

# Odzivanje avtonomnega vozila na slikovne simbole

Ernest Gungl, Zmago Brezočnik

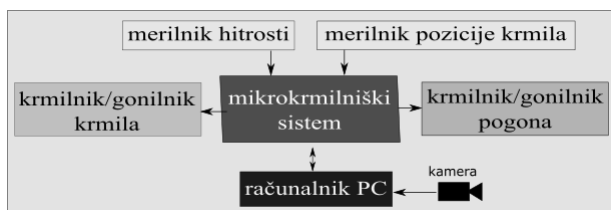
Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru  
Koroška cesta 46, 2000 Maribor, Slovenija  
E-pošta: ernest.gungl@um.si, zmago.brezocnik@um.si

## Responding of an Autonomous Vehicle to Image Symbols

**Abstract.** This paper presents the implementation of a computer vision software for a small autonomous vehicle that recognises image symbols like traffic signs in its environment and responds accordingly. The presented work is an enhancement of our basic version of the vehicle [1], which was able only to track a road. Since the creation of training examples to learn the neural network of the vehicle takes up a good deal of time, we developed a simple framework based on Python scripts. Neural network settings for image symbols detection and recognition are also described. Because the recognition of image symbols is not perfect, we wrote an algorithm that allows a correct final decision based on many temporary decisions.

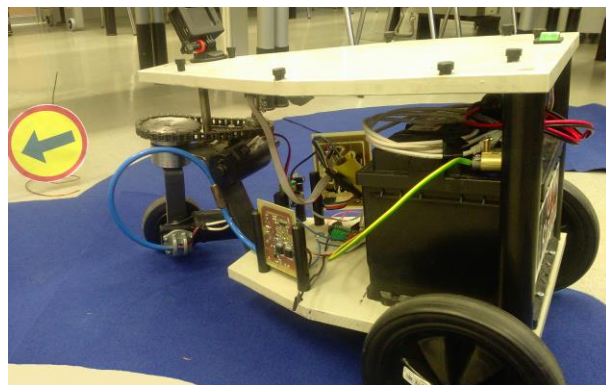
### 1 Uvod

V prispevku predstavljamo implementacijo računalniškega vida v kombinaciji z metodami strojnega učenja za razpoznavanje slikovnih simbolov (prometnih znakov) na majhnem testnem samovozečem vozilu. Vsebina se navezuje na prispevek z naslovom "Self-Driving Car Based on Machine Learning" [1], v katerem smo opisali implementacijo testnega vozila, opremljenega s kamero, ki na osnovi uporabljenih algoritmov avtonomno vozi po razpoznanem cestišču. Slika 1 prikazuje osnovne komponente vozila, slika 2 pa izdelani prototip.



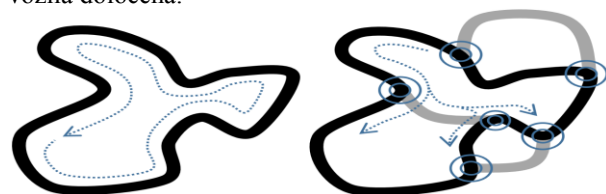
Slika 1: Blokovna shema krmilno-gonilnega dela vozila

Vozilo je opremljeno s senzorji za merjenje trenutnega položaja krmila, hitrosti vozila in krmilnikov/gonilnikov za nastavljanje želenega položaja in hitrosti. Senzorji so povezani na osrednji mikrokrmilniški sistem, ki komunicira z osebnim računalnikom, na katerem izvajamo omenjene algoritme.



Slika 2: Izdelani prototip vozila

V prispevku torej izhajamo iz obstoječega testnega vozila, ki avtonomno upravlja krmilo, tako da vozilo med vožnjo ostaja na cestišču [1]. Če cestišče ne vsebuje križišč in je morda izvedeno v obliki zanke, takšno vozilo napreduje po predvideni poti (slika 3 levo). Če so sestavni deli cestišča tudi križišča, označena s krožci na sliki 3 desno, napredovanja vozila po njem več ne moremo zagotovo napovedati. Če pa v križišču postavimo ustrezne slikovne simbole in vozilo nadgradimo, tako da jih razpozna, je nadaljnja pot vozila določena.



Slika 3. Enostavno in zahtevnejše cestišče

Vozilo smo nadgradili tako, da se pravilno odziva na šest slikovnih simbolov – prometnih znakov (povečaj hitrost, zmanjšaj hitrost, razveljavi hitrost, zavij levo, zavij desno in ustavi), ki so prikazani na sliki 7.

Želeli smo izdelati avtonomno vozilo, ki bi se zanesljivo obnašalo v konkretnem (omejenem) laboratorijskem oziroma industrijskem okolju. Zaradi želje po hitri realizaciji projekta smo izhajali kar iz lastne implementacije nevronske mreže v programskih jezikih Python in MatLab, uporabljene za samostojno vožnjo vozila v [1]. Poleg te že obstoječe nevronske mreže smo aplikaciji dodali še dve nevronske mreže – za zaznavanje in razpoznavanje obestnih prometnih znakov. Čeprav obstaja več prostodostopnih podatkovnih baz s prometnimi znaki [3, 4], smo se zaradi majhnega števila slikovnih simbolov, ki jih želimo razpoznati, in prvotne želje po unikatnih

slikovnih simbolih odločili za lastno izdelavo. Ker v fazi »učenja« vozila potrebujemo veliko količino učnih primerov, smo za njihovo lažjo tvorbo zgradili namensko okolje, napisano v programskem jeziku Python.

Za razpoznavanje slikovnih simbolov na sliki (npr. prometnih znakov), sta v splošnem potrebna dva koraka – zaznavanje simbola in njegovo razpoznavanje (klasifikacija) [7]. Zaznavanje objektov je najpogosteje izvedeno s pomočjo drsečega okna ali klasifikacije množice predlaganih lokacij [9, 12]. Ker pri metodi drsečega okna čez celotno sliko z drobnim korakom pomikamo pravokotnike različnih velikosti, da bi vanje ujeli iskan slikovni simbol, je metoda časovno zahtevna [2, 8, 9], saj mora algoritem za klasifikacijo pregledati veliko količino slik. Težavo olajša vpeljava grafičnih procesnih enot GPUs (Graphics Processing Units), ki omogočajo hitro vzporedno računanje [8]. Zaznavanje s pomočjo klasifikacije predlaganih lokacij je lahko mnogo hitrejšo, saj je število predlaganih lokacij, ki jih mora nevronska mreža (pogosto konvolucijska) pregledati, v splošnem mnogo manjše [3, 9]. Med preprostejšimi tehnikami, ki sodijo v ta sklop, je tudi barvna segmentacija, ki jo lahko uporabimo za detekcijo slikovnih simbolov z dobro definiranimi barvami (npr. prometni znaki) [3, 4, 10].

Veliko uspešnih sistemov za razpoznavanje objektov ne sporoči lokacije objekta na sliki [2]. Lokacija objekta je še posebej zanimiva, ko želimo ugotoviti prostorske relacije med njimi [2, 9]. Zaradi narave delovanja drsečega okna je ta metoda s tega vidika zelo priljubljena [2]. Ker so nekateri slikovni simboli, kot so npr. prometni znaki, značilnih barv in oblik, za njihovo zaznavo uporabljamo kombinirane metode (barvno segmentacijo in zaznavanje iskanih oblik z obrisi) [4, 10]. Ker je barvni model RGB zelo občutljiv na spremembe osvetlitve, nekateri avtorji uporabljajo barvne modele HSI, HSV ali  $L^*a^*b$  [4, 10].

Potencialna zaznana področja s pomočjo izbrane metode strojnega učenja (Support Vector Machine – SVM, Convolutional Neural Networks – CNN, AdaBoost, ...) potrdimo ali zavržemo kot področja z iskanimi slikovnimi simboli [3, 6, 11, 12]. Potrjena področja s slikovnim simboli z izbrano metodo nato razpoznamo [11]. Pri zaznavanju potencialnih področij s slikovnimi simboli imamo poleg pozitivnih učnih primerov, ki predstavljajo enega od želenih zaznavanih simbolov, še večjo množico negativnih učnih primerov, ki ne vsebujejo iskanega slikovnega simbola. Razmerje med pozitivnimi in negativnimi učnimi primer v [11] je 1/10. Če za razpoznavo ne uporabimo CNN, je za učinkovitejšo razpoznavo pomembno, da iz področij s slikovnim simbolom izvlečemo karakteristične značilnosti slikovnega simbola s pomočjo deskriptorjev, med katerimi je še posebej znan deskriptor HOG [11].

Ker so uporabljeni slikovni simboli v našem primeru konkretnih barv, postavljeni na dokaj kontrastno ozadje, smo za zaznavanje predlaganih področij uporabili barvno segmentacijo na osnovi modela HSV v kombinaciji z iskanjem obrisov. S tem smo poenostavili in tudi pohitrili zaznavanje v primerjavi z uporabo metode drsečega okna. Dokončno zaznavo slikovnih

simbolov in razpoznavo smo izvedli s pomočjo klasifikacije predlaganih področij z uporabo klasične polno povezane nevronske mreže [1].

Ker razpoznavanje slikovnih simbolov ni popolno, smo razvili preprost algoritem, ki na osnovi množice odločitev algoritma za razpoznavanje opazovanega prometnega znaka sprejme dokončno odločitev.

V drugem poglavju predstavimo tvorbo učnih primerov. Zaznavanje slikovnih simbolov in njihovo razpoznavanje opišemo v tretjem poglavju. Princip uporabe algoritmov na primeru testnega avtomobila je razložen v četrtem poglavju. V zaključku podamo opis rezultatov, zaznanih težav in možnosti nadaljnjih izboljšav.

## 2 Tvorba učnih primerov

Ker želimo razpoznati šest različnih slikovnih simbolov, potrebujemo šest skupin učnih primerov (slika 7). Da bo razpoznavanje slikovnih simbolov med vožnjo vozila uspešno, potrebuje algoritem strojnega učenja v fazi učenja za vsako skupino veliko količino pozitivnih učnih primerov (npr. veliko različnih slik za prometni znak *ustavi*) kot tudi negativnih učnih primerov (slik, ki ne vsebujejo prometnega znaka).

Tvorba učnih primerov je zamudno in naporno delo. Z uporabo namenskega okolja, ki smo ga razvili v ta namen, smo tvorbo učnih primerov zelo poenostavili in pohitrili. Okolje sestavlja nekaj map, ki poskrbijo za organizacijsko urejenost posnetkov slikovnih simbolov, pozitivnih in negativnih učnih primerov ter skriptov v jeziku Python. Skripti poskrbijo za izdelavo posnetkov slikovnih simbolov, tvorbo raznolikih učnih primerov, ki zajemajo različne zasuke, premike, zrcaljenja, osvetlitve in povečave simbolov, in njihov zapis v datoteko, da jo lahko uporabi algoritem nevronske mreže v fazi učenja (slika 7). Za izvajanje osnovnih operacij nad sliko uporabljamo funkcije OpenCV [5]. Dodali smo tudi skripte za učenje nevronske mreže v okolju MATLAB.



Slika 7: Pozitivni (zgoraj) in negativni (spodaj) učni primeri

Skripti s pomočjo barvne segmentacije in uporabe obrisov iz posnetka s prometnim znakom izrežejo pravokotnik s slikovnim simbolom, ga s pomočjo maske, ki jo ustvarimo z uporabo pragovne operacije nad sivinsko sliko, očistijo vseh sivinskih prvin ter ga s pomočjo inverzne maske kombinirajo z naključnim ozadjem. Slike ozadij posname skript ob samostojni vožnji vozila. Tako nastane en učni primer, ki ga pretvorimo še v sivinsko sliko, zmanjšano na velikost 30x40 pik. Takšen učni primer predstavlja vhodni

vektor (*Feature Vector*) za algoritem strojnega učenja [1]. Tako nastale vhodne vektorje skupaj s pripadajočim izhodom (levo, desno, ustavi, ...) zloži skript v datoteko, da jo lahko uporabimo v fazi strojnega učenja.

Po tem postopku smo izdelali v povprečju 1.460 pozitivnih slikovnih simbolov za vsak prometni znak (skupaj 8.760) in 100.000 negativnih primerov [11], med katerimi je 1/3 takih, ki vsebujejo delček zaznavanega prometnega znaka (slika 7).

### 3 Zaznavanje in razpoznavanje znakov

Za zaznavanje in razpoznavanje prometnih znakov smo uporabili nevronske mreže s tremi sloji [1]. Vsak učni primer, ki ga uporabi nevronska mreža, sestoji iz vhodnega vektorja velikosti števila pik slike, peljanega na vhod nevronske mreže, in pripadajoče izhodne vrednosti (slika predstavlja prometni znak ali ne). Ker ima slika, ki jo peljemo na vhod nevronske mreže, velikost 30 x 40 pik, ima vhodni sloj nevronske mreže 1.200 enot. Za skriti sloj smo izbrali 70 enot. Za učenje obeh nevronske mreže smo uporabili le 80 iteracij z izbrano vrednostjo regulacije L2 enako 0,1 [1].

Z zaznavanjem smo želeli le ugotoviti, ali slika vsebuje prometni znak ali ne, zato ima izhodni sloj nevronske mreže za zaznavanje prometnih znakov le dve enoti. V ta namen smo z uporabo pozitivnih in negativnih učnih primerov z algoritmom vzratnega razširjanja (Back Propagation Algorithm) izračunali uteži nevronske mreže [1].

Ko z zaznavanjem ugotovimo, da je na sliki prometni znak, uporabimo razpoznavanje, ki izmed šestih možnih prometnih znakov izbere najbolj verjetnega. Zatorej ima izhodni sloj nevronske mreže za razpoznavanje šest izhodnih enot. Za razpoznavanje posameznih prometnih znakov izračunamo uteži z uporabo pozitivnih učnih primerov (vsaka slika iz te množice predstavlja enega od šestih prometnih znakov).

Ker se v splošnem slika, ki jo posname kamera avtomobila, precej razlikuje od učnih primerov, s katerimi smo učili nevronske mreže, je potrebno preiskati celotno sliko in na njej poiskati pravokotna področja (npr. okvir znaka stop na sliki 8), ki morda vsebujejo prometne znake.



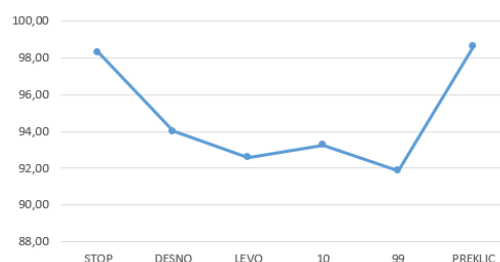
Slika 8: Najdeno področje prometnega znaka ROI (Region Of Interest) znotraj zajete slike

Za iskanje potencialnih področij s prometnimi znaki znotraj slike smo uporabili barvno segmentacijo in obrise na enak način kot pri izdelavi učnih primerov. Zaradi poenostavitve smo vseh šest prometnih znakov, ki jih želimo zaznati, obrobili z rdečo barvo (slika 7) in uporabili segmentacijo po rdeči barvi (barvni model HSV). Če tega ne bi storili, bi morali uporabiti še

segmentacijo po drugih značilnih barvah. Iz seznama najdenih obrisov na sliki smo izbrali le tiste pravokotnike, ki imajo smiselne dimenzije. Na teh mestih smo iz slike izrezali potencialne prometne znake in jih posredovali nevronske mreže, pristojni za razsodbo o tem, ali slika dejansko predstavlja prometni znak. Za njeno učenje smo uporabili 7.008 pozitivnih in 80.000 negativnih učnih primerov, preostale primere (21.752) smo uporabili za testiranje zaznavanja [11]. Delež pravilno zaznanih prometnih znakov iz testne množice je presegel 99%.

Slike, ki so potrjene kot prometni znaki, pošljemo nevronske mreže, pristojni za razsodbo o tem, katerega izmed šestih prometnih znakov slike predstavljajo.

Za učenje nevronske mreže smo od 8.760 izdelanih učnih primerov uporabili le 80 % naključno izbranih, preostanek pa za testiranje razpoznavanja. Odstotek pravilno razpoznanih slikovnih simbolov na učni (95,7 %) in testni množici (95,3 %) je bil približno enak. Razlog za to je verjetno v očitni redundanci ustvarjene množice primerov, iz katere smo še pred učenjem odbrali testne primere. Slika 9 prikazuje odstotke uspešnosti razpoznavanja za posamezne slikovne simbole.



Slika 9: Napovedi testnih prometnih znakov v %

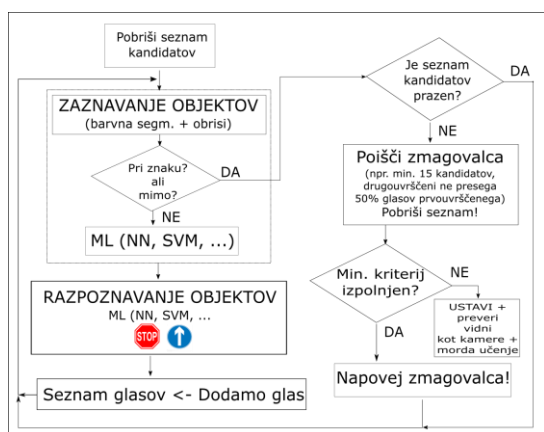
### 4 Uporaba algoritmov na primeru testnega vozila

Ker je vozilo že opremljeno z vso potrebno krmilno-gonilno elektroniko, ki prek mikrokrmilniškega sistema komunicira z računalnikom PC, je nadgradnja vozila le programska [1]. Po uspešnem testiranju delovanja nevronske mreže za zaznavanje in razpoznavanje šestih prometnih znakov smo algoritma vključili v vozilo, ki ima sedaj skupaj z nevronske mreže za samostojno vožnjo po cestišču kar tri nevronske mreže. Vse tri uporabijo isti posnetek kamere velikosti 640 x 480 pik. Preden sliko peljemo na vhod nevronske mreže za samostojno vožnjo vozila, jo pretvorimo v sivinsko in ji nastavimo velikost na 30 x 40 pik [1]. V postopku iskanja področij s potencialnimi prometnimi znaki (barvna segmentacija) uporabimo najprej izvorno velikost slike, iz katere izrežemo najdeni pravokotnik. Njegova ustreznost je izbrana na osnovi dolžine krajše stranice, ki je v našem primeru vsaj 40 pik. Ta pravokotnik sedaj pretvorimo v sivinsko sliko, mu nastavimo velikost 30 x 40 pik in ga peljemo na vhod nevronske mreže za zaznavanje. Zaznani prometni znak razpoznamo in ga shranimo v seznam razpoznanih prometnih znakov (seznam glasov) na sliki 9. Seznam polnimo z novimi razpoznanimi znaki (5 slik v sekundi)



vse do trenutka, ko zaradi približevanja vozila prometnemu znaku preideta širina in dolžina pravokotnika v območje med 70 in 100 pik oziroma vozilo zapelje mimo znaka. Ker razpoznavanje ni idealno, predstavlja seznam razpoznanih prometnih znakov kandidate za izvedbo dokončne odločitve o opazovanem prometnem znaku. Iz seznama ugotovimo število glasov prvo- in drugouvrščenega. Prvouvršeni kandidat je zmagovalec, če ima vsaj dvojno število glasov drugouvrščenega, pri čemer mora biti skupno število kandidatov najmanj 15. Če zmagovalca na ta način ne moremo dobiti, algoritem odločanja vozilo iz varnostnih razlogov ustavi in sporoči vzrok takšne odločitve.

Zaznavanje naslednjega prometnega znaka se začne, ko je daljša stranica zaznanega pravokotnika s potencialnim prometnim znakom manjša od 10 pik. S tem se izognemo ponovni detekciji že najdenega znaka ob vožnji vozila mimo njega.



Slika 9: Odločitev o zmagovalcu med kandidati

Ob razpoznavi prometnih znakov *ustavi, zmanjšaj, povečaj in razveljavi* hitrost, pošlje glavni program mikrokrmilniškemu sistemu preprosto zahtevo po štirih različnih hitrostih (uporaba pulzno širinske modulacije). Uveljavitev razpoznanih zahtev za levo in desno je izvedeno na preprost način. Ob zaznavi zahteve onemogočimo algoritem za samostojno vožnjo vozila po cestišču in za določen čas in kot, ki sta izbrana izkustveno, obrnemo krmilo v ustrezno smer. Nato algoritem za samostojno vožnjo vozila spet omogočimo.

## 5 Zaključek

Zaznavanje in razpoznavanje prometnih znakov smo uspešno implementirali in sprejete odločitve integrirali z mikrokrmilniškim sistemom avtomobila, ki se ustrezno odziva. Z uporabo barvne segmentacije smo poenostavili in pohitrili zaznavanje prometnih znakov v primerjavi z uporabo metode drsečega okna. Preizkusili smo tudi način zbiranja glasov do trenutka, ko vozilo zapelje mimo znaka, vendar se pojavi težava z nadaljnjo vožnjo vozila pri znaku *ustavi*, ker tega potem, ko ga odstranimo s cestišča, več ne vidi.

Zaradi majhnega števila razpoznavanih simbolov in dokaj omejenega laboratorijskega okolja smo z relativno majhnim številom učnih primerov in iteracij pri uporabi

regulacije dosegli dokaj visok odstotek razpoznavne. Že obstoječo lastno implementacijo polno povezane nevronske mreže smo uporabili zgolj zaradi želje po hitrem testiranju vozila. V nadaljevanju raziskav jo že nadomeščamo s konvolucijsko mrežo [6, 12].

Čeprav smo v želji po unikatnih slikovnih simbolih izdelali lastne objekte, je za primerjavo uspešnosti zaznave in razpoznave objektov v prihodnje nujna uporaba javno dostopnih baz.

Za fleksibilnejšo vožnjo vozila v križiščih bi bilo potrebno iz cestišča razpoznati levi in desni rob, s čimer bi dosegli uspešno vožnjo vozila v križiščih tudi takrat, ko je znak postavljen veliko pred njim (vozilo bi sledilo levemu ali desnemu robu).

Kratek video o avtonomni vožnji vozila je prikazan na naslovu: [https://www.youtube.com/watch?v=vzCab\\_3a4iY](https://www.youtube.com/watch?v=vzCab_3a4iY)

## Literatura

- [1] E. Gungl, Z. Brezočnik: Self-driving car based on machine learning. 3rd International Conference New Technologies NT-2016, 13. - 14. May 2016, Mostar, Bosnia and Herzegovina, pp. 370-376.
- [2] C. H. Lampert, M. B. Blaschko, T. Hoffman: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search, 23. - 28. June 2008, Anchorage, USA.
- [3] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, W. Liu: Traffic sign detection and recognition using fully convolutional network guided proposals, Journal Neurocomputing, Vol. 214, 2016, pp. 758-766.
- [4] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, L. D. Stefano: Traffic sign detection via interest region extraction, Pattern Recognition 48 (2015), pp. 1039-1049.
- [5] OpenCV (Open Source Computer Vision Library), <http://opencv.org/>
- [6] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proc. of the IEEE, Vol 86, Issue 11, November 1998.
- [7] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, T. Koehler: A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information, IEEE Intelligent Vehicles Symposium, 6. - 8. June, 2005, Las Vegas, USA, pp. 255-260.
- [8] C. Wojek, G. Dorkó, A. Schuly, B. Schiele, Sliding-Windows for Rapid Object Class Localization: A Parallel Technique, LNCS 5096, pp. 71-81.
- [9] J. Hosang, R. Benenson, P. Dollár, B. Schiele: What Makes for Effective Detection Proposals?, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, No. 4, April 2016.
- [10] K. Brkić, A. Pinz, S. Šegvić: Traffic sign detection as a component of an automated traffic infrastructure inventory system, 2009.
- [11] F. Zaklouta, B. Stanculescu: Real-time traffic sign recognition in three stages, Robotics and Autonomous Systems, Vol. 62, Issue 1, January 2014, pp. 16-24.
- [12] R. Stewart, M. Andriluka, A. Y. Ng: End-to-end people detection in crowded scenes, 2016 IEEE Conference on Computer Vision and Pattern Recognition, Chengdu, China, 2016.