke ulazne slike. Program je korišten kao provjera pri određivanju graničnih vrijednosti za izdvajanje informacije pojedinog kanala. Ono što nas zanima je samo interval žute boje, kakve su prometni znakovi koje moramo moći detektirati.

3. Obrada slike

Nakon što je učitana slika na kojoj želimo pronaći dotični prometni znak, krenula je razrada same ideje korak po korak. Cilj je bio dobiti sustav koji će na ulaznoj slici pronaći, odnosno detektirati žuti pravokutni prometni znak. Prije početka je trebalo napraviti bazu ulaznih slika, koje će služiti kao testni uzorci za sustav. Između skupljenih slika izdvojene su one na kojima se pojavljuju prometni znakovi D skupine. Tu skupinu čine žuti, zeleni i plavi pravokutni znakovi za vođenje prometa. Kako među slikama nije bilo zelenih i plavih ploča, sustav za detekciju je izrađen na prometnim znakovima žute boje, i napravljena je baza za učenje od 245 slika.



Slika 3 : Primjer ulaznih slika

3.1 Izdvajanje žute boje sa ulazne slike

Kako bismo pronašli područje prometnog znaka na ulaznoj slici, najprije smo morali izdvojiti žutu boju sa ulazne slike. Ta nakupina žute boje bi označavala mjesto gdje je naš potencijalni prometni znak. `cvGet2D` je OpenCV funkcija koja vraća vrijednosti piksela slike. Trebaju nam samo vrijednosti žute boje gdje se nalazi prometni znak, odnosno interval koji pokriva područje žute boje gdje je znak. Stoga su pomoću `x` i `y` koordinata znaka, te visine i širine, i funkcije `cvGet2D` dobivene vrijednosti piksela samo na području prometnog znaka. Nakon što su dobivene B, G i R (plava, zelena, crvena) vrijednosti **određenog dijela slike**, prebrojano je koliko ima piksela svake od vrijednosti B komponente, koliko od G, a koliko od R komponente. Na taj smo način dobili histograme za svaku komponentu. Vrijednosti BGR komponenti kreću se od 0-255, te su zbog tog razloga napravljena tri polja - polje B, polje G, i polje R od 256 elemenata. Kako bi izbrojali piksele, ta smo polja onda punili kako smo funkcijom prolazili kroz vrijednosti piksela dijela ulazne slike. Kad bismo naišli na piksel koji ima vrijednost B komponente npr. 24, povećali bismo brojač te vrijednosti. Na 25. mjestu u polju B bismo upisali vrijednost brojača, a ako je to prvi piksel sa tom vrijednošću, ovdje bi sada pisao broj 1. Tako je popunjeno cijelo polje B, a na identičan način polja G i R, koja uz polje B označavaju komponente RGB sustava boja.

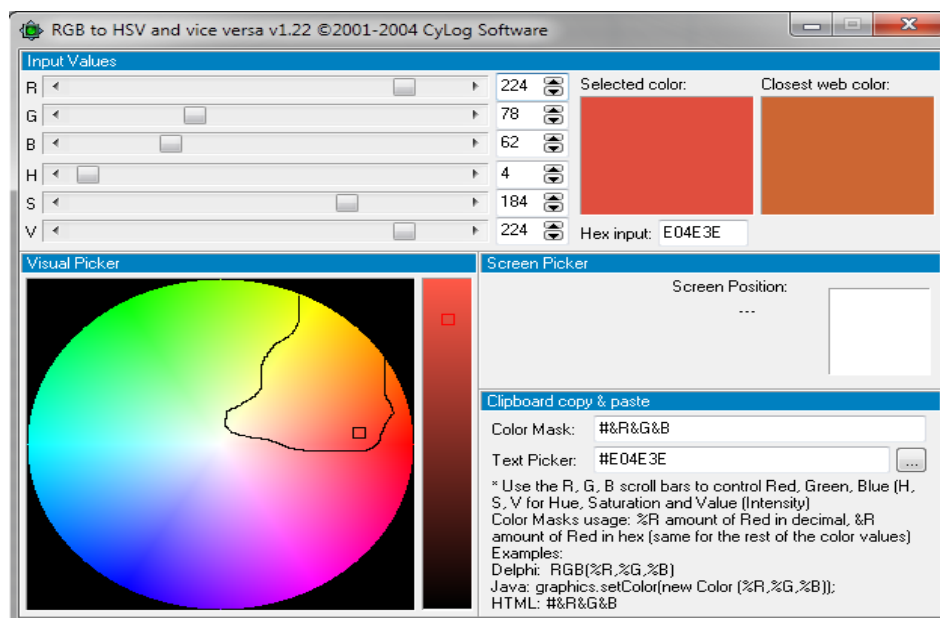


Slika 4 : Područje na kojemu tražimo RGB vrijednosti žute boje označeno je crvenim pravokutnikom

Kad su sva polja bila uspješno popunjena, mogli smo izračunati aritmetičku sredinu, varijancu i naposljetku standardnu devijaciju. Standardna devijacija je statistički pojam koji označava mjeru raspršenosti podataka u zadanom skupu. Interpretira se kao prosječno odstupanje od prosjeka, i to u apsolutnom iznosu. Kako bi odredili intervale gdje je najveća nakupina žute boje, iskoristili smo standardnu devijaciju, i izračunali odstupanje od maksimalne vrijednosti u histogramu. Histogramom prikazujemo frekvencije pojedinih vrijednosti, pa tako njegova maksimalna vrijednost predstavlja najčešću vrijednost određene komponente. Oduzeli smo od maksimalne vrijednosti standardnu devijaciju kako bi dobili minimum, te dodali na maksimalnu vrijednost standardnu devijaciju kako bi dobili maksimum intervala. Početna vrijednost našeg praga za žutu boju je sada minimum kojeg smo ranije izračunali, a konačna vrijednost praga je izračunati maksimum. Ukoliko su minimum i maksimum jednaki, nema intervala vrijednosti za dotičnu komponentu. U tom je slučaju žuta boja koju pokriva interval te komponente fiksna vrijednost same komponente.

Postupak je ponovljen za svaku komponentu, R, G i B komponentu RGB sustava boja. Na ovaj smo način dobili statistiku vrijednosti svake komponente, njihove maksimalne i minimalne vrijednosti. Te su vrijednosti onda dalje korištene kao pragovi, odnosno intervale za žutu boju u RGB sustavu boja za ulaznu sliku. Postupak je ponovljen **za svaku ulaznu sliku**, i sve je to zapisano u jednu tekstualnu datoteku `pragBGR.txt`. Prije samog početka rada, napravljen je popis koordinata prometnih znakova za svaku ulaznu sliku. Taj je popis pospremljen u datoteku sa nazivom `TableD.txt`. Osim nove ulazne slike, trebalo je promijeniti i vrijednosti koordinata kako bi dobili piksele samo dijela gdje je znak. Te su vrijednosti izvučene iz dokumenta gdje je bio popis svih koordinata. Podaci o prometnom znaku koji nam trebaju su njegove koordinate, dakle x i y komponenta na kojoj se znak nalazi, te njegova visina i širina. Na kraju je pomoću skupa intervala iz tekstualne datoteke donesen zaključak o mogućem globalnom intervalu žute

boje. Taj globalni interval bi vrijedio za svaku sliku koja se preda sustavu kao ulazna slika iz prethodno stvorene baze slika.



Slika 5 : Globalni interval označen u RGB to HSV and vice versa v1.22 programu

Kako bi bili sigurni u intervale R, G i B komponente, napravljena je i provjera granica pragova za žutu boju sa već gotovim programom [10], odnosno srednjih vrijednosti pragova. Zapravo smo provjerili da li za svaku sliku maksimalne vrijednosti R, G i B komponente u histogramu odgovaraju pikselu žute boje. Radi se o „RGB to HSV and vice versa“ programu, koji za unesene vrijednosti R, G i B komponenti daje mogućnost da i vidimo boju određenu tim vrijednostima. Upravo na taj način smo na slici označili određeno područje koje pokrivaju intervale žute boje svake slike, a naposljetku i globalni interval. Program ima i mogućnost Screen Pickera, što znači da u bilo kojem trenutku možemo saznati vrijednost bilo koje točke sa ekrana. Isto tako, možemo vidjeti i vrijednost istog piksela ne samo u RGB sustavu, već i u drugom sustavu boja, preciznije HSV sustavu boja.

3.1.1 Usporedba sustava boja RGB - HSV

Nakon što je izveden postupak za dobivanje intervala žute boje, istu smo stvar željeli učiniti i za HSV sustav boja. To je sustav boja koji čine tri glavne komponente: H za boju (engl. hue), S za zasićenje (engl. saturation), i V za vrijednost boje (engl. value). Jedan od velikih nedostataka RGB sustava je nemogućnost izravnog izdvajanja informacije o boji od informacije o osvjetljenju. Najčešće za S komponentu navedenog sustava boja možemo u literaturi pronaći i naziv osvjetljenje (engl. brightness). Zbog stalne promjene uvjeta i osvjetljenja tijekom vožnje, ulazne slike su različite kvalitete, i žuta je boja negdje bolje, a negdje lošije izražena. Za razliku od RGB sustava, u HSV sustavu boja to je puno jednostavnije, pretpostavka je da će biti možda lakše izvući intervale žute boje. Kako ova činjenica ne bi bila samo subjektivna procjena, obavljena je pretvorba ulazne slike iz RGB sustava u HSV sustav boja, i **ponovljen je identičan postupak**. Izračunali smo vrijednosti piksela samo na području prometnog znaka koji se nalazi na ulaznoj slici. Prebrojano je koliko ima piksela svake od vrijednosti H, S, i V komponente. To je sve pospremljeno opet u tri različita polja, nakon čega smo izračunali standardnu devijaciju niza brojeva. Osim standardne devijacije, trebala nam je i maksimalna vrijednost H, S, i V komponente u histogramu, za svaku ulaznu sliku. Kako bi dobili intervale žute boje, računali smo ponovno donju i gornju granicu intervala na jednak način kao za RGB sustav boja. Razlika maksimalne vrijednosti u histogramu komponente i standardne devijacije dala nam je minimum intervala, a njihov zbroj maksimum intervala. Dobili smo pragove žute boje i na slikama u HSV sustavu, te je donesen zaključak o globalnom intervalu. Sada kad smo dobili interval žute boje i u HSV sustavu, napravljena je usporedba vrijednosti.

Trebamo odrediti interval koji će vrijediti za svih 245 slika, a opet, da taj interval ne bude prevelik, odnosno da njime zaista pokupimo većim dijelom samo žutu boju sa slike. Želimo da vrijednosti unutar intervala odgovaraju najviše onima za žutu boju.



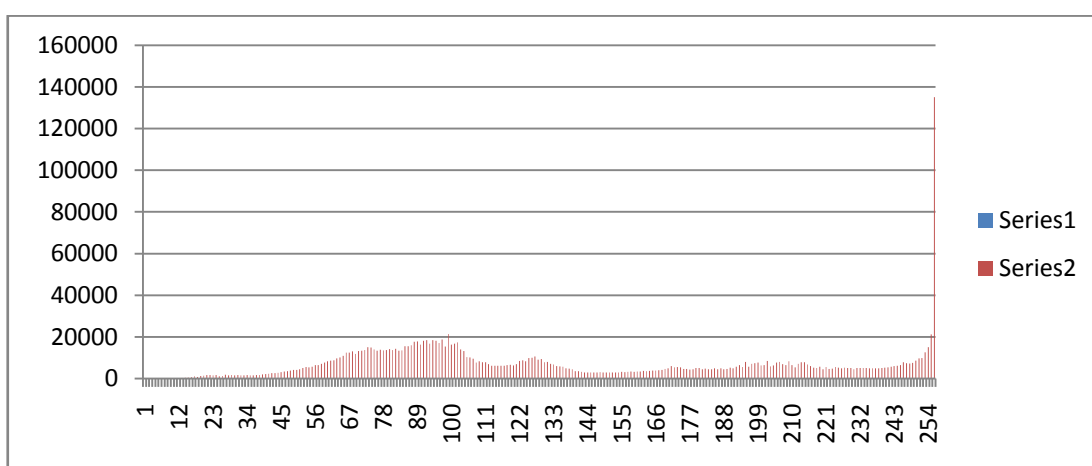
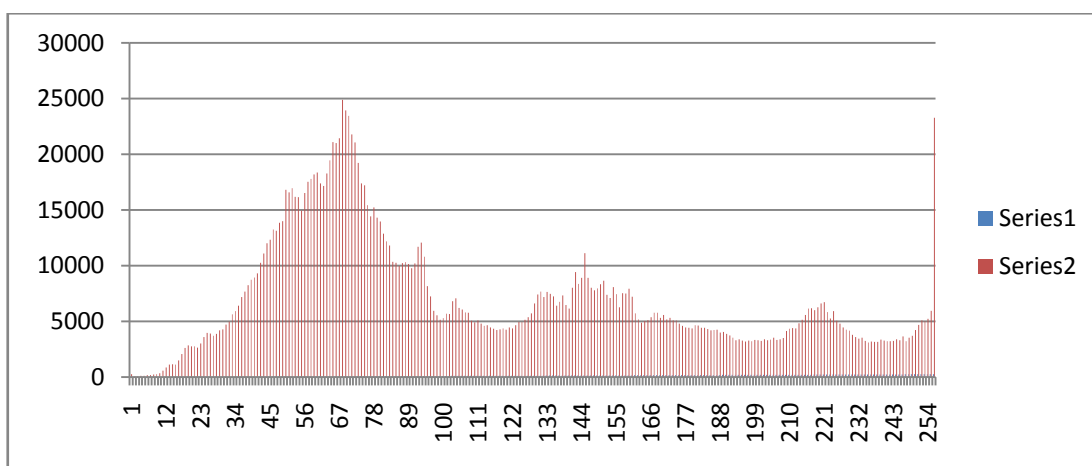
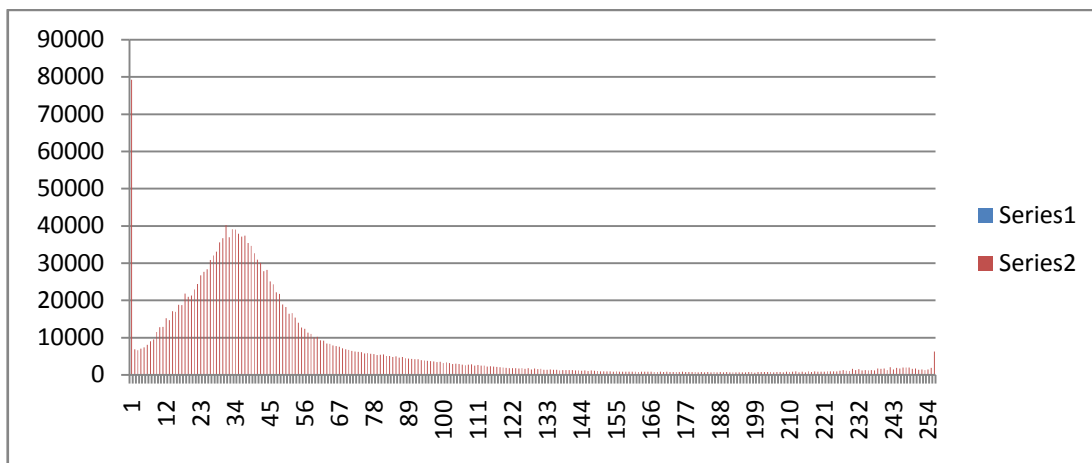
Slika 6 : Usporedba slike u RGB i HSV sustavu boja

Dosta teško je odrediti globalni interval iz 245 intervala u RGB sustavu boja, te drugi globalni interval za još 245 intervala u HSV sustavu boja, upravo zbog toga što bi to bila nagađanja i subjektivna procjena. S obzirom na to, odabir sustava boja smo odlučili pokušati dobiti na drugi način. Zaključak usporedbe dvaju sustava boja je bila pretpostavka da bi nam prilikom detekcije prometnih znakova na slici više odgovarao HSV sustav boja. To je bilo iz razloga što je prag koji određuje interval žute boje puno uži, nego onaj u RGB sustavu boja. Tako smo recimo za sliku D09_0014.bmp (slika 6) dobili interval žute boje u BGR sustavu (0-121, 198-255, 204-255), dok je prag žute boje za istu sliku u HSV sustavu boja (3-48, 180-255, 212-255). Kao što se i na HSV slici može vidjeti, žuta boja je prilično jaka, odnosno svijetli. Ako se to pogleda i sa „RGB to HSV and vice versa“ programom, koji nudi mogućnost prikaza vrijednosti komponenata boja RGB sustava, kao i HSV sustava boja, bilo koje točke sa ekrana, možemo vidjeti da su RGB vrijednosti puno veće. Tako da su i razlike vrijednosti komponenata od znaka do znaka jako velike. Pragovi komponenata G i S su slični, kao i pragovi komponenata R i V. Velike razlike uočavamo kad promatramo B i H komponentu, odnosno njihove pragove za žutu boju. Nama treba što uži interval, jer ćemo tako i puno vjerojatnije pokupiti samo žutu boju sa slike. Stoga smo za daljnji proces implementacije sustava odlučili raditi sa slikama u HSV sustavu boja, kao i sa vrijednostima piksela u HSV sustavu.

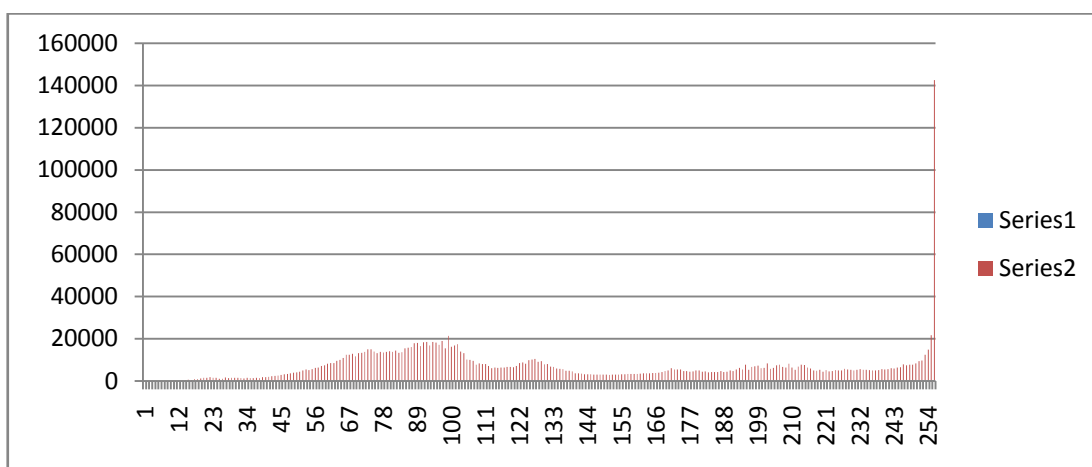
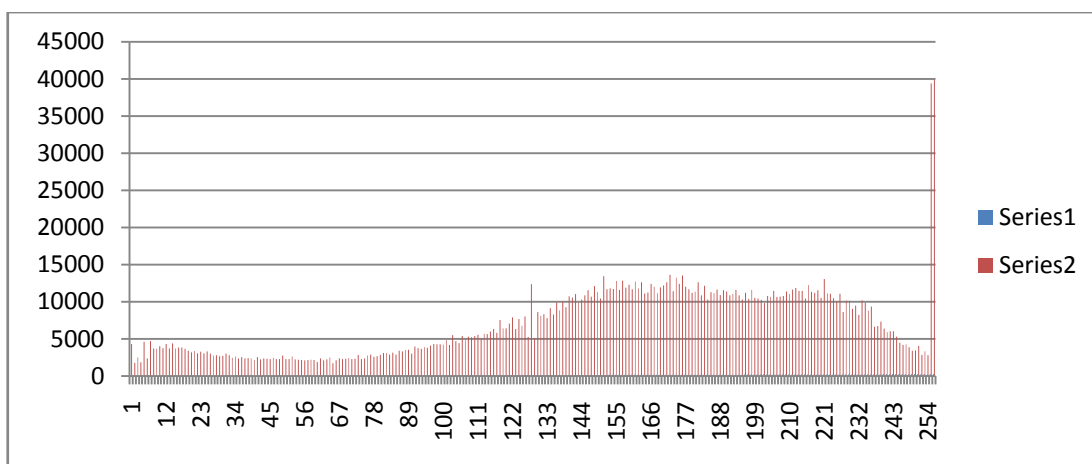
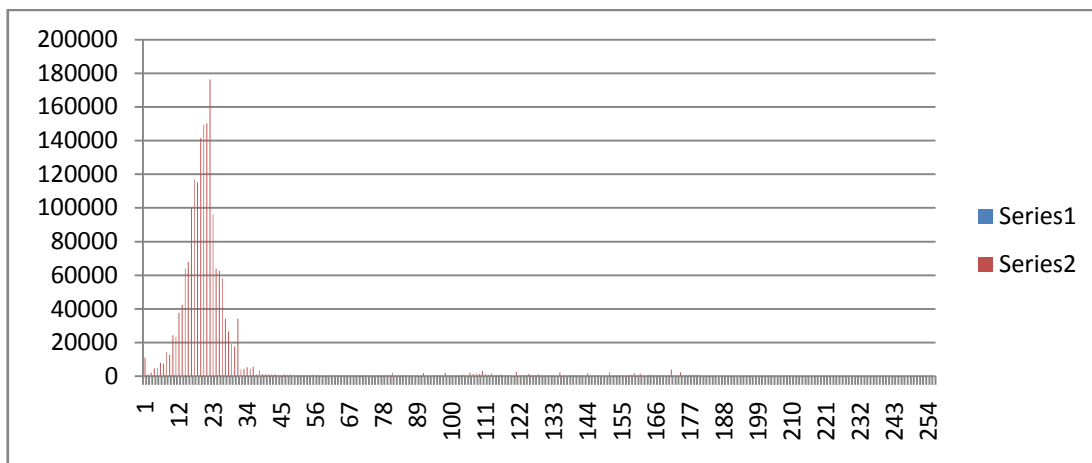
3.1.2 Uvođenje histograma

Kako uvođenje onih brojača vrijednosti komponenata RGB, ali i HSV sustava boja ne bi bilo bez razloga, ta smo polja vrijednosti iskoristili na dobar način. Osim pronalaženja pragova na statistički način, korištenjem značenja pojma standardne devijacije, odlučili smo upotrijebiti i ova polja vrijednosti. Zato smo grafički prikazali kako zapravo izgledaju vrijednosti žute boje na histogramu. Općenito, histogram je definiran kao način prikazivanja podataka raspoređenih u određene kategorije. Prvi korak u kreiranju histograma je skupljanje, i razvrstavanje prikupljenih podataka u kategorije. Nadalje moramo odrediti koje su varijable zavisne, a koje nezavisne. Karakteristika po kojoj smo grupirali podatke u kategorije predstavlja nezavisnu varijablu, a broj prikupljenih podataka koji spadaju u određenu kategoriju predstavlja zavisnu varijablu. Histogram je zapravo stupčasti graf, koji na osi apscisa ima vrijednosti nezavisne varijable, a na osi ordinata vrijednosti zavisne varijable. Oznake na osima trebale bi biti linearno raspoređene. Graf se crta tako da se prvo na os apscisa nanese vrijednosti svih kategorija, čime dobivamo os apscisa podijeljenu na intervale. Zatim se broj podataka koji odgovaraju toj kategoriji crta kao horizontalna linija iznad odgovarajućeg intervala. Upravo to je razlog zbog kojeg dobivamo stupčasti graf.

PoljeB, poljeG, i poljeR, odnosno poljeH, poljeS, i poljeV su polja od 256 elemenata, a ujedno i brojači piksela svake vrijednosti komponenata određenog sustava boja. Sve smo te vrijednosti pohranili u isto toliko tekstualnih datoteka, u kojima se onda nalazilo 245 puta po 256 vrijednosti. Broj ulaznih slika je 245, a svaka komponenta ima vrijednosti od 0-255. Na taj smo način mogli vidjeti raspon vrijednosti svake komponente. Sve smo to na kraju odlučili uz pomoć programa Microsoft Excel grafički prikazati, koji na temelju unesenih podataka lako iz tablica može stvarati grafikone. To smo i napravili sa podacima svake komponente, te time dobili njihove histograme.



Slika 7 : Histogram B, G i R komponente



Slika 8 : Histogram H, S i V komponente

Ono što smo zaključili je da se najveća raspršenost piksela toliko ni ne razlikuje od RGB do HSV sustava boja, te da se najveći broj piksela u jednom i drugom sustavu nalazi između sličnih vrijednosti komponenata. Subjektivnom procjenom smo došli do sljedećih intervala za svaku komponentu RGB i HSV sustava boja:

B: (0-71)	H: (9-30)
G: (8-118, 119-255)	S: (0-95, 96-246)
R: (52-138, 139-255)	V: (52-113, 114-255)

Analizirajući podatke dobivene histogramima, možemo reći kako nam R, odnosno V komponenta ne utječu na odabir intervala za žutu boju. Histogramima smo prikazali frekvencije pojedinih vrijednosti piksela sa slike. S obzirom da su količine piksela gotovo svake vrijednosti (0-255) komponente R relativno slične, zanimljivije su nam druge dvije komponente, B i G, odnosno H i S. G komponenta je u donjoj polovici znatno izražena, dok je to slučaj kod S komponente gornja polovica intervala. Sa B i H komponentama dolazimo do podjednakog zaključka. H komponenta ipak ima uži interval gdje se nalazi većinski broj piksela, nego što je to kod B. Nama i odgovara što kraći interval, iz razloga što bi on preciznije određivao samo jednu boju, žutu. To nam puno znači pri odabiru sustava boja za daljnji postupak. Stoga smo se odlučili za vrijednosti HSV sustava, i interval H: (9-30), i S: (119, 255). Upravo te vrijednosti nam određuju globalni interval žute boje, koji onda vrijedi za većinu naših ulaznih slika.

Kako je raspršenost posljednje komponente V podjednaka na cijelom svom intervalu, odlučili smo se na to da tu komponentu ne diramo, i ne određujemo interval pomoću sve tri komponente. Ova subjektivna procjena na temelju grafičkog prikaza piksela unutar određenog područja pokazala se znatno boljom od nagađanja globalnog intervala iz skupnih intervala za svaku pojedinu sliku. Stoga sa ovim intervalom za žutu boju nastavljamo svoj proces prepoznavanja žute boje prometnog znaka na ulaznoj slici.

3.1.3 Prepoznavanje žute boje na ulaznoj slici

Nakon što smo dobili interval žute boje koji vrijedi za ulazne slike, sa slike smo izdvojili samo piksele koji ulaze u raspon vrijednosti intervala.

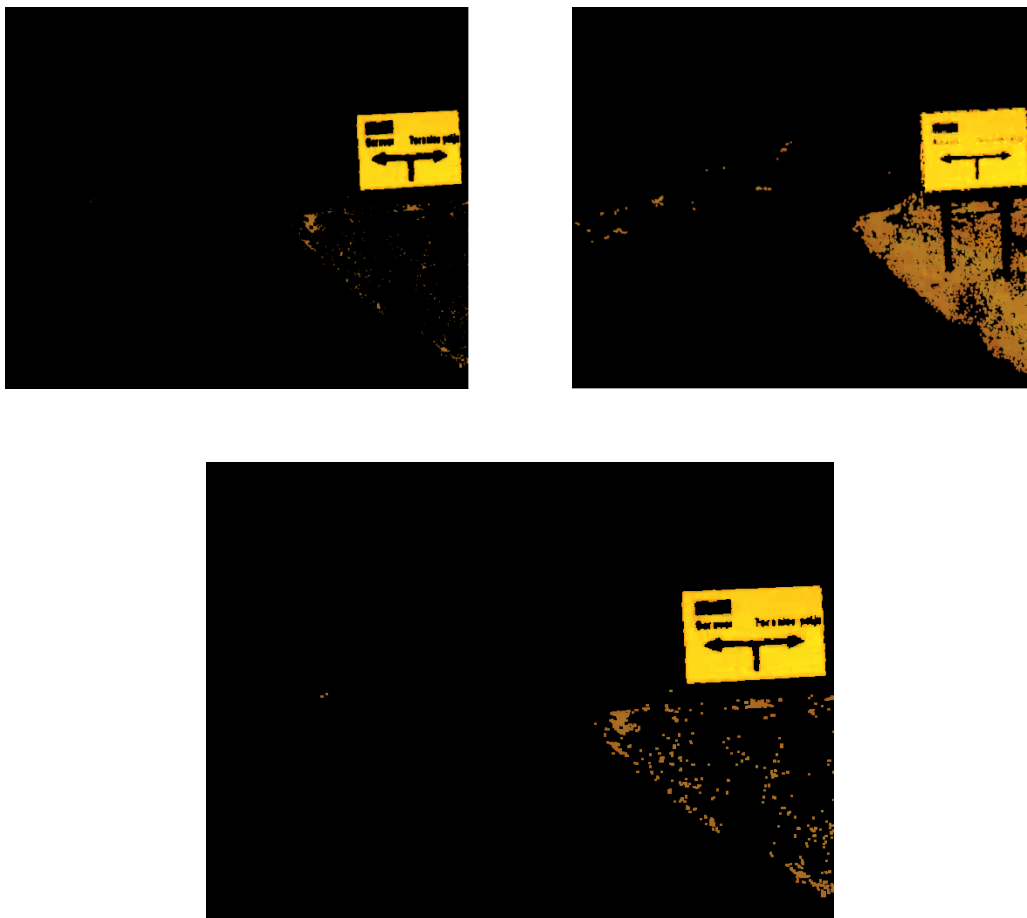


Slika 9 : Izdvajanje piksela sa slike koji odgovaraju zadanom intervalu

Sada radimo sa slikama u HSV sustavu boja, pa je izdvajanje samo određenih piksela učinjeno na način da smo za svaki piksel sa slike provjerili da li njegove vrijednosti H, S, i V ulaze u raspon vrijednosti za žutu boju. Ako je piksel unutar zadanog intervala, posprema se u polje vrijednosti. Kasnije se svi takvi pikseli jedan po jedan unose na novu sliku, i tako smo dobili izlaznu sliku samo sa pikselima žute boje. Sliku smo pretvorili iz RGB sustava u HSV sustav boja. Sa cvGet2D funkcijom dobivamo vrijednosti piksela u HSV sustavu, identično kao i u ranijem postupku. Sada znamo vrijednosti piksela koji su unutar intervala, uzimamo vrijednost piksela na tim istim koordinatama samo sada sa RGB slike, i na novu sliku unosimo takve piksele. To smo učinili sa RGB vrijednostima, jer nam nisu vizualno zanimljive vrijednosti piksela „iskrivljene“ slike, one u HSV sustavu boja.

3.1.4 Uklanjanje šuma sa slike

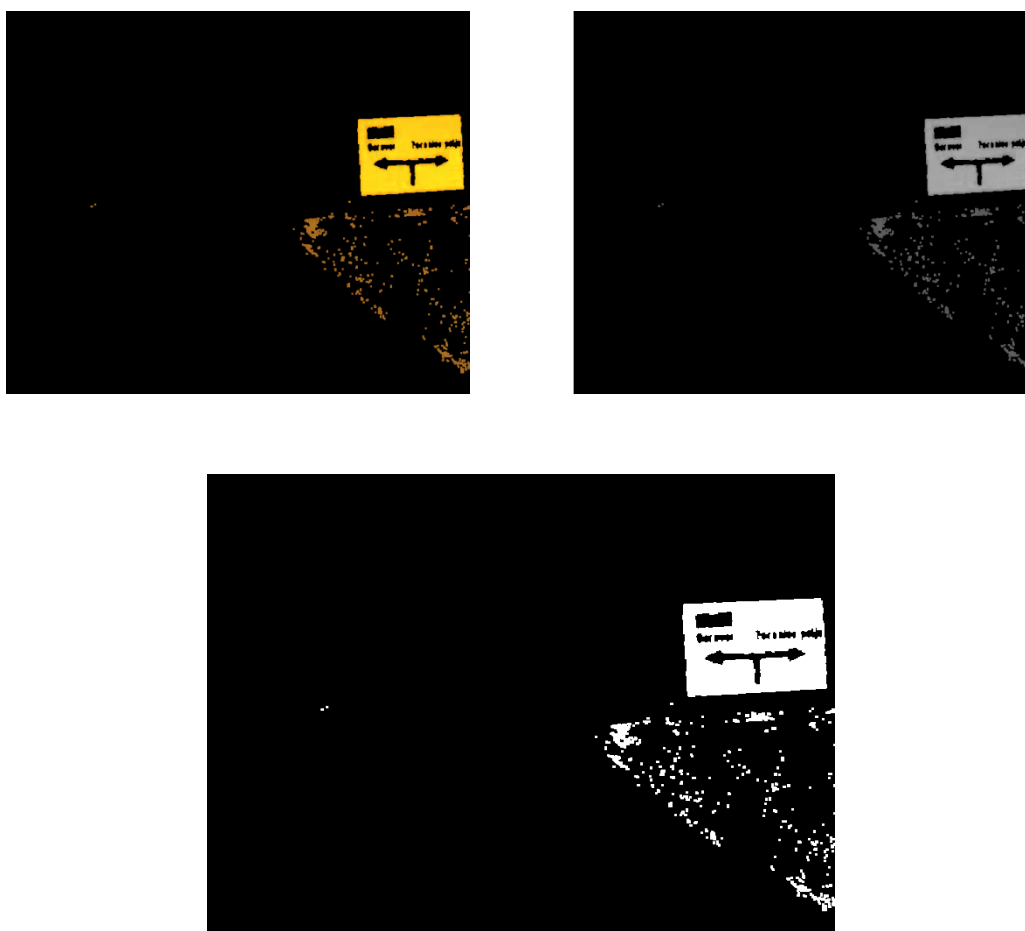
Kako bi sve te piksele traženih vrijednosti pokupili u što veće nakupine, napravili smo postupak erozije, a potom proveli i dilataciju nad slikom. Željeli smo što „čišći“ znak, pa je stoga bilo potrebno uklanjanje šuma, odnosno mrlja žute boje na slici. To je napravljeno sa OpenCV funkcijama cvErode, i cvDilate. I jedna i druga funkcija se mogu iskoristiti od jednom do nekoliko puta, to su iteracijski postupci. cvErode radi eroziju nad slikom, dok cvDilate radi dilataciju, prva funkcija kao da „nagrizi“ sliku, dok druga popravlja sliku, i ispunja lijepo prazne piksele oko nakupine. Erozijom smo maknuli sve one nakupine piksela žute boje koje smo dobili raspršene po slici, dok smo sa dilatacijom lijepo zagladili veće nakupine piksela. Dva puta smo pokrenuli funkciju erozije, a nakon nje i tri puta funkciju dilatacije.



Slika 10 : Erozija, dilatacija, i erozija sa dilatacijom

Sa slike se selektiraju svi pikseli koji su žute boje, no ono što je nama potrebno, potrebna je samo nakupina žute boje gdje se nalazi prometni znak. Zato je potrebno ukloniti šum na slici, i dobiti samo „čisti“ prometni znak. Algoritam je osmišljen tako da traži najveću nakupinu žute boje. Pretpostavka je da je to naša žuta pravokutna ploča, koja označava prometni znak.

Prije nego započnemo prepoznavanje oblika određene nakupine, odlučili smo sliku prebaciti ponovno u drugi sustav boja. Ona tako više nije u HSV sustavu, već smo od trokanalne slike napravili jednokanalnu. Rezultat tog postupka je crna-bijela slika. Cijela slika je zapravo crne boje, a oni dijelovi koji su prije bili žuti, sada su bijeli. Sve to je učinjeno kako bi prepoznavanje po obliku bilo što efikasnije, upravo zbog funkcija koje koristimo.



Slika 11 : Proces dobivanja binarne, crno-bijele slike

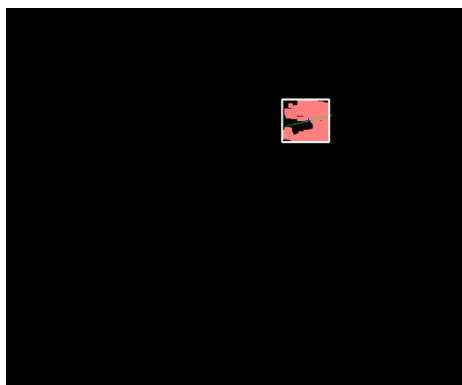
3.2 Prepoznavanje pravokutnog oblika prometnog znaka

Nakon što je obavljen cijeli proces prepoznavanja žute boje na slici kako bi se suzilo područje potrage za prometnim znakom, slijedi postupak prepoznavanja oblika. Pod pretpostavkom da je pronađena nakupina upravo nakupina žute boje, i da označava traženi prometni znak (znak D skupine), slijedi postupak prepoznavanja oblika izdvojene nakupine. Ono što tražimo je prometni znak žute boje pravokutnog oblika. To je učinjeno na način da smo sa određenim OpenCV funkcijama koje rade sa blobovima, odnosno nakupinama detektirali nakupine na slici. S obzirom da funkcije koje rade sa blobovima rade samo na jednokanalnim slikama, bilo je potrebno sliku najprije prebaciti u jednokanalnu. To smo i napravili. Slika je najprije pretvorena u sivu (grayscale) sliku, a onda smo nad tom sivom slikom proveli postupak `cvThreshold`. Razlog prebacivanja sustava boja iz HSV u crno-bijelu sliku je taj što funkcija `cvThreshold` radi na jednokanalnim slikama. Ona se tipično koristi za dobivanje binarne slike iz sive slike, ali i za uklanjanje šuma, kao što je to filtriranje piksela koji imaju prevelike ili premale vrijednosti. Funkcija `cvThreshold` nam je obojila sve tamnije piksele u crnu boju, a sve svjetlije u bijelu, i dobili smo crno-bijelu sliku, gdje su bijela područja označavali pikseli koji su ranije bili žuti.



Slika 12 : Izdvajanje nakupina žute boje na slici

Kad je pronađena najveća nakupina žute boje, analizira se površina tog područja, i uspoređuje sa površinom opisanog pravokutnika oko tog područja. Ukoliko je površina te pravokutne oznake područja znaka dovoljno slična, odnosno ispunjena većim dijelom žute boje, kažemo da je nakupina pravokutnog oblika. Time je postupak za implementaciju sustava za detekciju prometnog znaka koji je žute boje, i pravokutnog oblika završen.



Slika 13 : Pronalazak najveće nakupine žute boje na slici

Cijeli postupak pretvaranja sustava boja slika na kraju izgleda na sljedeći način : BGR - HSV, BGR - GRAYSCALE slika - BW slika, i na kraju ponovno vraćanje u početni oblik, u RGB sustav boja. Na toj originalnoj polaznoj slici smo onda označili crvenom bojom mjesto gdje je pronađen prometni znak sa OpenCV funkcijom crtanja pravokutnika cvRectangle.



Slika 14 : Rezultat postupka za detekciju prometnih znakova

4. Rezultati i uspješnost dobivenog sustava

4.1 Testiranje

Uspješnost implementiranog sustava za detekciju prometnih znakova ispitana je na unaprijed pripremljenoj bazi ulaznih slika iz stvarnoga svijeta. Ona se sastoji se od 245 slika u .bmp formatu, u boji, i rezolucije 720x576 piksela. Ulazne slike su fotografije izrezane iz video materijala onih kadrova gdje se pojavljuju prometni znakovi koji su nam potrebni. Snimke su napravljene iz vozila u pokretu na području Zagreba i okolice, te na području slavonskih cesta, u različitim vremenskim uvjetima, i u različita doba dana.

Rezultati uspješne detekcije prometnih znakova daju sljedeći postotak ispravnosti sustava:

$$230/243 * 100\% = 94,65\%$$

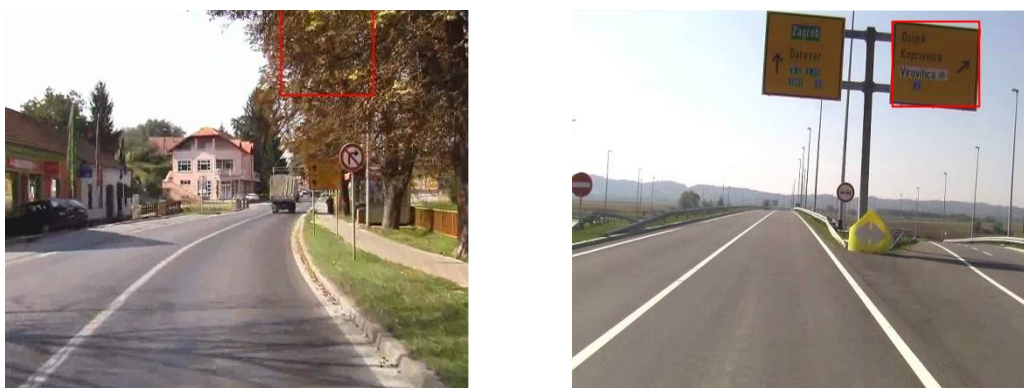


Slika 15 : Primjer uspješnih detekcija prometnih znakova na slici

Pogreška implementiranog sustava:

$$100 - \text{uspješnost} = 13/243 * 100\% = 5,35\%$$

Na ispitanih 245 ulaznih slika iz baze, bilo je 243 prometnih znakova, od kojih je uspješno pronađeno 234 znaka, dok ih 13 algoritam nije uspio pronaći. Bilo je 8 neuspješnih detekcija, gdje je sustav detektirao samo jedan prometni znak na slici gdje su dva prometna znaka. A i neuspješno su detektirani znakovi koji su jako tamni, gdje je sustav detektirao samo dio znaka kao prometni znak. Osim 8 neuspješnih, imamo i 5 lažnih detekcija, što znači da je algoritam označio područje na kojem nema prometnog znaka, kao da tamo jest prometni znak. U slučajevima gdje su žuti prometni znakovi premali da bi ih sustav detektirao kao takve, sustav nije detektirao ništa. Algoritam je napravljen tako da prilikom analize nakupina boja zanemari područja koja su površine manje od 200 piksela.



Slika 16 : Primjer neuspješnih detekcija prometnih znakova na slici

S obzirom na različitu kvalitetu ulaznih slika, globalni interval žute boje ipak nije pokrio žutu boju na svim slikama. A imamo i slučaj djelomičnih detekcija. Slike na kojima se pojavljuje više prometnih znakova, a ne samo jedan, daju djelomičnu uspješnost detekcija. Naime, jedan znak uvijek prepozna, dok će ostale zanemariti, što svaki puta daje jednu uspješnu detekciju, i jednu ili više neuspješnih detekcija prometnih znakova na ulaznoj slici.

Pomicanjem granica intervala žute boje, točnije S komponente (saturation, zasićenje boje) HSV sustava boja, mogli bismo dobiti uspješno detektirane prometne znakove na slikama gdje to prije nije bilo uspješno. Ali time ne bismo povećali postotak uspješnosti sustava za detekciju, jer ne bismo uspjeli detektirati većinu ovih znakova koji su do sada bili uspješno detektirani (slike 17 i 18).



Slika 17 : Izlazne slike prije, i nakon povećanja intervala komponente S, sa minimalne vrijednosti 119, na 159

Možemo primijetiti kako je povećavanjem donje granice praga za S komponentu sa 119 na 159 detektiran prometni znak na slikama gdje smo prije imali neuspješne detekcije, i gdje je detektirana ograda umjesto znaka. To je zato što je ograda nešto tamnijih vrijednosti od prometnog znaka, no isti prag nije pomogao kako bi se uspješno detektirao znak i na svim ulaznim ostalim slikama.



Slika 18 : Izlazne slike prije, i nakon smanjivanja intervala komponente S, sa minimalne vrijednosti 119, na 99

4.2 Analiza dobivenih rezultata

4.2.1 Stapanje okoline sa prometnim znakom

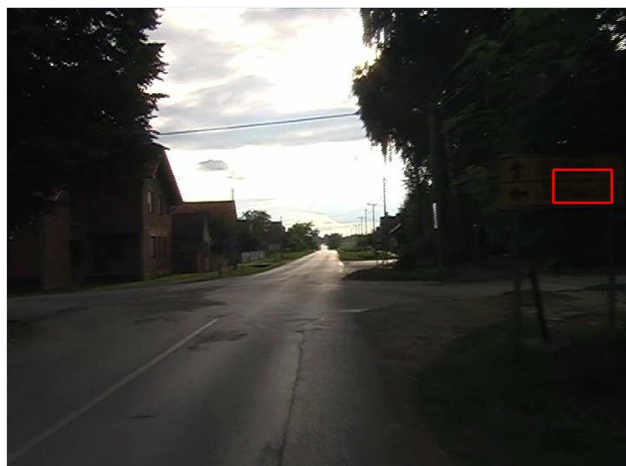
Zadnji je korak statistika podataka dobivenih implementiranim sustavom, da li je prometni znak na ulaznoj slici uspješno detektiran, ili nije. Ono što najmanje pogoduje ispravnim rezultatima sustava je stapanje boje okoline sa bojom prometnog znaka na slici. Tako je ovdje naveden primjer gdje su žute boje : ograda, znak uz ogradu C skupine prometnih znakova, prometni znak D skupine (koji ovaj sustav zapravo mora pronaći na danoj slici), te lišće na stablu u pozadini. Sva četiri područja su nakupine žute boje, i to vrlo sličnih vrijednosti. Stoga sustav može pogrešno detektirati traženi prometni znak, ili čak negdje i veće područje oko samoga znaka. Do detekcija većeg dijela slike može doći kad je okolina prometnog znaka slične boje, pa sustav osim znaka detektira i njegovu pozadinu.



Slika 19 : Stapanje okoline sa prometnim znakom

4.2.2 Loša kvaliteta prometnog znaka na ulaznoj slici

Još neki od nedostataka implementiranog sustava su loši vremenski uvjeti, sjena, ili pak snimanje u sumrak, te loš kut snimanja. To su sve uvjeti koji znatno utječu na boju prometnog znaka na slici, jer ćemo tako snimajući najčešće dobiti slike koje su znatno lošije kvalitete. U takvim lošim uvjetima i prometni znak postaje lošije kvalitete, pa često ne ispadne žute boje, već tamno sive, smeđe, zelenkaste, ili čak ponegdje i crne boje. Ovdje je dan primjer jedne takve ulazne slike, gdje bi znak trebao biti žute boje, ali nije. Možemo primijetiti kako boja prometnog znaka i jedan veći dio slike imaju jako slične vrijednosti. Osim što je boja znaka neprepoznatljivo žuta, toliko je tamna da se i vrlo dobro stopila sa okolinom znaka. Kad ne bi znali da je to zapravo znak žute boje, ne bismo znali da na tom mjestu sustav treba pokazati da je detektirao prometni znak. Vrijednosti komponente koja daje žutu boju su ipak još dovoljno blizu žutim nijansama da sustav jedan dio tog znaka može uspješno detektirati.



Slika 20 : Loša kvaliteta ulazne slike

4.2.3 Prepoznavanje više od jednog znaka na slici

Ono što je problem izvedenog sustava je pronalaženje više od jednog znaka na ulaznoj slici. Na slici mogu biti dva, ili čak više prometnih znakova D skupine koje sustav mora detektirati. U našoj bazi testnih uzoraka, nalaze se četiri slike koje imaju i do tri prometna znaka. Sustav to neće uspješno izvesti zbog toga što traži najveću nakupinu žute boje na slici, nakon čega radi prepoznavanje oblika te nakupine, koja mora biti pravokutna. Ako su dva znaka približno jednake veličine, ili je jedan od drugoga puno manji, sustav će detektirati samo jedan znak kao traženi prometni znak, onaj najveći. Kad bismo sustav drugačije napravili, pogreška bi bila znatno veća s obzirom na razne dijelove slike, koji također mogu biti žute boje. Tako je to bio primjer sa žutom ogradom uz cestu, ili sa lišćem na drveću, koje ima poprilično slične vrijednosti boje kao prometni znak. U slučaju drugačije implementacije, gdje bi ostale nakupine tražili po sličnosti površine najveće nakupine, ili relativnoj blizini najvećoj nakupini, dobili bismo puno lošije rezultate. Takvi rezultati su za očekivati, upravo iz razloga što bi sustav onda detektirao više nakupina žute boje, umjesto samo jedne, tamo gdje to ne bi trebao.



Slika 21 : Neprepoznavanje više od jednog znaka na slici

5. Upute za korištenje ostvarenog sustava

Za ispravno korištenje programski ostvarenog sustava, potrebno je prilikom poziva programa predati programu argument u obliku puta do ulazne slike. To je prvi argument, i označava ulaznu sliku nad kojom se provodi sustav za detekciju prometnih znakova. Primjerice, ukoliko na ulaz programski ostvarenog sustava pošaljemo ulaznu sliku u BMP formatu (koja ujedno mora biti slika u boji), sa imenom D03_0006, koja se nalazi na D disku u direktoriju PrometniZnakovi, i poddirektoriju TableD, tada će prvi predani argument biti: D:\PrometniZnakovi\TableD\D03_0006.bmp. Pokretanjem programski ostvarenog sustava kao rezultat procesa detekcije dobit ćemo prozor „Detekcija prometnih znakova na temelju boje i oblika“. Taj prozor nam predstavlja rezultat detekcije našeg sustava, gdje je ista ona originalna ulazna slika. Osim ulazne slike, u prozoru možemo vidjeti i da je crvenim pravokutnikom označeno područje oko pronađenog znaka. Za kraj rada sustava potrebno je pritisnuti bilo koju tipku na tipkovnici.



Slika 22 : Izlaz programski ostvarenog sustava

6. Zaključak

Ovaj sustav za detekciju prometnog znaka na slici ostvaren je na temelju boje i oblika dotičnog znaka. Predajemo mu ulaznu sliku, algoritam obrađuje sliku, i pokaže nam područje gdje je detektirao znak. Sustav to radi na temelju prepoznavanja žute boje, a potom i pravokutnog oblika prometnog znaka. Kako bi sustav imao što bolju daljnju primjenu, trebalo bi poboljšati sustav tako da se može koristiti za sve znakove. Potrebno je povećati granice dosega algoritma, kako bi njegova primjena bila šira od samo određene skupine prometnih znakova. Ono što najmanje pridonosi uspješnosti detekcije je stapanje znaka sa njegovom okolinom. Osim toga, tu su i različiti vremenski uvjeti u kojima je napravljena baza ulaznih slika za testiranje sustava. S obzirom da u različitim uvjetima možemo dobiti različite boje znakova koji su zapravo iste boje, puno toga ovisi i o tome kako radimo ulazne slike. Sve su to zapravo sitnice koje ne bi trebale znatno utjecati na rezultate same detekcije, ali nažalost utječu. Kako bi poboljšali sustav, mogli bi ga implementirati zajedno uz metode za prepoznavanje znaka na slici. Ukupni rezultat uspješnosti od preko 94% daje veliko povjerenje u sustav, no to je samo jedan eksperimentalan broj. Taj nam broj upravo daje mogućnost, i svejedno puno mjesta za daljnje poboljšavanje izvedbe samoga sustava. Slike lošije kvalitete bi tako mogli obraditi, izjednačiti im histograme kako bi dobili ljepšu vidljivost boje znaka. Isto tako, sustav bi se mogao izvesti strojnim učenjem. Metoda prikupljanja što više ulaznih slika kojima bi prikazali što više prometnih znakova, učenje sustava o karakteristikama znakova, dali bi neke nove poglede na detekciju. Od sustava koji bi podržavao detekciju i prepoznavanje prometnog znaka, vozači bi mogli imati puno koristi. Kao takav bi ubrzao promet, i omogućio lakše snalaženje, a time i poboljšao kvalitetu vožnje.

7. Literatura

- [1] Prometna-zona.com: Znakovi obavijesti za vođenje prometa
http://www.prometna-zona.com/prometni_znakovi-obavijesti_za_vodjenje_prometa.html, listopad 2009.
- [2] AutoŠkola Formula: Znakovi za vođenje prometa
<http://autoskola-formula.hr/vodjenje-prometa>, listopad 2009.
- [3] OpenCV
<http://en.wikipedia.org/wiki/OpenCV>, listopad 2009.
- [4] Prometni znakovi
http://hr.wikipedia.org/wiki/Prometni_znakovi, listopad 2009.
- [5] Ministarstvo mora, turizma, prometa i razvitka - Pravilnik o prometnim znakovima, signalizaciji i opremi na cestama : Znakovi obavijesti za vođenje prometa
<http://narodne-novine.nn.hr/clanci/sluzbeni/dodatni/314288.htm>, listopad 2009.
- [6] Open Computer Vision Library
<http://sourceforge.net/projects/opencvlibrary/files/>, listopad 2009.
- [7] I hate (love) OpenCV - Getting started with OpenCV in Visual Studio 2005
<http://opencv.blogspot.com/2006/04/getting-started-with-opencv-in-visual.html>, listopad 2009.
- [8] Introduction to programming with OpenCV
<http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/index.html>, listopad 2010.

- [9] Fofonjka M. : Detekcija prometnih znakova na temelju modela oblika, završni rad, 2009.
- [10] CyLog Software: RGB to HSV and vice versa v1.22, 2001-2004
<http://www.cylog.org/graphics/rgb2hsv.jsp>, studeni 2010.
- [11] Bradski G., Kaehler A. Learning OpenCV Computer Vision with the OpenCV Library. SAD, O'Reilly Media, Inc., Sebastopol, 2008
- [12] Image Processing and Analysis Reference
http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef_ImageProcessing.htm, prosinac 2010.
- [13] Elezović M., Samaržija B., Senji I., Suhina V. : Detekcija rubova s primjenom u lokalizaciji objekata, seminar, 2005.
- [14] Rojković D. : Raspoznavanje prometnih znakova križnom korelacijom, završni rad, 2009.
- [15] Krnjić F. : Raspoznavanje prometnih znakova neuronskim mrežama, završni rad, 2009.
- [16] Šverko M. : Pronalaženje prometnih znakova Houghovom transformacijom za kružnice, završni rad, 2009.
- [17] Prof. dr. sc. Bojana Dalbelo-Bašić : Strojno učenje
<http://www.fer.hr/predmet/su>, siječanj 2010.

Detekcija prometnih znakova na temelju boje i oblika

Sažetak

U ovome radu smo pokazali jedan od načina detekcije prometnih znakova na slikama. Obrada se odvijala u dva koraka. Prvi je korak bilo prepoznavanje boje, a drugi prepoznavanje oblika. Time smo detektirali žute prometne znakove pravokutnog oblika na slikama snimljenima iz automobila u pokretu. Algoritam je implementiran programskim jezikom C/C++. Rezultati uspješnosti statistički su obrađeni, i prikazani na stvarnim slikama.

Ključne riječi

Prometni znakovi, prepoznavanje boje, prepoznavanje oblika

Traffic signs detection based on color and shape

Abstract

In this work we demonstrated one way of traffic signs detection on images. Processing took place in two steps. First step was identifying color, and the other one shape recognition. Thus we have detected yellow rectangular traffic signs on images taken from car in motion. The method has been implemented in programming language C/C++. Results of the efficiency were statistically processed and presented on real images.

Keywords

Traffic signs, identifying color, shape recognition