

Softverska segmentacija i klasifikacija saobraćajnih znakova

Napravljen je algoritam za prepoznavanje saobraćajnih znakova koji omogućuje vozaču da u svakom trenutku ima uvid u trenutna ograničenja i upozorenja na određenoj deonici puta. Algoritam se bazira na jasno istaknutoj osobini svih saobraćajnih znakova – boji. Nakon detekcije znaka, izdvaja se slika koja se nalazi u sredini znaka, pronalazi oblik znaka i proverava veličina. U poslednjem koraku vrši se klasifikacija, i kao rezultat dobija konačna informacija o tome koji se znakovi nalaze na slici. Testiranje je pokazalo da algoritam daje zadovoljavajuće rezultate pri klasifikaciji i segmentaciji, uz određena ograničenja koja je daljim razvojem moguće prevazići.

Uvod

Jedan od ciljeva razvoja modernih saobraćajnih sistema jeste povećanje bezbednosti na putu. Softversko prepoznavanje saobraćajnih znakova omogućava vozaču da u svakom trenutku ima informaciju o ograničenjima i zabranama na deonici puta na kojoj se nalazi, eliminišući faktor vozačeve nepažnje i znatno povećavajući bezbednost na putu. Razvoj ovakvog softvera je deo i mnogo šireg problema konstruisanja potpuno autonomnog vozila.

Saobraćajni znaci su dizajnirani tako da budu što uočljiviji. Radi toga oni se jasno ističu svojom bojom i oblikom, a upravo ove osobine se mogu iskoristiti pri razvoju softvera za njihovo prepoznavanje.

Cilj ovog rada je postizanje što boljih rezultata u prepoznavanju znakova sa realne slike korišćenjem postojećih i implementacijom novih metoda. Skup znakova na kojem je algoritam testiran pripada skupu

znakova zabrana i upozorenja koji se koriste na prostoru Republike Srbije ali ga je moguće adaptirati i za ostale klase saobraćajnih znakova.

Algoritam se sastoji iz dva dela. Prvi deo čini pronalaženje regiona u kome se nalazi znak na ulaznoj slici – segmentacija, a drugi čini prepoznavanje znaka u datom regionu – klasifikacija.

Segmentacija regiona u kojima se nalaze znaci (dalje ROI, eng. region of interest) se vrši na osnovu boje kojom se znaci jasno ističu. ROI mora da sadrži određenu količinu piksela i da bude određene veličine što će omogućiti precizno prepoznavanje; u suprotnom se odbacuje.

Klasifikacija se vrši za svaki ROI koji daje prvi deo algoritma. Prvo se pronalazi oblik znaka na osnovu vrednosti piksela u uglovima ROI i pretraga se sužava na samo taj oblik znaka. Zatim se pravi pozadinska maska kojom se izdvaja unutrašnjost znaka. Izdvojena slika se normalizuje po veličini i poredi sa pripremljenim šablonima da bi se izračunao procenat poklapanja i na taj način prepoznao znak unutar ROI.

Pri izradi programa korišćena je biblioteka openCV. Ova biblioteka je dizajnirana upravo za rešavanje problema iz oblasti kompjuterskog vizualnog prepoznavanja (eng. computer vision) u šta pripada i problem obrađen u ovom radu.

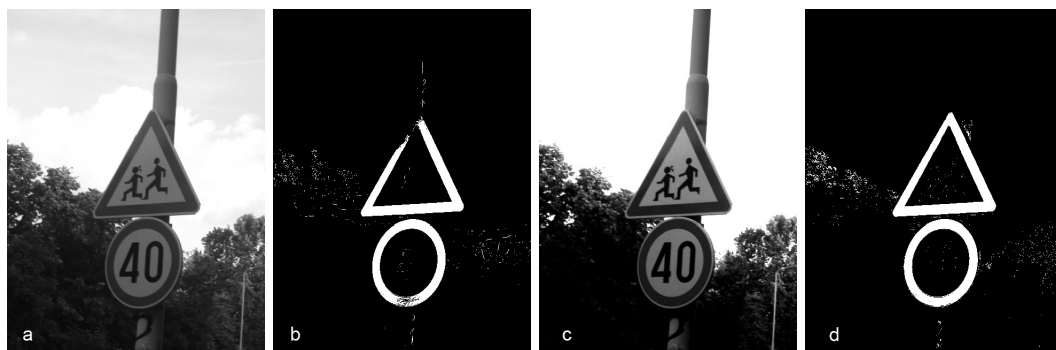
Detaljan opis segmentacije

1. Priprema ulazne slike za obradu

Ulazna slika se nalazi u RGB modelu boja i predstavlja fotografiju (ili frame videa) u kojoj se nalaze saobraćajni znaci. Ukoliko bi se slika prosledila bez prethodne pripreme dobio se loš rezultat pri odbacivanju piksela neodgovarajuće boje (eng.

Miloš Stojanović (1993), Beograd, Bul. Arsenijeviča Černojevića 183/13, učenik 2. razreda IX beogradske gimnazije „Mihailo Petrović Alas”

MENTORI:
*Dragan Toroman
Milan Gornik*



Slika 1. a) ulazna slika, b) trešhold nepripremljene slike (nepotpune konture), c) slika posle pripreme – istaknute boje, d) trešhold pripremljene slike (pune konture)

Figure 1. a) original image, b) thresholding of original image (incomplete contours), c) edited image (higher contrast), d) thresholding of edited image (full contours)

thresholding, dalje trešholdovanje) za koji je teško i vrlo neprecizno odrediti ROI (slika 1b). Zbog toga se vrši priprema slike gde je cilj dodatno istaknuti boje (slika 1c) da bi se posle trešholda boje dobila puna kontura znaka koju je moguće identifikovati kao ROI (slika 1d). Isticanje boja se vrši na osnovu ulazno/izlazne krive (eng. input/output curve) koja se na x-osi sužava da bi se novodobijena kriva ponovo rastegla na opseg od 0 do 255.

2. Trešholdovanje

Trešholdovanje predstavlja odbacivanje piksela čije vrednosti ne ulaze u zadati opseg. Slika kojom manipuliše program je u RGB modelu boja i trešholdovanje se vrši na osnovu poređenja RGB komponentata. Ukoliko je tražena crvena boja biće zadržani pikseli u kojima je vrednost u R kanalu znatno veća u odnosu na vrednosti u G i B kanalu. Rezultat trešholdovanja je crno-bela slika u kojoj zadržani pikseli imaju belu, a odbačeni crnu boju.

Pri rešavanju ovog problema razmotreno je korišćenje HSV modela u kome se piksel predstavlja trima komponentama: bojom (hue), zasićenošću (saturation) i intenzitetom (value), a hue je dat kao ugao. To je pogodno za trešholdovanje jer se boja, koja je ključna osobina piksela pri raspoznavanju znakova, svodi na jednu promenljivu nezavisno od zasićenosti i intenziteta, pa bi u primeru traženja crvene boje trebalo zadržati piksele koji imaju vrednost hue komponente od npr. 340° – 20° . Problem kod ovog modela je to što je potrebno konvertovati ulaznu RGB sliku u HSV, što nikada ne može biti potpuno precizno.

Najveći broj saobraćajnih znakova je određen crvenom ili plavom bojom koje imaju svoju komponentu u RGB modelu boja, pa konverzija u HSV ne donosi veliko povećanje kvaliteta pri trešholdovanju, a često može dovesti i do pada kvaliteta trešholdovanja.

3. Pronalaženje kandidata

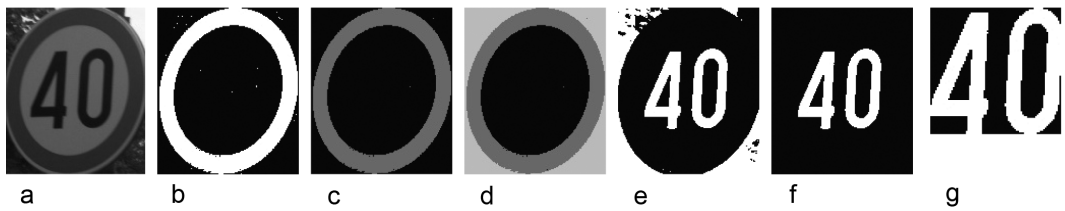
Crno-bela slika koja je dobijena kao rezultat trešholdovanja služi kao ulaz za flood fill algoritam kojim se detektuju oblasti povezanih piksela bele boje. Rezultat flood fill-a je ROI koji može biti kandidat za klasifikaciju ukoliko se njegova čija se površina, dimenzije i broj belih piksela nalaze unutar prethodno definisanih granica. Ukoliko je ROI identifikovan kao kandidat njegove koordinate se čuvaju da bi se koristio u klasifikaciji.

Detaljan opis klasifikacije

1. Parametri klasifikacije

Parametri na osnovu kojih se vrši klasifikacija znakova su boja, oblik i sadržaj sredine znaka. Prvi parametar je već određen u prethodnim koracima i ograničiće pretragu na znakove samo određene boje. Parametar oblika može znatno da suzi pretragu i tako dovede do rasta performansi.

S obzirom da su saobraćajni znaci jednostavni geometrijski oblici, oblik znaka se može jednostavno i dovoljno precizno utvrditi posmatranjem piksela u uglovima ROI.



Slika 2. Izdvajanje sadržaja znaka

Figure 2. Singling out the content

2. Izdvajanje sadržaja znaka

Boja i oblik nisu dovoljni za prepoznavanje znaka; oni služe samo za sužavanje izbora mogućih poklapanja. Najbitniji parametar je slika koja se nalazi u sredini znaka – sadržaj. Upoređivanjem sadržaja znaka sa pripremljenim šablonima možemo dobiti pouzdano prepoznavanje.

Algoritam za izdvajanje sadržaja iz sredine znaka funkcioniše na sledeći način:

- Na ulazu dobija kandidata u obliku pravougaonog ROI (slika 2a),
- Na toj slici izvrši trešholdovanje tako da se zadržava data koja za taj ROI (određena u ranijim koracima (slika 2b),
- Vrší flood fill da bi se pronašao obod znaka (slika 2c),
- Pronalazi pozadinu primenom flood fill algoritma i ona sa obodom čini masku za izdvajanje informacije (slika 2d),
- Vrší trešholdovanje tako da ostanu samo izrazito crni pikseli koji će predstavljati sadržaj, ali ostaje i deo pozadine – višak koji se treba odbaciti maskom (slika 2e),
- Primenjuje se maska na sliku dobijenu iz prethodnog koraka i kao rezultat se dobija sadržaj (slika 2f),
- Slika se seče tako da nastaje novi region u kome se nalazi samo sadržaj (slika 2g).

3. Poređenje sa šablonima

Slika koja se dobije iz prethodnog koraka (sadržaj) je pogodna za poređenje sa bazom šablona. Potrebno je još samo izvršiti normalizaciju tako da veličina sadržaja odgovara veličini šablona u bazi.

Za poređenje sa šablonima (pattern matching) potrebni su sledeći podaci:

- true positives (tp) – broj piksela konture (beli) sadržaja koji se poklapa sa pikselima konture šablona,
- false positives (fp) – broj piksela konture šablona koji se ne nalaze u konturi sadržaja,
- true negatives (tn) – broja piksela pozadine (crni) sadržaja koji se poklapaju sa brojem piksela pozadine šablona,
- false negatives (fn) – broj piksela pozadine šablona koji se ne nalaze u pozadini sadržaja.

Na osnovu ovih podataka moguće je izračunati procenat poklapanja sadržaja i šablona. U ovom radu je primenjen metod računanja razlike između udela tp u odnosu na ukupan broj pozitivna (preciznost) i udela fn u odnosu na ukupan broj negativna označen kao prilagođeni algoritam (formula 1). U testiranje je takođe uključena i mera preciznosti testiranja (F-score algoritam) koja uzima u obzir odnos preciznosti (P) i senzitivnosti (S) (formula 2). Vrednosti koje se mogu dobiti pri prilagođenom metodu su u opsegu od -100 do $+100$. One se dalje mogu normalizovati na opseg od 0% do 100% .

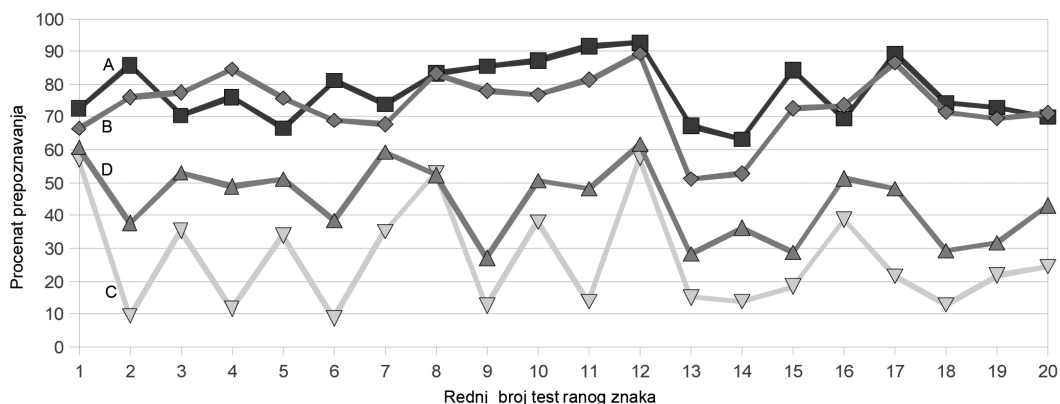
$$P = \frac{tp}{tp + fp} - \frac{fn}{tn + fn} \quad (1)$$

$$F = 2 \times \frac{P \times R}{P + R} \quad (2)$$

gde je $P = \frac{tp}{tp + fp}$, a $S = \frac{tp}{tp + fn}$

Rezultati

Testiranje algoritma vršeno je na slikama rezolucije 640×480 piksela u kojima je znak zauzimao relativno veliki deo površine. Na jednoj slici se može naći veći broj znakova. Fotografije su načinjene



Slika 3. Rezultati testiranja algoritama: A – prilagođeni, B – F-skor, C – najbolji odbačeni prilagođeni, D – najbolji odbačeni F-skor

Figure 3. Algorithm test results: A – adjusted, B – F-score, C – best rejected adjusted, D – best rejected F-score

u različitim delovima dana i samim tim pod različitim osvetljenjem. Algoritam se pokazao uspešnim pri segmentaciji, sa malim procentom nepronađenih i netačno pronađenih znakova.

Pri klasifikaciji su testirana oba algoritma. Pored procenta konačnih poklapanja kandidata sa bazom za oba algoritma, prikazani su i procenti najboljeg odbačenog poklapanja kandidata sa bazom (slika 3).

Prilagođeni algoritam i F-score algoritam su dali slične konačne rezultate. Kod F-score algoritma postoji manja razlika između konačnog i najboljeg odbačenog rezultata nego kod prilagođenog, i usled toga je moguće očekivati češće greške kod ovog algoritma nego kod prilagođenog.

Prilagođeni algoritam bolje diferencira konačan i najbolji odbačeni rezultat i usled toga daje pouzdanije rezultate.

Okruzija nije rešavana, tako da preklapanje znakova ili pojavljivanje izrazito crvene pozadine iza znaka dovodi do netačnih rezultata. Takođe je pretpostavljen kontinuitet cele spoljne konture znaka (crvene linije), koju u realnosti može narušiti na primer veća grana koja prelazi preko znaka.

Ovi problemi se mogu prevazići detektovanjem ivica i na osnovu toga izdvojiti oblik znaka od pozadine. Pronalaženjem ivica znaka moguće je pretpostaviti kakav će položaj imati kontura u prekinutom delu i time rešiti problem kontinuiteta.

Zaključak

Algoritam je pokazao dobre performanse, jer se bazira na korišćenju metoda koje ne zahtevaju veliku procesorsku snagu. Rezultati su uglavnom zadovoljavajući, uz ograničenja koja je u daljem razvoju moguće prevazići. Trenutno je algoritam implementiran na manjem skupu znakova (upozorenja i opasnosti), a krajnji cilj je prepoznavanje svih znakova.

Zahvalnost. Autor se zahvaljuje Draganu Toromanu, Zoranu Rilaku, Milanu Gorniku i Milošu Saviću na saradnji, podršci i značajnoj pomoći pri nastajanju ovog rada.

Literatura

Escalera A., Moreno L., Salichs M. A., Armingok J. M. 1997. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, **44** (6): 848.

Senekal F. P. 2008. Traffic Sign Detection and Classification using Color and Shape Cues. *19th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA 2008)*. Cape Town, South Africa, 27-28 November 2008.

Dostupno na:
http://researchspace.csir.co.za/dspace/bitstream/10204/3319/1/Senekal_2008.pdf

Shneie M. 2005. Road Sign Detection and Recognition. Dostupno na:
http://www.isd.mel.nist.gov/documents/shneier/Road_Sign_Detection.pdf

OpenCV Biblioteka:
<http://opencv.willowgarage.com/wiki/>

Miloš Stojanović

Software Classification and Segmentation of Traffic Signs

Road safety is one of the key problems of modern road traffic. Road safety can be compromised due to various technical reasons (bad road, vehicle mal-

function, etc.), but more importantly due to the inattentiveness of the vehicle operator. This is the main cause of a large number of traffic incidents. The goal of this project is to devise an efficient algorithm for the automated recognition of traffic signs, thus providing the vehicle operator with a list of limitations and warnings in effect for a specific section of the road. The algorithm is based on a prominent characteristic of all traffic signs – color. After a sign is detected, the image in the center of the sign is extracted, the shape of the sign is established and its size is verified. In the final step classification takes place, the result of which is the information about all the traffic signs recognized in the input image. Testing has showed that the method outlined in this paper provides satisfactory results for the classification and segmentation, and that the current limitations of the algorithm are possible to overcome with further development. 