

Implementacija sistema za upozorenje na napuštanje saobraćajne trake

Sofija Filipović, Mihailo Žarković, Dušan Veselinović,
Goran Ferenc, Dragoje Timotijević

Symira d.o.o.

Beograd, Srbija

sofija.filipovic@syrmia.com,

mihailo.zarkovic@syrmia.com,

dusan.veselinovic@syrmia.com, goran.ferenc@syrmia.com,

dragoje.timotijevic@syrmia.com

Danijela Simić

Faculty of Mathematics

University of Belgrade

Belgrade, Serbia

danijela.simic@matf.bg.ac.rs

Sažetak — Pojava autonomnih vozila jedna je od najznačajnijih inovacija u saobraćajnoj industriji. Autonomna vozila imaju za cilj unapređenje efikasnosti, sigurnosti i održivosti saobraćaja putem upotrebe specifičnih senzora. Senzori, kategorizovani u percepciju okoline, praćenje kretanja vozila i navigaciju, predstavljaju ključnu komponentu ovih vozila. Rad se fokusira na implementaciju ADAS algoritma koji doprinosi povećanju bezbednosti u saobraćaju. Specifično, istražuje se primena kamere kao senzora za percepciju okoline, koja se koristi za upozorenje vozača ukoliko napusti svoju saobraćajnu traku. Ova tehnološka aplikacija ima potencijal da unapredi bezbednost u saobraćaju i predstavlja korak ka efikasnijem i sigurnijem transportnom sistemu.

Ključne reči – ADAS – Advanced Driver Assistance Systems; ADAS senzori; Promena perspektive gledanja; Obrada slike; Detekcija linija na transformisanoj slici; Markiranje linija na ulaznoj slici i upozoravanje vozača; Automotive; EuroRC Car; Autonomous Driving; LiDAR; Radar; Kamera

I. UVOD

Saobraćaj je kompleksan sistem koji obuhvata kretanje vozila i pešaka na putevima, a njegovo funkcionisanje zavisi od mnogobrojnih faktora kao što su vremenski uslovi, gustina saobraćaja, stanje puteva i mnogi drugi. Zbog značaja saobraćaja, ceo svet posvećuje pažnju regulisanju uslova i ponašanja u njemu. On predstavlja neizbežni preduslov razmene dobara. Drugim rečima saobraćaj nije izbor pojedinca i bez saobraćaja se ne može zamisliti savremena država. Merilo razvoja bilo koje države može se videti kroz nivo dostignutog razvoja saobraćajnog sistema, pa tako, bez izuzetka, najrazvijenije države imaju najrazvijeniji saobraćajni sistem.

Autonomna vozila za cilj imaju da unaprede efikasnost, sigurnost i održivost saobraćaja pomoću senzora koji se nalaze na njima.

Cilj ovog rada je predstavljanje implementacije jednog od ADAS (Advanced Driver Assistance Systems) [1] algoritama koji doprinosi povećanju bezbednosti u saobraćaju. ADAS sistemi su tehnološki napredni sistemi koji se koriste u vozilima kako bi poboljšali sigurnost i udobnost

vožnje, kao i kako bi pomogli vozačima u raznim situacijama na putu. Ovi sistemi se oslanjaju na različite senzore, kamere, radare, i sofisticirane algoritme kako bi analizirali okolinu vozila i pravili brze odluke u stvarnom vremenu. Ova aplikacija koristi kameru kao senzor za percepciju okoline i upozorava vozača ukoliko je napustio saobraćajnu traku kojom se kretao.

II. ADAS SENZORI ZA PREPOZNAVANJE OKOLINE

ADAS (Advanced Driver Assistance Systems) sistemi su tehnološki napredni sistemi koji se koriste u vozilima kako bi poboljšali sigurnost i udobnost vožnje, kao i kako bi pomogli vozačima u raznim situacijama na putu. Ovi sistemi se oslanjaju na različite senzore, kamere, radare, i sofisticirane algoritme kako bi analizirali okolinu vozila i pravili brze odluke u stvarnom vremenu. Na Sl. 1 [1] prikazana je ilustracija ADAS senzora u modernom autonomnom vozilu [1].



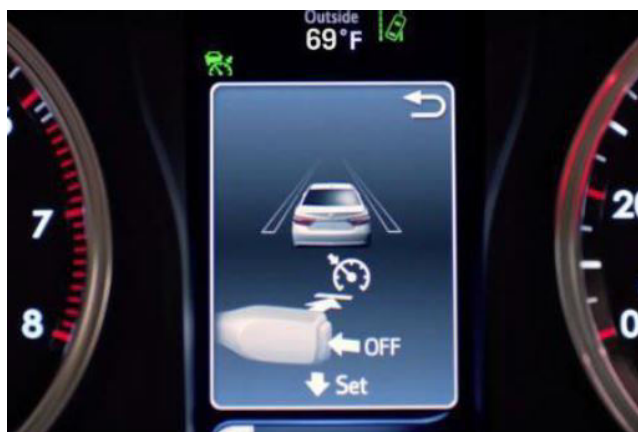
Slika 1. Ilustracija ADAS senzora u modernom autonomnom vozilu

Podaci dobijeni putem radara, LiDAR uređaja, kamera i drugih senzora se objedinjuju kako bi se stvorio jedan model - vozilo sa sistemom fuzije senzora. Takvo vozilo može kombinovati najpreciznije podatke iz svih svojih komponenata (pojedinačnih sistema), zanemarujući one manje bitne podatke. Budući da radari, LiDAR uređaji, kamere i ostali senzori u autonomnom vozilu imaju različite mogućnosti, njihovi podaci se moraju kombinovati u cilju unapređivanja funkcija ADAS sistema. Na primer, kamere su pogodnije za detektovanje saobraćajnih znakova i drugih vozila na putu, radari su pogodni za precizno merenje brzine

kretanja vozila, dok *LiDAR* uređaji mogu pomoći pri određivanju položaja vozila. Kada se podaci uzeti sa svake od ove tri komponente ukombinuju, dobija se jedan precizan i pouzdan ADAS sistem.

III. OPIS SISTEMA

Sistem upozorenja za napuštanje saobraćajne trake (*Lane Departure Warning System*) [2] jedan je od ADAS sistema čija je svrha prvenstveno povećanje bezbednosti u saobraćaju. Sistem koristi različite senzore, obično kamere usmerene prema napred, tako da snimaju put ispred vozila i prate i analiziraju markacije na njemu. Kada registruje da vozilo prelazi preko linija trake bez uključenih pokazivača pravaca, sistem generiše upozorenje. Obično, vozač će primiti vizuelno upozorenje na instrument tabli ili na vetrobranskom staklu. Takođe, može se koristiti i zvučno upozorenje u vidu zvučnih signala ili vibracija volana. Efikasnost ovog sistema zavisi od vremenskih uslova, brzine vozila i uslova na putu. Na Sl. 2 [3] prikazan je primer upozorenja koje sistem prikazuje na instrument tabli.



Slika 2. Ilustracija upozorenja koje sistem prikazuje na instrument tabli

Sistem za upozorenje na napuštanje saobraćajne trake posebno je koristan u situacijama kada vozač postane umoran, nepažljiv ili sklon distrakcijama. Ovaj sistem često radi u tandemu sa drugim ADAS sistemima, kako bi pružio sveobuhvatnu zaštitu i udobnost tokom vožnje.

IV. IMPLEMENTACIJA

Sistem je implementiran u programskom jeziku *Python* [4], uz korišćenje *Matplotlib* [5], *OpenCV* [6] i *NumPy* [7] biblioteka.

Sistem je razvijen pod pretpostavkom da se pokazivači pravca u automobilu tokom vožnje nikad ne aktiviraju, te upozorava vozača pri svakom napuštanju saobraćajne trake.

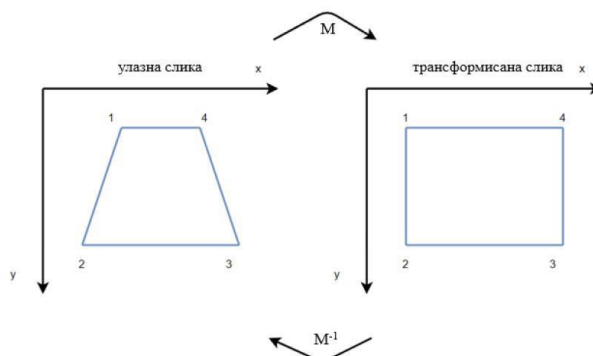
Sistem obrađuje ulazni video okvir po okvir (*frame by frame*). Na svaki okvir primenjuju se sledeća četiri koraka:

- A. Promena perspektive gledanja
- B. Obrada slike
- C. Detekcija linija na transformisanoj slici

D. Markiranje linija na ulaznoj slici i upozoravanje vozača

A. Promena perspektive gledanja

Ulazna slika je jedan okvir iz ulaznog videa, gledano iz perspektive vozača. Rezultat ove faze je transformisana ulazna slika, sa pogledom iz ptičije perspektive. Na Sl. 3 ilustrovana je promena perspektive gledanja, tj. prva faza ovog algoritma.



Slika 3. Ilustracija prvog koraka algoritma: Promena perspektive gledanja

Na ulaznoj slici određuju se četiri tačke koje predstavljaju koordinate gornje leve i desne, kao i donje leve i desne tačke regije od interesa. Kreiraju se dva niza: *src* i *dst*, koji predstavljaju tačke na ulaznoj i transformisanoj slici, respektivno. Primenom različitih računskih operacija na koordinate ulazne slike, dobijaju se dužina i širina transformisane slike, kao i elementi niza *dst*. Upotrebom odgovarajuće funkcije iz *OpenCV* biblioteke, dobija se matrica promene perspektive.

$$M = cv2.getPerspectiveTransform(src, dst);$$

$$M_inv = cv2.getPerspectiveTransform(dst, src);$$

Prvi argument *getPerspectiveTransform()* funkcije predstavlja niz koordinata u ulaznoj slici, dok drugi argument predstavlja niz koordinata u izlaznoj slici. Matrica *M* koristi se u prvoj, dok je matrica *M_inv* potrebna tek u poslednjem koraku.

Primenom *warpPerspective()* funkcije kao rezultat dobija se pogled iz ptičije perspektive na osnovu ulazne slike.

$$warped_img = cv2.warpPerspective(image, M, (w,h), flags=cv2.INTER_LINEAR);$$

Prvi argument *warpPerspective()* funkcije je ulazna slika, drugi argument je matrica promene perspektive, treći argument je par koji predstavlja širinu i visinu izlazne slike. Poslednji argument označava metod interpolacije koji se koristi. Povratna vrednost ove funkcije je slika sa promenjenom perspektivom, koja ujedno predstavlja i krajnji produkt prvog koraka ovog algoritma.

Na Sl. 4 prikazana je ulazna slika, na početku ovog koraka, dok je na Sl. 5 prikazana transformisana slika, koja se dobija na kraju prvog koraka.



Slika 4. Ulazna slika na početku prvog koraka sa obeleženom regijom od interesa



Slika 5. Transformisana slika dobijena na kraju prvog koraka

B. Obrada slike

Ulaz u ovaj korak je slika sa promenjenom perspektivom koja je dobijena na kraju prvog. Nakon ovog koraka, dobija se crno-bela slika, na kojoj će beline biti samo linije saobraćajne trake. Ostali detalji nisu od interesa, pa se uklanjaju primenom određenih funkcija za obradu slike iz *OpenCV* biblioteke. Na Sl. 6 prikazan je deo koda koji predstavlja korišćene funkcije iz *OpenCV* biblioteke.

```
# 2. Image filtering

gray_img = cv2.cvtColor(warped_img, cv2.COLOR_BGR2GRAY)

se = cv2.getStructuringElement(cv2.MORPH_RECT, (5,5))
imgClose = cv2.morphologyEx(gray_img, cv2.MORPH_CLOSE, kernel=se, iterations=3)

th_img_th = cv2.threshold(imgClose, 150, 255, cv2.THRESH_BINARY)

cv2.imshow("filtered",img_th)
```

Slika 6. Deo koda koji prikazuje funkcije upotrebene za uklanjanje detalja koji nisu od interesa

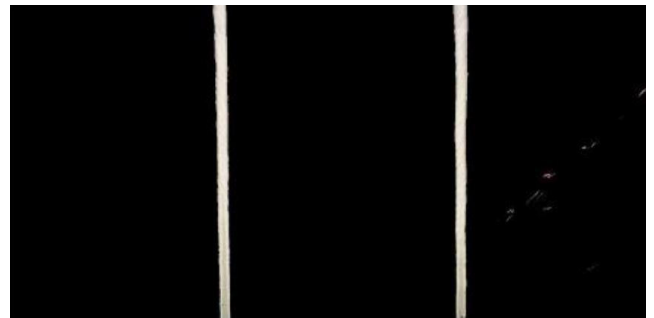
Da bi se uklonili nepotrebni detalji, potrebno je sliku prvo prebaciti iz *BGR* palete boja u crno-belu paletu koristeći *cvtColor()* funkciju iz *OpenCV* biblioteke. Prvi argument ove funkcije je slika čija paleta boja se menja, dok drugi argument funkcije predstavlja oznaku prelaska iz prvobitne u novu paletu. Povratna vrednost ove funkcije je slika u novoj paleti, koja je ujedno i ulazna slika za sledeći korak.

Naredna linija koda kreira strukturni element, koji se koristi u morfološkim operacijama, pozivajući

getStructuringElement(). Prvi argument predstavlja oblik strukturnog elementa, dok drugi predstavlja njegovu veličinu. U ovom slučaju koristi se pravougaoni strukturni element dimenzija 5x5 piksela, koji se kao rezultat ove operacije smešta u promenljivu *se*.

Zatim na red dolazi morfološka operacija zatvaranja na crno-beloj slici *gray_img* koristeći strukturni element *se*. Zatvaranje je postupak koji se upotrebljava za popunjavanje rupa i uklanjanje malih objekata na slici. Parametar *iterations* je postavljen na tri, što znači da će se operacija zatvaranja primeniti tri puta. Rezultat ove operacije je slika smeštena u *imgClose* promenljivu.

Sledeća linija koda predstavlja binarnu segmentaciju slike primenom *threshold()* funkcije na sliku *imgClose*. Drugi i treći argument predstavljaju granične parametre. Svi pikseli na slici koji imaju vrednost manju od 150 postaju 0 (crni), a svi pikseli koji imaju vrednost veću od 150 postaju 255 (beli). Rezultat ovoga je crno-bela slika, gde su belom bojom označene linije saobraćajne trake, dok su svi ostali detalji nevažni, te su uklonjeni sa slike. Rezultat binarne segmentacije smešten u promenljivoj *img_th* prikazan je na Sl. 7. Ta slika ujedno predstavlja i izlaznu sliku iz ovog, kao i ulaznu sliku za naredni korak.



Slika 7. Filtrirana slika dobijena na kraju drugog koraka

C. Detekcija linija na transformisanoj slici

Ukoliko se vrednosti piksela transformisane slike iz drugog koraka sumiraju po vertikalnoj osi, dobija se kriva sa dva izdvojena stupca. *X* koordinate tih stupaca definišu položaj smerova koji odgovaraju longitudinalnim oznakama puta na transformisanoj slici. Na Sl. 8 prikazan je deo koda koji odgovara trećem koraku ovog algoritma.

```
column_sums = np.sum(img_th, axis=0)

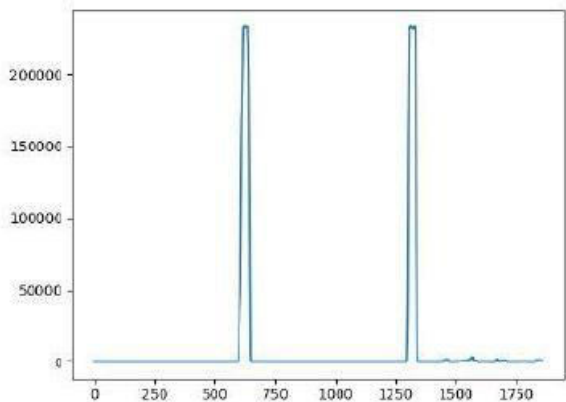
# Prikaz peak-ova za frame - peak oznacava pojavu bele boje

x = range(len(column_sums))
y = column_sums
plt.plot(x, y)
plt.xlabel('X-osa')
plt.ylabel('Y-osa')
plt.title("column_sums graph")
plt.show()
```

Slika 8. Deo koda koji odgovara trećem koraku: Sumiranje vrednosti piksela na ulaznoj slici

Korišćenjem `sum()` funkcije iz `NumPy` paketa programskog jezika `Python` sumiraju se vrednosti piksela sa transformisane slika dobijene na kraju drugog koraka. Rezultat sumiranja smešta se u `column_sums` promenljivu. Na Sl. 9 nalaze se izdvojeni stupci prikazani na histogramu pomoću `Matplotlib` biblioteke.

Nekad se može desiti da se na histogramu iz prethodnog koraka pojavi više od dva stupca. Razlog za to može biti preveliko osvetljenje u nekim trenucima (na primer kada automobil izlazi iz tunela) ili pojava još dve spoljašnje bele linije.



Slika 9. Prikaz histograma na kom se vide izdvojeni stupci koji predstavljaju poziciju belih linija na originilanoj slici

D. Markiranje linija na ulaznoj slici i upozoravanje vozača

Detektovani stupci iz prethodnog koraka predstavljaju izdvojene bele linije na ulaznoj slici. Potrebno je pronaći dva stupca sa maksimalnim vrednostima, koji ispunjavaju određene uslove. Ta dva stupca će predstavljati pozicije leve i desne linije. Oni definišu površinu između dve bele linije na putu koje označavaju saobraćajnu traku u kojoj se vozilo nalazi. Na Sl. 10 prikazan je deo koda koji se koristi pri pronalazanju odgovarajućih belih linija na slici koje odgovaraju granicama saobraćajne trake.

```
global leftLine
global rightLine
carPosition = 1000

maximum_val = column_sums[0]
index = 0
for i in range(1, len(column_sums)):
    if (column_sums[i] > maximum_val and i < carPosition and (carPosition - i) < 100):
        maximum_val = column_sums[i]
        index = i

if (index != 0):
    leftLine = index

maximum_val = column_sums[0]
index = 0
prevIndex = leftLine
for i in range(1, len(column_sums)):
    if ((column_sums[i] > maximum_val and (i != prevIndex) and (i > (leftLine + 100)) and (i < (leftLine + 900)) and (i > carPosition)):
        maximum_val = column_sums[i]
        index = i

if (index != 0):
    rightLine = index
```

Slika 10. Definisane pozicije leve i desne bele linije koje određuju granice saobraćajne trake

Analizom koda sa Sl. 10 dolazimo do sledećih zaključaka: algoritam radi sa slikama rezolucije 1080x1920 piksela. Slika predstavlja ono što kamera postavljena na sredinu vetrobranskog stakla vidi. Imajući to u vidu, centar slike pozicionira se na sredinu slike. Prva `for` petlja služi za

pronalazak indeksa elementa sa najvećom vrednošću u `column_sums` nizu, uz uslov da se taj element na `X` osi nalazi levo od centra automobila.

Druga `for` petlja pronalazi desnu liniju obavljajući sličan posao, uz nekoliko uslova i dodataka. Uslovi koji moraju biti ispunjeni su:

- Indeks pronađenog elementa ne sme biti isti kao indeks elementa koji je već određen za levu liniju u prethodnoj `for` petlji
- Pronađeni element na `X` osi ne sme biti previše udaljen od prethodno određene leve linije
- Pronađeni element na `X` osi mora se nalaziti sa desne strane automobila

Nakon ove dve petlje, precizno su određene linije koje označavaju granice saobraćajne trake u kojoj se auto nalazi. One će poslužiti za određivanje koordinata pravougaonika koji predstavlja površinu kojom automobil treba da se kreće. Koordinate temena pravougaonika smeštaju se u niz `pts`. Na Sl. 11 prikazan je deo koda za određivanje tih koordinata. Na Sl. 12 može se videti deo koda za ispis poruke upozorenja i bojenje pravougaonika određenog elementima niza `pts`.

```
tl = (leftLine, 0) # top left
tr = (rightLine, 0) # top right
bl = (leftLine, warped_img.shape[0]) # bottom left
br = (rightLine, warped_img.shape[0]) # bottom right

pts = np.array([tl, tr, br, bl])
```

Slika 11. Određivanje koordinata temena pravougaonika

```
if(abs(carPosition - leftLine) < 150 or abs(carPosition - rightLine) < 150):
    cv2.putText(image, "Warning!", (850, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0, 0, 255), 2)
    color = (0, 0, 255)
else:
    color = (0, 255, 0)

cv2.fillPoly(warped_img, pts=[pts], color=color)
```

Slika 12. Ispis poruke "Warning!" i bojenje pravougaonika određenog elementima niza `pts`

Poruka o napuštanju saobraćajne trake ispisuje se ukoliko je apsolutna razlika između centra automobila i leve, odnosno desne linije, manja od 150 piksela (to jest ako se automobil "pomerio" ka jednoj o te dve linije). Poruka sa tekstom "Warning!" ispisuje se crvenom bojom. Pravougaonik će u tom slučaju biti obojen istom bojom. Ukoliko je auto u granicama saobraćajne trake, poruka neće biti ispisana, a pravougaonik će biti obojen zelenom bojom. Funkcija `fillPoly()` iz `OpenCV` biblioteke crta na slici koja je zadata kao prvi argument. Drugi argument je niz tačaka koje određuju pravougaonik koji treba obojiti, a treći argument je željena boja.

Funkcija `putText()` iz `OpenCV` biblioteke ispisuje tekst zadat drugim argumentom na slici koja je zadata kao prvi argument. Treći argument je pozicija gde će se tekst ispisati. Četvrti argument je font kojim će tekst biti ispisano. Peti argument predstavlja faktor razmere fonta koji se množi sa osnovnom veličinom fonta. Šesti argument je boja kojom će

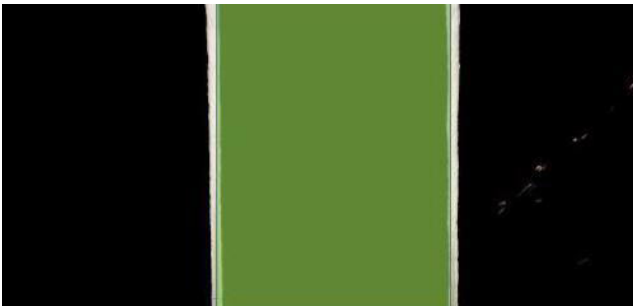
tekst biti ispisan. Sedmi argument predstavlja debljinu linije kojom će tekst biti ispisan.

Sl. 13 prikazuje deo koda koji služi za transformisanje trenutne slike u originalnu perspektivu.

```
final_img = cv2.warpPerspective(warped_img, M_inv, (1920, 1080), flags=cv2.INTER_LINEAR)
bitwise_or = cv2.bitwise_or(image, final_img)
cv2.imshow("final", bitwise_or)
```

Slika 13. Transformisanje u originalnu perspektivu (vozačevu)

Koristeći matricu transformacije M^{-1} , određenu u prvoj fazi, detektovana površina transformiše se u originalni okvir iz videa ponovo koristeći *warpPerspective()* funkciju iz *OpenCV* biblioteke. Rezultat se smešta u *final_img* promenljivu. Finalna slika dobija se kao rezultat *bitwise_or()* funkcije iz *OpenCV* biblioteke. Ona kao parametre prima dve ulazne slike nad kojima primenjuje operaciju bitsko ili. Konačan rezultat smešten je u promenljivu *bitwise_or* i prikazuje se koristeći *imshow()* funkciju iz *OpenCV* biblioteke. Na Sl. 14 i Sl. 15 prikazana je finalna slika pre i posle primene *warpPerspective()* funkcije, respektivno.

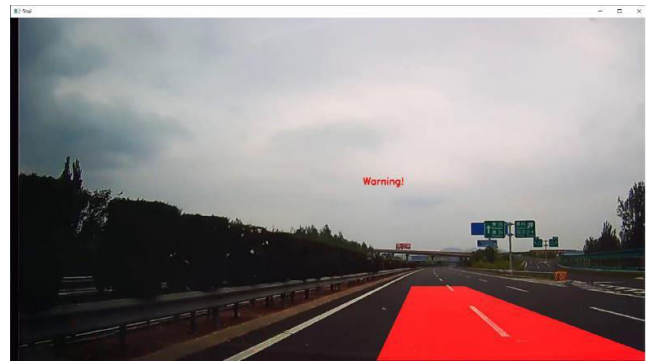


Slika 14. Finalna slika pre primene *warpPerspective()* funkcije



Slika 15. Finalna slika nakon primene *warpPerspective()* funkcije

Na Sl. 16 prikazan je primer ispisa poruke upozorenja i crveno obojen pravougaonik u slučaju da je vozač napustio traku kojom se kretao.



Slika 16. Finalna slika u slučaju da je vozilo napustilo saobraćajnu traku kojom se kretalo

V. EURORC

Ovaj algoritam testiran je i na autonomnom vozilu [8]. Reč je o realističnoj kopiji *Ford Mustang RTR Spec 5* [9]. Ovaj Mustang, izrađen u razmeri 1/10, opremljen je najnovijim *HPI Racing Touring Car* šasijom. Posедуje stalni pogon na sva četiri točka, potpuno zatvoren pogonski sklop, vodootporna 2.4GHz elektronika i radio opremu, kao i nezavisno vešanje. Još jedan deo opreme je i regulator brzine koji može biti korišćen sa modernim litijum-polimer (LiPo) baterijama. Upravljanje se vrši servo motorom koji se kontroliše *PWM* signalom. Drugi *PWM* signal prosleđuje se regulatoru brzine koji upravlja *FIREBOLT 15T* motorom. Na Sl. 17 prikazan je model odabranog autonomnog vozila.



Slika 17. EuroRC Ford Mustang RTR Spec 5

Kao interfejs za kontrolu automobila korišćena je *Clicker 4* ploča od proizvođača MikroElektronika. Uloga ploče je da generiše *PWM* signale za upravljanje servo motorom i regulatorom brzine. Pored toga, ona pruža i četiri *microBus* interfejsa za različite senzore.

Pomenuta ploča dobija komande putem *UART* komunikacije od *Open-Q 865XR* [10] razvojnog kompleta sa *Open-Q SOM*-om, sa *Linux* operativnim sistemom. Svi algoritmi za autonomnu vožnju izvršavaju se na *Linux* operativnom sistemu, generišu se upravljačke komande koje se dalje propagiraju do ploče zadužene za upravljanje samim automobilom. Na Sl. 18 prikazan je *Open-Q 865XR* komplet, *Open-Q SOM*, kao i rezultat testiranja ovog algoritma pomoću pomenutnog kompleta i *EuroRC* vozila.



Slika 18. Open-Q 865XR komplet, Open-Q SOM i rezultat testiranja

VI. REZULTATI

U ovom radu opisana je implementacija sistema za upozorenje na napuštanje saobraćajne trake pri sunčanom i suvom vremenu, prateći ISO 26262 standard [11]. Pri uslovima koji nisu idealni (kiša, magla), ovaj algoritam ne radi najbolje, i mogao bi biti unapređen. Takođe, ovaj rad ne pokriva slučaj kvara neke od kamera u vozilu.

Ograničenje ovog ADAS sistema ogleda se u tome što se sistem za upozorenje na napuštanje saobraćajne trake uglavnom oslanja na vidljive trake na putu. Ukoliko su trake izbledele ili nevidljive, sistem može napraviti grešku. Trake prekrivene snegom ili stare iscrtane trake koje se više ne koriste takođe mogu praviti problem u funkcionisanju sistema. Neki napredniji sistemi za upozorenje na napuštanje saobraćajne trake prate kretanje vozila ispred i na osnovu njega donose odluke, pa im tako gore-pomenute situacije i gužva u saobraćaju ne stvaraju probleme. Postoje i sistemi koji imaju mogućnost automatskog pomeranja volana radi pozicioniranja vozila u centar saobraćajne trake. Algoritam opisan u ovom radu je jednostavniji od nekih naprednijih, ali je brži, jer ne obrađuje kompleksne situacije.

ZAHVALNICA

Autori zahvaljuju Fondu za inovacionu delatnost Republike Srbije koji je delom omogućio izradu ovog rada u okviru projekta broj 50301 pod nazivom "Platform for Remote development of Autonomous Driving algorithms in realistic environment - READ".

LITERATURA

- [1] "What is ADAS?" blog, internet adresa: <https://www.synopsys.com/automotive/what-is-adas.html>, pristupano septembar 2023.
- [2] S. Dragaš, R. Grbić, M. Brisinello and K. Lazić, "Development and Implementation of Lane Departure Warning System on ADAS Alpha Board," 2021 International Symposium ELMAR, Zadar, Croatia, 2021, pp. 53-58.
- [3] "Advanced car safety tech will roll out gradually: HDLI" blog, internet adresa: <https://www.automotive-fleet.com/143632/institute-projects-timeline-for-spread-of-vehicle-safety-tech>, pristupano januar 2024.
- [4] Python dokumentacija, internet adresa: <https://www.python.org/doc/>, pristupano septembar 2023.
- [5] Matplotlib dokumentacija, internet adresa: <https://matplotlib.org/stable/index.html>, pristupano septembar 2023.
- [6] OpenCV dokumentacija, internet adresa: <https://docs.opencv.org/4.x/>, pristupano septembar 2023.
- [7] NumPy dokumentacija, internet adresa: <https://numpy.org/doc/>, pristupano septembar 2023.
- [8] M. Žarković, G. Ferenc and S. Stefanović, "An Autonomous Vehicle Emulator in the Cloud," 2023 22nd International Symposium

INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2023

- [9] FITENTH dokumentacija, internet adresa: <https://fitenth.org/build.html>, pristupano oktobar 2023.
- [10] Lantronix internet adresa: <https://www.lantronix.com/products/open-q-865xr-som-development-kit/>, pristupano maj 2023.
- [11] "ISO26262 and Functional Safety" internet adresa: <https://xed.ch/h/iso26262>, pristupano februar 2024.

ABSTRACT

The emergence of autonomous vehicles represents one of the most significant innovations in the transportation industry. Autonomous vehicles aim to enhance the efficiency, safety, and sustainability of traffic by utilizing specific sensors. These sensors, categorized into environmental perception, vehicle motion tracking, and navigation, constitute a crucial component of these vehicles. The focus of this study is on the implementation of an Advanced Driver Assistance Systems (ADAS) algorithm contributing to increased traffic safety. Specifically, the application of a camera as an environmental perception sensor is explored, serving to alert the driver if they deviate from their traffic lane. This technological application has the potential to improve traffic safety and represents a step towards a more efficient and secure transportation system.

IMPLEMENTATION OF A LANE DEPARTURE WARNING ALGORITHM

Sofija Filipović, Mihailo Žarković, Dušan Veselinović,
Goran Ferenc, Danijela Simić, Dragoje Timotijević