

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1928

**EKSPERIMENTALNA EVALUACIJA
USPJEŠNOSTI DETEKCIJE RAZLIČITIH
RAZREDA PROMETNIH ZNAKOVA**

Ivan Čulina

Zagreb, lipanj 2011.

SADRŽAJ

1. UVOD	1
2. METODA VIOLE I JONESA	3
2.1 HAAROVE ZNAČAJKE	4
2.2 INTEGRALNA SLIKE	5
2.3 GRADNJA KLASIFIKATORA KORIŠTENJEM ADABOOST METODE	7
2.4 ULANČAVANJE KLASIFIKATORA U KASKADU	10
3. PROGRAMSKO RJEŠENJE	10
3.1 OPENCV	12
3.2 OPIS PROGRAMA	13
4. EKSPERIMENTALNI REZULTATI	17
4.1. PARTICIONIRANJE PROMETNIH ZNAKOVA U HOMOGENE RAZREDE	17
4.2 REZULTATI EVALUACIJE.....	21
5. ZAKLJUČAK.....	26
6. LITERATURA	28

1. UVOD

Česta je uporaba računalnih programa koji automatski detektiraju, raspoznaju i prate prometne znakove. U zadnje vrijeme, sve veća. Vozači često previde prometne znakove što lako dovodi do nesreće. Ako automobil sadrži računalno koje automatski raspoznaje te znakove i upozori vozača, značajno se može povećati sigurnost u prometu. Događa se da prometni znakovi nisu u stanju ili položaju u kojem bi trebali biti. Mogu biti oštećeni, prekriveni biljem ili nestali. Službe zadužene za sigurnost u prometu moraju to popraviti. Proces pronalaska takvih znakova treba biti automatiziran. Ako bi se napravilo vozilo koje bi se trebalo samo kretati po cesti, bez pomoći čovjeka, računalno bi moralo automatski raspoznavati prometne znakove...



Slika 1.1: Detekcija

Detekcija je pronalaženje položaja prometnog znaka na slici. Raspoznavanje je određivanje vrste detektiranog prometnog znaka. Praćenje je određivanje pomaka detektiranog prometnog znaka.

Ovaj završni rad bavi se detekcijom prometnih znakova. Ta primjena nalazi se na presjeku umjetne inteligencije, raspoznavanja uzoraka i računalnog vida. Postoji više metoda koje se mogu koristiti za detekciju prometnih znakova. Prethode studije su uputile na dobre rezultate detekcije najčešćih homogenih nadrazreda prometnih znakova korištenjem metode Viole i Jonesa. Homogeni nadrazredi znakova su grupirane vrste znakova koje slično izgledaju i mogle bi se detektirati istim klasifikatorom (znakovi opasnosti, znakovi ograničenja brzine, znak stop...). Metoda Viole i Jonesa temelji se na kaskadama ojačanih (eng. *boosted*) Haarovih klasifikatora. Dobri rezultati detekcije najčešćih homogenih nadrazreda ne garantiraju dobre rezultate za ostale homogene nadrazrede. Ovaj rad ispituje rezultate korištenja iste metode i na ostalim nadrazredima. Na prošlogodišnjem projektu [2] pronašlo se 97,55% trokutastih prometnih znakova u skupu za testiranje. Koristilo se 2150 slika za učenje. Većina homogenih nadrazreda iz ovog rada ne koristi približno toliko, ali je većina znatno homogenija od nadrazreda koji sadrži trokutaste prometne znakove.

U ovom radu opisana je metoda Viole i Jonesa. Particioniran je skup svih prometnih znakova u homogene nadrazrede. Za svaki nadrazred pokrenuto je učenje klasifikatora i evaluacija. Prikazani su dobiveni rezultati.

2. METODA VIOLE I JONESA

Paul Viola i Michael J. Jones su stvorili metodu za robusno i ekstremno brzo pronalaženje objekata na slikama. Njeno vrlo bitno svojstvo je detekcija objekata u realnom vremenu. Za razliku od ostalih metoda za detekciju ova ne koristi pomoćne informacije poput boja i uspoređivanja slika koje slijede jedna drugu u videu. Koriste se samo informacije koje su dobivene iz pojedinačnih slika prikazanih u nijansama sive boje. Metoda koristi strojno učenje. Primarna svrha joj je detekcija lica, ali učinkovita je i kod detekcije ostalih objekata.

Metoda se sastoji od faze učenja i faze detekcije. U fazi učenja nad skupom za učenje se stvara klasifikator. Skup za učenje čine slike koje mogu biti pozitivni i negativni primjeri. Na pozitivnim primjerima traženi objektni se pojavljuju, dok kod negativnih to nije slučaj. Slike koje čine skup za učenje prethodno su obrađene i zna se sadrže li objekt i na kojem se položaju objekt nalazi ako ga sadrže. Primjeri moraju biti kvadratnog oblika.

Ako je napravljen klasifikator, može se raditi detekcija. U fazi detekcije koristi se pomično okno (potprozor). U oknu se nalazi dio slike nad kojom se vrši detekcija. Početni sadržaj okna nalazi se gore lijevo na slici. Klasifikator određuje sadrži li okno traženi prozor ili ne. Nakon svakog klasificiranja sadržaja okna, okno se pomiče jedan piksel u desno. Kada okno dođe do kraja desno, pomiče se jedan piksel prema dolje i skroz lijevo. Kada se okno više ne može pomaknuti ni desno ni dolje povećava se i ponovno prolazi slikom. Zadnji sadržaj okna je cijela slika.

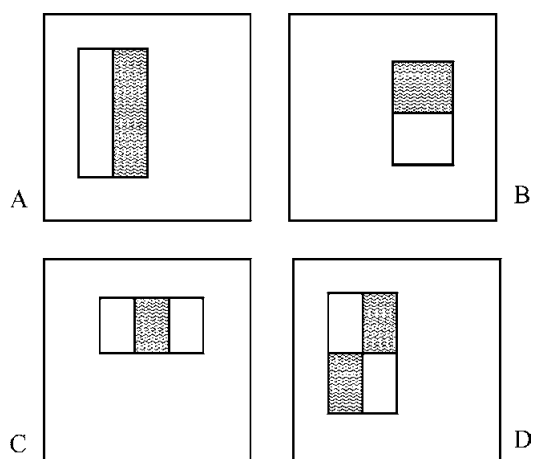
Metoda koristi kaskade ojačanih Haarovih klasifikatora. Glavna tri doprinosa metode su:

- integralna slika
- izrada klasifikatora korištenjem AdaBoost algoritma
- ulančavanje klasifikatora u kaskadu

2.1 ZNAČAJKE

Detekcija se temelji na značajkama (eng. *features*). Koriste se četiri osnovna tipa. Prikazana su na *slici 2.1.1* (Viola, 2004). Značajke A i B su Haarove značajke, a C i D su dodali Viola i Jones. Značajke su pravokutne. To ih čini grubima, ali zato se jako brzo određuju njihove vrijednosti. Vrijednost značajki se određuju kao razlika zbroja svih vrijednosti piksela unutar dvaju područja slike. Te vrijednosti govore o prisutnosti nekih karakteristika slike.

Značajke se određuju na slici koja je fiksne veličine. Obično je ta veličina 24x24 piksela. Skup značajki za sliku veličine 24x24 piksela sadrži 160 000 značajki. Traže se značajke koje najbolje odvajaju pozitivne od negativnih primjera.



Slika 2.1.1: Osnovni tipovi značajki

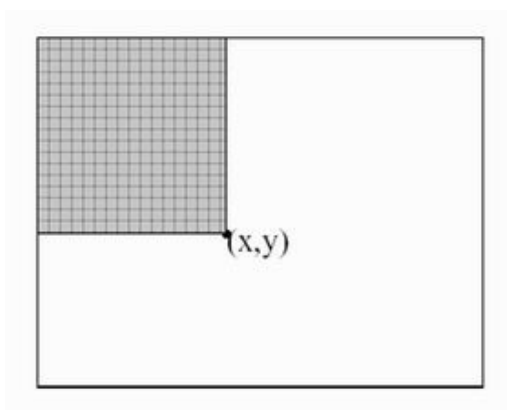


Slika 2.1.2: Pronađena značajka

2.2 INTEGRALNA SLIKA

Pri učenju klasifikatora mnogo puta se određuju vrijednosti značajki. Stoga je jako bitno ubrzati određivanje vrijednosti značajke. Najbolji rezultat dobiva se korištenjem integralnih slika. Svaki piksel u integralnoj slici ii izračunava se kao zbroj vrijednosti svih piksela originalne slike i lijevo i iznad piksela uključujući i originalnu vrijednost samog piksela.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (1)$$



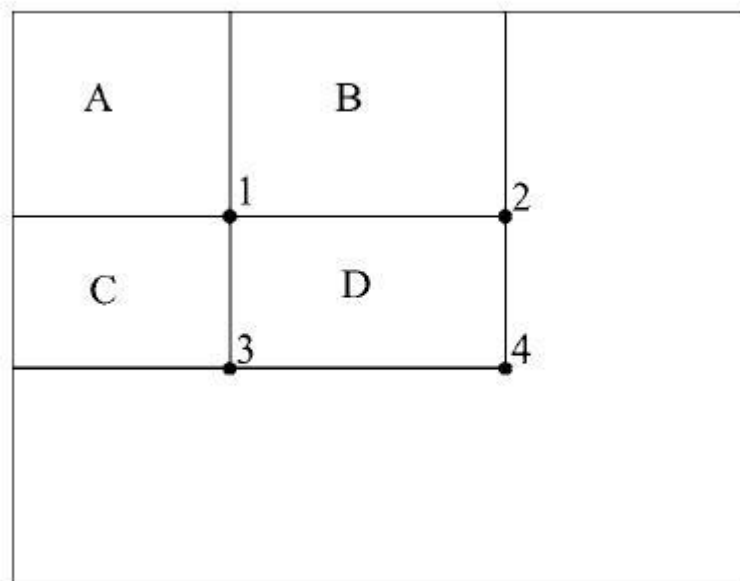
Slika 2.2.1: Vrijednost piksela u integralnoj slici

Na *slici 2.2.1* vrijednost piksela (x, y) u integralnoj slici je zbroj vrijednosti svih piksela u originalnoj slici koji se nalaze u sivom pravokutniku.

Postupak izračunavanja vrijednosti piksela integralne slike može se obaviti u samo jednom prolazu originalne slike koristeći sumu svih piksela u stupcu s iznad traženog piksela.

$$\begin{aligned}
 s(x, y) &= s(x, y - 1) + i(x, y) \\
 ii(x, y) &= ii(x - 1, y) + s(x, y)
 \end{aligned}
 \tag{2}$$

Jednom kad se izračuna vrijednost integralne slike, računanje značajki postaje ekstremno brzo. Zbroj vrijednosti piksela unutar pravokutnika se vrlo lako izračuna. Bilo koji zbroj vrijednosti piksela unutar pravokutnika može se izračunati koristeći četiri reference na polje na polje podataka. To znači da se značajke mogu izračunati koristeći samo 6, 8 ili 9 referenci na polje podataka. Na primjer, želimo izračunati sumu piksela unutar pravokutnika D na *slici* 2.2.4. Dovoljno je zbrojiti vrijednosti piksela u točki 4 i 1 te oduzeti vrijednosti u točkama 2 i 3.



Slika 2.2.2: Zbroj piksela u $D = 4 + 1 - (2 + 3)$

2.3 ADABOOST METODA

Binarni Haarov klasifikator dobivamo usporedbom značajke sa pragom. Ovisno o rezultatu usporedbe, dobiva se pozitivan ili negativan odgovor. Zbog dva moguća odgovora Haarov klasifikator nosi naziv binarni. Samo jedan ovakav klasifikator neuspješan je u detekciji. On se naziva slabi klasifikator (eng. *weak classifier*) jer ga čini samo jedna značajka. Nužno je kombinirati više značajki kako bi se dobila dobra detekcija. Klasifikator koji nastaje kombinacijom više značajki naziva se jaki klasifikator (eng. *strong classifier*). Nužno je kombinirati više značajki da bi se dobili dobri rezultati detekcije. Zbog toga se koristi AdaBoost metoda koja ojačava slabe klasifikatore. Metoda od svih mogućih značajki odabire ključne koje čine klasifikator. Stvara se T pojedinačnih značajki (slabih klasifikatora). Konačni klasifikator je njihova linearna kombinacija s dodanim težinama. Težine su obrnuto proporcionalne greškama nastalim u učenju.

Selekcija slabog klasifikatora se odvija na sljedeći način. Za svaku značajku primjeri se poredaju ovisno o vrijednosti značajke za taj primjer. Optimalni prag za jednu značajku se može izračunati u jednom prolazu. Za svaki element sortirane liste 4 sume određuju vrijednost:

$T+$ - ukupan zbroj težina pozitivnih primjera

$T-$ - ukupan zbroj težina negativnih primjera

$S+$ - ukupan zbroj težina pozitivnih primjera ispod primjera kod kojeg je algoritam trenutno

$S-$ - ukupan zbroj težina negativnih primjera ispod primjera kod kojeg je algoritam trenutno

Izabire se ona značajka, tj. onaj slabi klasifikator koji ima minimalnu e vrijednost. e se računa prema formuli (3).

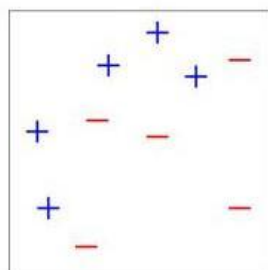
$$e = \min (S^+ + (T^- - S^-), S^- + (T^+ - S^+))$$

(3)

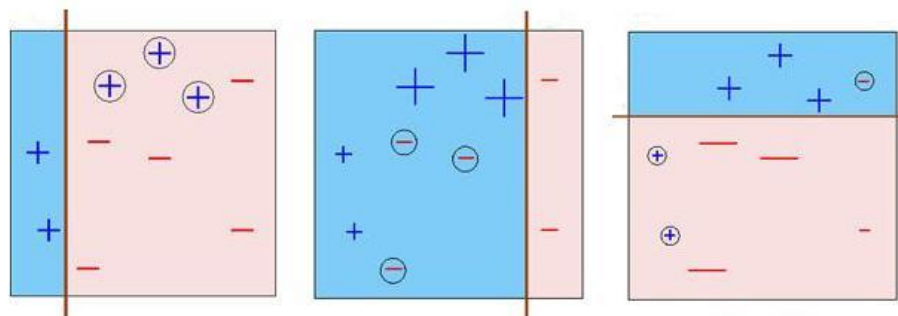
AdaBoost algoritam:

- odredi slabi klasifikator
- dodaj težine primjerima na kojima klasifikator griješi
- nauči novi slabi klasifikator uzimajući u obzir težine
- kombiniraj novi klasifikator sa starim
- dodaj težine primjerima na kojima klasifikator griješi
- nauči novi slabi klasifikator (uzimajući u obzir težine)
- kombiniraj novi klasifikator sa starim
- ...

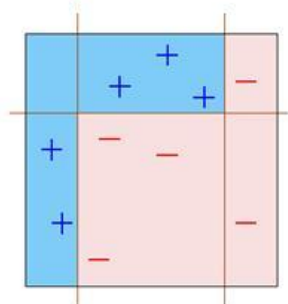
"+" - pozitivni primjeri
 "-" - negativni primjeri



slabi klasifikatori



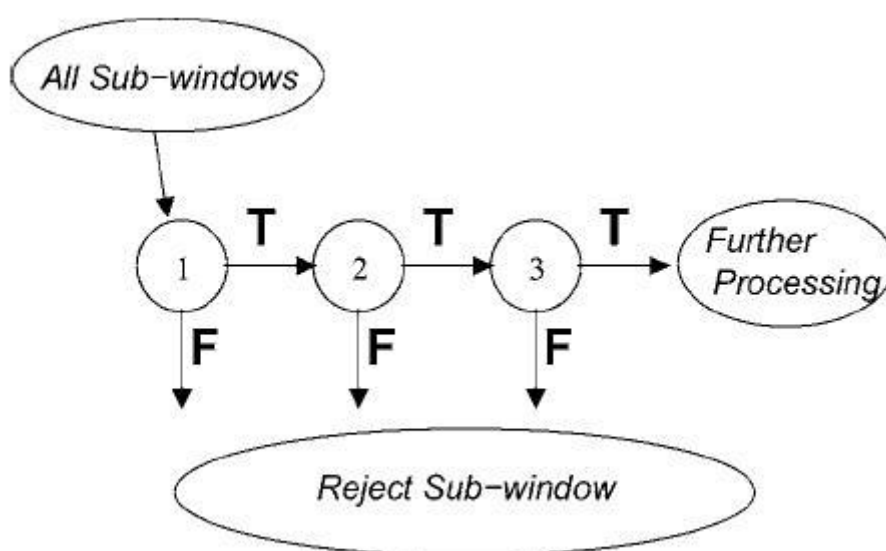
jaki klasifikator



Slika 2.3.1 : AdaBoost algoritam

2.4 ULANČAVANJE KLASIFIKATORA U KASKADU

Dobar klasifikator koristi jako puno značajki. Sadržaji okna koji se ispituju uglavnom su negativni. Određivanje vrijednosti svih značajki za svako okno predugo bi trajalo. Zato se koristi kaskada. Kaskada se sastoji od ulančanih jakih klasifikatora i ima oblik degeneriranog stabla odlučivanja. Prikazana je na *slici 2.4.1* (Viola, 2004). Nakon što se nauči klasifikator, dodaje se u kaskadu i računa se novi klasifikator koji u kaskadi treba biti iza njega. Negativni primjeri za učenje novome klasifikatoru postaju oni koje je prethodni klasifikator odbacio. Od početka prema kraju kaskade klasifikatori su sve složeniji. Ako neki klasifikator sadržaj okna označi kao negativni primjer, odmah prestaje klasificiranje. Jednostavni klasifikatori mogu uz malo računanja odbaciti većinu negativnih primjera. Tako kaskada u početku ispitivanja odbaci mnogo negativnih primjera i više se usmjeri na one primjere koji „obećavaju“. Tako se u većini slučajeva izbjegava korištenje složenijih klasifikatora koji dugo računaju. Korištenjem kaskade značajno se ubrzava detekcija.



Slika 2.4.1: Kaskada

Prema formuli (4) (Viola, 2004.) određuje se stopa detekcije kaskade.

$$D = \prod_{i=1}^K d_i, \quad (4)$$

D je stopa detekcije kaskade. K je broj klasifikatora u kaskadi. d_i je stopa detekcije svakog klasifikatora u kaskadi.

Prema formuli (5) (Viola, 2004) računa se stopa lažnih pozitiva kaskade.

$$F = \prod_{i=1}^K f_i, \quad (5)$$

F je stopa lažnih pozitiva kaskade. K je broj klasifikatora u kaskadi. f_i je stopa lažnih pozitiva svakog klasifikatora u kaskadi.

Ako se pri učenju klasifikatora kao parametre zadaju: $d_i = 0,995$, $f_i = 0,5$, broj klasifikatora koji se pokušavaju napraviti i koji će činiti kaskadu: 14, onda $D = 0,995^{14} = 95,11\%$ i $F = 0,5^{14} = 0.0061\%$. U stvarnosti se jako teško dobivaju ovi rezultati. Uglavnom se ne uspijeva napraviti svih 14 klasifikatora za kaskadu. U svakom jakom klasifikatoru koji je dio kaskade izgenerira se onoliko primjera koliko d_i i f_i odrede. Ukoliko nije moguće zadovoljiti d_i i f_i učenje se prekida i ostaju samo dotad izgenerirani klasifikatori.

3. PROGRAMSKO RJEŠENJE

U okviru rada bilo je potrebno automatizirati postupke ekstrakcije slika znakova pojedinačnih homogenih nadrazreda iz sveukupnih skupova za učenje i testiranje. Trebalo je i automatizirati postupke učenja klasifikatora i evaluacije kvalitete detekcije za svaki predloženi homogeni nadrazred. Napisan je program koji to obavlja. Program koristi neke gotove postupke koji su implementirani u biblioteci OpenCV.

3.1 OPENCV

Open Computer Vision je biblioteka koja sadrži mnoštvo algoritama iz područja računalnog vida. Biblioteka je dostupna za sve programske sustave, a pisana je u programskim jezicima C i C++. Sadrži algoritme za obradu slike, rad s video sadržajima, rad s matricama, strojno učenje... U radu se koriste 3 programa koji su dio OpenCV biblioteke. To su *opencv_createsamples.exe* koji siječe i skalira slike za učenje, *opencv_haartraining.exe* koji trenira klasifikator i *opencv_performance.exe* koji evaluira uspješnost detekcije.

3.2 OPIS PROGRAMA

Program je napisan u programskom jeziku C#. Pokreće se iz komandne linije. Program kao parametre prima:

- datoteku u kojoj su zadani nadrazredi
- datoteku u koju se spremaju rezultati detekcije
- opisnu datoteku slika za učenje
- opisnu datoteku slika za testiranje.

Opisne datoteke su u *.seq* formatu, a *opencv_createsamples.exe* kao parametar prima opisnu datoteku zapisanu u *.dat* formatu. Zato za svaki zadani homogeni nadrazred program opisne datoteke iz *.seq* formata program pretvara u *.dat* format.

Oblik datoteke *.seq* formata je:

```
kazaloDoSlike/vrstaPrometnogZnaka/koordinate(gornja lijeva)/visina/širina
```

Ovo je primjer linije iz takve datoteke:

```
[e:\sekvence\pune\B31_0025.bmp]:B32@(x=438,y=244,w=26,h=26)&B31@(x=441,y=272,w=24,h=26)
```

Oblik datoteke *.dat* formata je:

```
imeSlike          brojZnakovaNaSlici          koordinataXGornjegLijevogKuta  
koordinataYGornjegLijevogKuta  širinaZnaka  visinaZnaka
```

Gore navedeni redak iz *.seq* pretvoren u *.dat* izgleda:

```
B31_0025.bmp 2 438 244 26 26 441 272 24 26
```

Prilikom konverzije iz *.seq* u *.dat* program istovremeno ekstrahira retke. Odabir redaka koji će se ekstrahirati ovisi o nadrazredu za kojeg želimo pokrenuti učenje i evaluaciju. U opisnoj datoteci primjera za testiranje označeni su svi znakovi koji se nalaze na slici. Programu su potrebne oznake samo onih znakova koji pripadaju određenom nadrazredu.

Tako je potrebno prije preslikavanja oznake provjeriti pripada li znak određenom nadrazredu. Iz gornjeg primjera *.seq* datoteke znak *B32* neće biti opisan u *.dat* datoteci. Nakon primjene samo ovog kriterija, gornji redak pretvoren u *.dat* format će izgledati:

```
B31_0025.bmp 1 441 272 24 26
```

Ipak, gornji primjer iz *.seq* datoteke uopće neće biti prepisan u *.dat* datoteku jer postoji još jedan kriterij. Visina ni širina slike ne smije biti manja od 25 piksela.

Nakon podjele opisnih datoteka korištenjem *opencv_createsamples.exe* dobije se *.vec* datoteka koja sadrži skalirane slike zapisane u vektorskom formatu. Nakon skaliranja slike su u obliku kvadrata jer je to potrebno za rad algoritma Viole i Jonesa.

Nakon što program dobije *.vec* datoteku pokreće se učenje klasifikatora pozivom *opencv_haartraining.exe* koji kao parametre prima :

- ime klasifikatora koji će se naučiti
- *.vec* datoteku koja sadrži slike za učenje
- datoteku s popisom pozadina – slike iz perspektive vozača na kojima se ne nalaze pozitivni primjeri
- *nneg* – broj negativnih primjera koji se rade iz jedne pozadine
- parametri koji određuju način na koji program uči:
 - *nstage* (eng. *number of stages*)– željeni broj jakih klasifikatora koji čine kaskadu
 - *mhr* (eng. *minimum hit rate*)- udio pozitivnih primjera koji za svaki jaki klasifikator u kaskadi mora biti potvrđen
 - *mfa* (eng. *maximum false alarm*) – postotak koji određuje koliko najviše negativnih primjera smije biti označeno

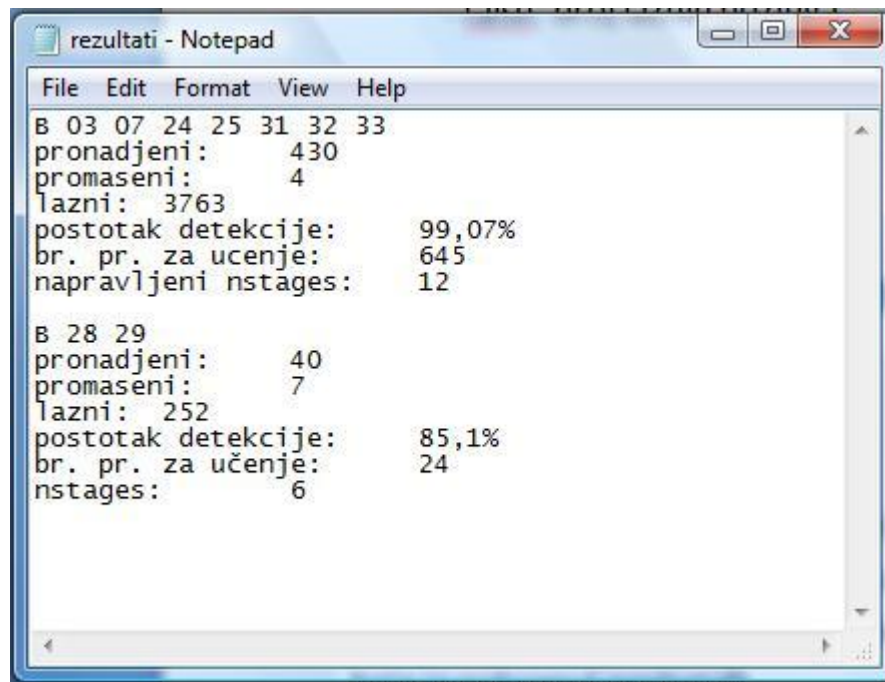
Nakon što se nauči klasifikator program poziva *opencv_performance.exe* koji evaluira kvalitetu detekcije. Parametri koje prima su:

- klasifikator
- opisnu datoteku skupa za testiranje
- datoteku u koju se spremaju rezultati detekcije
- *sf* (eng. *scale factor*) – određuje koliko se povećava okno svaki put kada ponovo kreće iz početka slike

Rezultati koje daje *opencv_performance.exe* se zapisuju u privremenu datoteku. Ta datoteka se parsira i iz nje se uzimaju tri vrijednosti:

- *hits*: broj pronađenih pozitivnih primjera u skupu za testiranje
- *missed*: broj pozitivnih primjera koji nisu pronađeni
- *false*: broj lažnih pozitiva

Program još računa stopu detekcije (*hits/missed*) te uz gore navedene vrijednosti, broj primjera za učenje i broj napravljenih klasifikatora koji čine kaskadu sprema ga u zadanu datoteku. Tu se nalaze rezultati za sve klase za koje je pokrenut postupak. Primjer sadržaja datoteke nakon što je program završio s radom nalazi se na *slici 3.2.1.*



slika 3.2.1: Zadana datoteka u koju se spremaju rezultati detekcije za zadane nadrazrede

B_03_07_24_25_31_32_33 i B_28_29

4. EKSPERIMENTALNI REZULTATI

4.1. PARTICIONIRANJE PROMETNIH ZNAKOVA U HOMOGENE NADRAZREDE

Iz skupa svih prometnih znakova predloženo je 14 homogenih nadrazreda. Članovi istog nadrazreda imaju sličan oblik, strukturu i značenje. Zbog toga se mogu detektirati istim klasifikatorom. Što su veće sličnosti, klasifikator je bolji, a time i detekcija. Ime nadrazreda sastoji se od slova koje označava razred znakova u nadrazredu i indeksa svakog znaka odvojenog podvlakom. Tipovi znakova su: znakovi opasnosti (*A*), znakovi izričitih naredbi (*B*), znakovi obavijesti (*C*), znakovi obavijesti za vođenje prometa (*D*), dopunske ploče (*E*) i promjenljivi prometni znakovi (*F*).

Ispod su prikazani svi predloženi nadrazredi. Za svaki nadrazred prikazani su svi ili nekoliko članova.

(1)

A_01_03_04_05_06_07_08_09_10_11_12_13_14_15_17_20_22_23_25_32_33_34_41_43_44_45_46

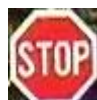
Nadrazred se sastoji od skoro svih znakova opasnosti. Izuzetak je andrijin križ. Čine ga trokutasti prometni znakovi. Neki članovi su prikazani ispod.



(2) B_01



(3) B_02



(4) B_03_07_24_25_31_32_33

Nadrazred se sastoji od okruglih prometnih znakova koji imaju crveni obrub, a u sredini dominira bijela boja. Neki od članova su prikazani ispod.



(5) B_28_29



(6) B_39



(7) B_50_52_57_59_60_61

Nadrazred čine okrugli plavi znakovi koji u sebi imaju bijelu strelicu ili strelice. Neki su prikazani na slikama ispod.



(8) C_02



(9) C_08



(10) C_11_13_16

Nadrazred čine bijeli znakovi. Prekriženi su crtom sive boje. Svi članovi su prikazani na slikama ispod.



(11) C_35



(12) C_44



(13) C_70



(14) E_35



4.2 REZULTATI EVALUACIJE

Za svaki predloženi nadrazred napravljena je evaluacija. Za izradu klasifikatora korištene su vrijednosti parametara koje su se na prethodnim radovima pokazale najbolje. To su:

- *nstage*: 14
- *nneg*: 10000
- *mhr*: 0,995
- *mfa*: 0,5

Za učenje se koristi 712 pozadina. Za detekciju se koristi *scale factor* iznosa 1.05. Za ovu vrijednost parametra *scale factor* dobiva se dobra detekcija, ali ne prevelik broj lažnih pozitiva. Kada se još smanji *scale factor* dobiva se malo veći broj pronađenih pozitiva, ali puno veći broj lažnih pozitiva i duža detekcija. Uspješnost detekcije nadrazreda jednaka je broju pronađenih pozitiva u odnosu na sve pozitive u skupu za testiranje tog nadrazreda.

U *tablici 4.2.1* prikazana je ukupna statistika. Rezultati učenja i evaluacije za svaki homogeni nadrazred prikazani su u *tablici 4.2.2*. Iz primjera koji se koriste za evaluaciju uklonjeni su znakovi sa rezolucijom manjom od 25x25 piksela. Početna veličina pomičnog okna koji se koristi unutar *opencv_performance.exe* je 24x24 piksela. Znakove sa malom rezolucijom teško je detektirati, a pogotovo znakove čija je rezolucija manja od početne veličine okna. U *tablici 4.2.3* su usporedbe uspješnosti detekcije kad se uklone znakovi manji od 25x25 piksela i kad ne.

tablica 4.2.1: Statistika

prosječan broj primjera za učenje	prosječan broj klasifikatora u kaskadi	prosječna uspješnost detekcije	broj lažnih pozitiva po pronađenom pozitivu
314	8	79,73%	7.67

tablica 4.2.2: Rezultati učenja i evaluacije

ime homogenog nadrazreda	broj pozitivnih primjera za učenje	nstage napravljen	pronađeni pozitivivi	promašeni pozitivivi	lažni pozitivivi	uspješnost detekcije
(4)	645	12	377	0	3673	100,00%
(14)	224	9	14	0	199	100,00%
(8)	213	8	188	3	1771	98,43%
(9)	337	11	48	1	632	97,95%
(12)	46	8	138	5	826	96,50%
(1)	2150	12	852	34	3850	96,16%
(2)	106	8	7	1	97	87,50%
(11)	59	5	7	1	31	87,50%
(10)	53	10	120	18	1916	86,95%
(5)	24	6	40	7	252	85,11%
(7)	59	11	25	8	250	75,75%
(3)	456	10	5	3	89	62,50%
(6)	18	5	18	25	515	41,86%
(13)	9	2	0	2	16	0,00%

tablica 4.2.3: Uspješnost detekcije

ime homogenog nadrazreda	bez slika manjih od 25x25	sa slikama manjim od 25x25
(4)	100,00%	80,88%
(14)	100,00%	66,67%
(8)	98,43%	79,56%
(9)	97,95%	82,81%
(12)	96,50%	84,50%
(1)	96,08%	90,35%
(2)	87,50%	62,50%
(11)	87,50%	58,33%
(10)	86,95%	87,50%
(5)	85,11%	63,24%
(7)	75,75%	41,93%
(3)	62,50%	18,75%
(6)	41,86%	30,77%
(13)	0,00%	0,00%

Nadrazred (4) ima 100%-tnu uspješnost detekcije. Za izradu klasifikatora koristi se relativno visok broj primjera. 100%-tnoj uspješnosti mnogo pridonosi uklanjanje znakova manjih od 25x25 piksela iz skupa za testiranje. Sa tim znakovima uspješnost detekcije je 80,88%.

Nadrazred (14) također ima 100%-tnu uspješnost detekcije, iako ne koristi mnogo primjera za učenje. Razlog 100%-tne uspješnosti je mali broj primjera za testiranje. Da se koristi više primjera za testiranje, uspješnost bi sigurno bila manja. Uspješnosti detekcije uvelike pridonosi velika homogenost nadrazreda. Homogenost je velika zbog toga što nadrazred sadrži samo jednu vrstu znakova.

Uspješnost detekcije nadrazreda (8) je 98,43% što je relativno visoko, a koristi se relativno mali broj primjera za učenje. Razlog tome je velika homogenost nadrazreda koji sadrži samo jednu vrstu znakova.

Uspješnost detekcije nadrazreda (9) je 97,95%, a koristi se prosječan broj primjera za učenje u odnosu na ostale nadrazrede. U jedinom primjeru gdje je promašen pozitivni primjer, klasifikator ga zapravo pronalazi, ali ne pogađa veličinu dobro. Visokoj uspješnosti detekcije pridonosi jako velika homogenost razreda koji sadrži samo jednu vrstu znakova.

Uspješnost detekcije nadrazreda (12) je relativno visoka unatoč malom skupu za učenje. Uspješnosti detekcije pridonosi velika homogenost razreda koji sadrži samo jednu vrstu znakova. U primjerima za evaluaciju znakovi se često nalaze na desnom rubu slike pa su mali. To otežava detekciju.

Uspješnost detekcije nadrazreda (1) iznosi 96,08%. Visokom postotku pridonosi mnogo primjera za učenje. Velika većina promašenih pozitivnih primjera se događa kod detekcije znaka A33 (obilježen pješački prijelaz). Osim znakova male veličine, čest razlog promašaja pozitivnih primjera je loše određivanje njegove veličine i tamni znakovi. Tamni znakovi su problem jer smanjuju kontrast pa ih klasifikator slabije detektira u nekim slikama. Znak A15 (suženje ceste) ne može se detektirati jer je nakošen. Primjer sa znakom A15 prikazan je na *slici 4.2.1*. Za razliku od većine ostalih homogenih nadrazreda, ovaj ima jako mnogo različitih vrsta pa je homogenost mala u odnosu njih. Relativno mala homogenosti ne pridonosi uspješnosti detekcije. Korištenjem znakova manjih od 25x25 iz skupa za testiranje uspješnost detekcije nadrazreda (1) je 90,35%. To je najveća uspješnost detekcije nekog nadrazreda kada su u skup za testiranje uključeni znakovi manji od 25x25 piksela.

Uspješnost detekcije nadrazreda (2), (10) i (11) iznosi oko 87%. Sva tri nadrazreda sadrže relativno malo primjera za učenje. U nadrazredu (10) problem detekciji predstavljaju znakovi koji su dosta udaljeni od kamere. Za svaku četvrtu sliku (najudaljenija od kamere iz skupa slika potpuno istih znakova) jako je upitno hoće li se detektirati.

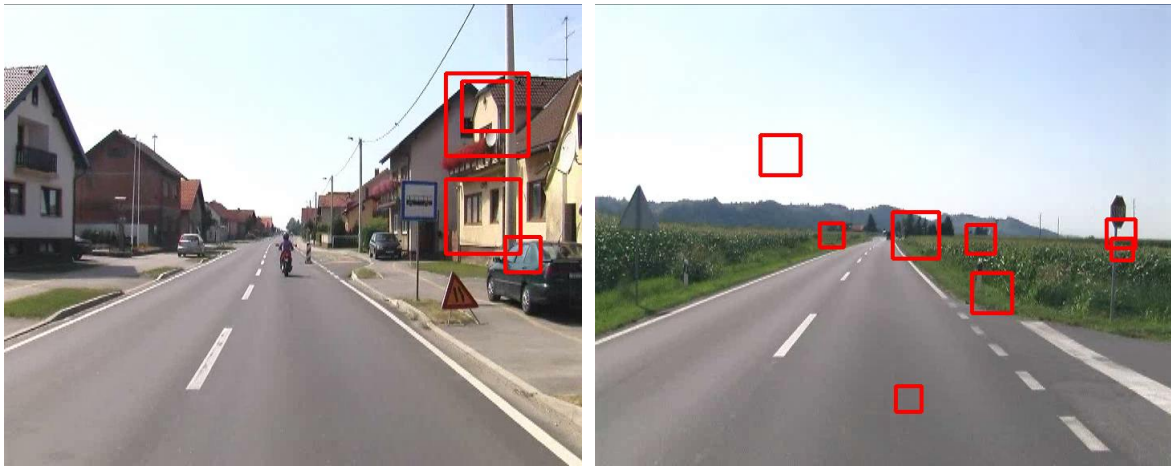
Uspješnost detekcije nadrazreda (5) iznosi 85,10%. Na četiri slike znak B28 je oštećen. Samo je jedan pozitivan primjer detektiran od ta četiri. Na tri slike je prilično taman znak B28 i ni jedan pozitivan primjer nije detektiran od tri. Zbog ovih stvari dobivena uspješnost detekcije je manja nego što bi trebala biti.

Uspješnost detekcije nadrazreda (7) iznosi 75,75%. Detekcija nije kvalitetna, ali koristi se malo primjera za učenje.

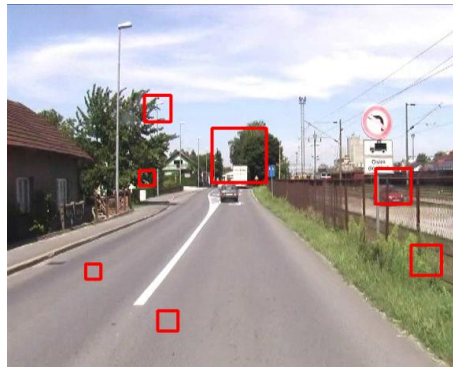
Nadrazred (3) unatoč relativno visokom broju primjera za učenje ima malu uspješnost detekcije koja iznosi 62,5%. Najveći razlog tome je što su slike znaka *stop* (jedinu član nadrazreda) na primjerima za evaluaciju uglavnom nakošene i loše kvalitete. To je zato što se na velikom dijelu slika znak nalazi na sporednoj cesti koja se priključuje glavnoj. Primjer je prikazan na slici 4.2.1.

Nadrazred (13) nema nimalo uspješnu detekciju. Razlog tome je premali broj primjera za učenje. Kaskada se sastoji od samo dva klasifikatora. Svi znakovi na primjerima za evaluaciju su loše kvalitete.

Ostali nadrazredi s jako malim skupom za učenje nisu predloženi u ovom radu. Neki nadrazredi se nisu mogli predložiti jer imaju oblik pravokutnika. Algoritam Viole i Jonesa zahtjeva kvadratne slike iz kojih uči. Nema smisla skalirati takve slike jer su na slikama za testiranje primjeri pravokutnika. Primjer nadrazreda čiji članovi imaju oblik pravokutnika bio bi *E_01_02_03_04_08_12_18_19_34*.



Slika 4.2.1: Nemogućnost detekcije zbog nakošenosti



Slika 4.2.2: Nemogućnost detekcije zbog oštećenosti



Slika 4.2.3: Nemogućnost detekcije zbog niske rezolucije znaka

5. ZAKLJUČAK

U okviru ovog rada predložena je particija skupa svih prometnih znakova u homogene nadrazrede koji se detektiraju istim klasifikatorom. Ipak, neki homogeni nadrazredi nisu predloženi zbog premalog broja slika za učenje, a neki zbog toga što znakovi koji ih čine imaju oblik pravokutnika. Potonji problem bi se eventualno mogao riješiti na način da primjeri za učenje postanu izrezani zajednički dijelovi slika.

Za nadrazrede s nemalim i kvalitetnim skupom za učenje i testiranje dobivena je i uspješna detekcija. Uspješna detekcija nekih nadrazreda pokazala se većom nego kod trokutastih prometnih znakova koji imaju relativno velik broj primjera za učenje. Razlog tome je znatno veća homogenost tih nadrazreda. Detekciju najviše otežavaju mali, tamni i nakošeni znakovi.

Napravljen je program koji sadrži mehanizme koji se traže u zadatku. Program bi se mogao nadograditi *GUI*-em preko kojeg korisnici mogu zadavati nadrazrede, odabirati koje mehanizme žele koristiti, podesiti parametre za učenje i evaluaciju...

6. LITERATURA

[1] Viola P., Jones M., Robust Real-Time Face Detection, International Journal of Computer Vision 57(2), 137–154, 2004.

[2] Benussi T., Bosilj P., Čuljak M., Jurić D., Miklenić M., Morava M., Trbojević A., Zadrija L., Pronalaženje i praćenje prometnih znakova, Projekt, Fakultet elektrotehnike i računarstva, Zagreb, 2009.

[3] Brkić, K., Pinz, A., Šegvić, S.: Traffic sign detection as a component of an automated traffic infrastructure inventory system, OAGM'09, Stainz, 2009.

EKSPERIMENTALNA EVALUACIJA USPJEŠNOSTI DETEKCIJE RAZLIČITIH RAZREDA PROMETNIH ZNAKOVA

SAŽETAK

U ovom radu opisan je postupak detektiranja objekata prema Violi i Jonesu. Razmatra se primjena postupka za detekciju prometnih znakova. Predložen je skup homogenih nadrazreda prometnih znakova za koje se može napraviti zajednički klasifikator. Za svaki nadrazred naučen je klasifikator i evaluirana uspješnost detekcije. Prikazani su dobiveni rezultati. Za dio nadrazreda dobivena je kvalitetna detekcija.

Implementirana je automatska ekstrakcija redaka iz opisne datoteke, automatsko učenje klasifikatora i automatska evaluacija uspješnosti detekcije za zadane homogene nadrazrede. Za učenje klasifikatora i evaluaciju uspješnosti detekcije koristi se OpenCV biblioteka.

Ključne riječi: detekcija prometnih znakova, Viola i Jones, integralna slika, AdaBoost

EXPERIMENTAL PERFORMANCE EVALUATION OF DETECTION OF DIFFERENT CLASSES OF TRAFFIC SIGNS

ABSTRACT

This paper describes the object detection technique, proposed by Viola and Jones. Application of the procedure for the detection of traffic signs has been taken into consideration. This paper proposes a set of the homogenous superclasses whose common classifier can be learned. For each superclass, single classifier has been learned and performance of detection has been evaluated. These evaluations are shown in the paper. For some of the superclasses, high detection performance has been achieved.

Automatic extraction of rows from the descriptive file, automatic classifier learning and automatic performance evaluation of the default superclass have been implemented. OpenCV library has been used for training classifiers and performance evaluation.

Keywords: traffic sign detection, Viola and Jones, integral picture, AdaBoost

PRIVITAK

Uz rad je priložen DVD-ROM sa izvornim kodom i izvršnom verzijom razvijenog programskog rješenja te skupovi slika za učenje i testiranje.