



Примена модерних алгоритама и метода у системима аутономних возила

Application of Modern Algorithms and Methods in Autonomous Vehicle Systems

Дуња Шпановић, Лидија Крстановић, Факултет техничких наука, Нови Сад

Студијски програм – АНИМАЦИЈА У ИНЖЕЊЕРСТВУ

Кратак садржај – Овај рад проучава и примењује савремене алгоритме и методе за перцепцију околине и планирање кретања у системима аутономне возање. Повезујући теоријске основе и имплементацију у програмском језику Пајтон, детаљно су истражене три целине: детекција и праћење саобраћајне траке методама компјутерске визије, имплементација алгоритама за планирање оптималне путање (BFS, DFS и A* алгоритам) и креирање CNN модела за класификацију саобраћајних знакова на скупу података GTSRB.

Кључне речи: аутономна возила, компјутерска визија, детекција саобраћајне траке, планирање путање, класификација саобраћајних знакова

Abstract – This paper consists of three main parts: lane detection and tracking using computer vision methods, implementation of path-planning algorithms (Breadth-First Search, Depth-First Search, and A*), and development of a CNN model for traffic sign classification using the GTSRB dataset.

Keywords: autonomous vehicles, computer vision, lane detection, path planning, traffic sign recognition

НАПОМЕНА: Овај рад проистекао је из мастер рада чија менторка је била др Лидија Крстановић, ванр. проф.

1. УВОД

Аутономна возила представљају један од највећих технолошких напретака у области транспорта. Захваљујући способности да функционишу без директног учешћа људи, значајно утичу на безбедност саобраћаја, као и на друштвене и економске аспекте живота. Развој ове технологије подстакнут је потребом да се смањи број саобраћајних незгода, од којих је више од 90% последица људских грешака [1]. Захваљујући могућности комуникације са другим возилима и саобраћајном инфраструктуром у реалном времену, аутономна возила доприносе бољој координацији и оптималнијем протоку саобраћаја. Поред тога, ефикаснија возања коју омогућавају може смањити емисију штетних гасова, допринети очувању

животне средине и олакшати кретање особама са ограниченом покретљивошћу.

У пракси, кључни изазови обухватају детекцију и праћење саобраћајне траке у променљивим условима, препознавање саобраћајне сигнализације, уочавање препрека, процену стања возача, као и помоћне функције попут асистенције при паркирању. Њихово решавање ослања се на алате компјутерске визије и вештачке интелигенције, од класичних метода обраде слике до дубоког учења, у комбинацији са напредним сензорима, рачунарима и алгоритмима који омогућавају возилу да тумачи околину, доноси одлуке и управља у складу са тренутним условима на путу.

2. ТЕОРИЈСКЕ ОСНОВЕ

Теоријске основе дају сажет преглед кључних концепата и технологија система за самосталну возању: историјски и технички развој, улогу компјутерске визије у перцепцији окружења и преглед главних софтверских алата и библиотека, са посебним нагласком на програмски језик Пајтон.

2.1. Системи аутономних возила

Аутономна возила интегришу модуле за перцепцију, планирање и контролу како би се самостално кретала. Њихов развој почиње од да Винчијевих самоходних кола из 16. века и првих радио-контролисаних аутомобила из 20. века, наставља се кроз академске пројекте и DARPA такмичења, све до данашњих индустријских решења [2].

Степен самосталности описује се SAE класификацијом од нултог до петог нивоа, при чему се нижи нивои већ примењују, а виши се тестирају у ограниченим условима [3]. У Србији је 2023. уведен правилник за испитивање трећег и четвртог нивоа, што подстиче даљи развој и њихову већу примену.

Перцепција се ослања на камере, лидар, радар и ултразвучне сензоре, док GPS и IMU обезбеђују позиционирање. Фузија ових података, уз вештачку интелигенцију и V2X комуникацију, омогућава доношење одлука у реалном времену [4].

2.2. Компјутерска визија

Компјутерска визија из података са камера добија поздане информације за доношење одлука и, у споју са алгоритмима дубоког учења, омогућава перцепцију окружења у реалном времену. Због ограничења класичних метода, предност су добиле конволуционе

неуронске мреже које представљају основу савремених система за детекцију, праћење, сегментацију и класификацију објеката [5]. Њихова архитектура (конволуциони слој, груписање, активациони слој, потпуно повезани слој) издваја карактеристике и омогућава класификацију. Имплементација се ослања на *OpenCV*, једну од кључних библиотека компјутерске визије, док се за обуку модела примењују *TensorFlow* и *Keras*.

2.3. Програмски језик Пајтон

Програмски језик Пајтон је вишенаменски програмски језик који је стекао велику популарност због своје једноставне синтаксе и прилагодљивости. Као интерпретабилни језик, омогућава извршавање кода без претходног компајлирања. Он подржава различите стилове програмирања, попут процедуралног, објектно оријентисаног и функционалног, што га чини изузетно свестраним алатом.

У овом раду програмски језик Пајтон служи као основа развоја у *Visual Studio Code*-у, док његов екосистем покрива цео ток обраде података: *NumPy* и *SciPy* за нумеричке прорачуне, *Pandas* за табеларне формате и *Matplotlib* и *Seaborn* за визуализацију.

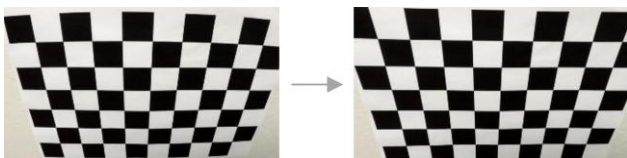
3. ИМПЛЕМЕНТАЦИЈА ФУНКЦИОНАЛНОСТИ

Практичан део овог рада заснива се на имплементацији кључних функционалности које чине основу система за аутономну вожњу: детекција саобраћајне траке, алгоритми за планирање оптималне путање и препознавање саобраћајних знакова.

3.1. Детекција саобраћајне траке

Детекција саобраћајне траке представља функционалност која се у потпуности ослања на податке прикупљене путем камере. Применом алгоритма компјутерске визије идентификују се линије које са леве и десне стране дефинишу границе саобраћајне траке. На основу њих обезбеђује се праћење и одржавање одговарајућег положаја возила у оквиру предвиђене путање. Процес детекције траке у овом раду заснива се на редоследу корака описаних у [6], где свака фаза припрема податке за наредну повећавајући тачност резултата.

Први корак подразумева калибрацију камере која повезује 3D тачке сцене са њиховим 2D пројекцијама и одређује интринзичне и екстринзичне параметре и коефицијенте дисторзије. Урађена је на 20 снимака шаховске табле 10×7 аквизираних помоћу камере која је на возилу. На основу добијених параметара је отклоњена дисторзија као што је приказано на слици 1.



Слика 1. Отклањање дисторзије

Следећи корак је примена перспективне трансформације, тако да линије на путу које су на необрађеној слици конвергентне постану паралелне.

На овај начин се добија поглед на пут из птичије перспективе, што даље омогућава прецизнију детекцију траке. Перспективна трансформација се заснива на израчунавању матрице, која се добија на основу четири улазне и четири циљне тачке. Улазне тачке су одређене положајем леве и десне линије траке које се у даљини спајају, док циљне тачке формирају правоугаоник у који се оригинални трапезни облик пута пресликава.

Поступак обухвата конвертовање у сиву слику, затим отклањање шума *Gaussian* филтром и детекцију ивица *Canny* детектором. На издвојеним ивицама се дефинише регион од интереса који маскира слику тако да се задржи само доњи део слике у облику трапеза. Након тога, линијски сегменти се детектују *Hough*-овом трансформацијом, а затим се групишу по нагибу на леве и десне линије. Групе линија се усредњавају тако да се добије по једна репрезентативна лева и десна линија. Из њихових горњих и доњих тачака формира се улазни трапез чије се пресликавање врши помоћу матрице хомографије у правоугаоник, односно у саобраћајну траку из птичије перспективе. На слици 2 је приказан резултат перспективне трансформације након које су линије на путу постале паралелне.



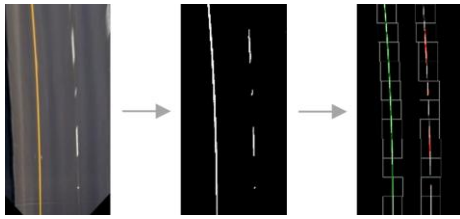
Слика 2. Перспективна трансформација

Претходни кораци се извршавају пре почетка обраде података у реалном времену. Једном добијени параметри камере и матрица трансформације чувају се у засебном документу и користе током целокупне анализе видеа. Први корак који се спроводи на сваком фрејму је бинаризација, чији је циљ да се пиксели линија представе белом бојом, а остатак слике црном.

Да би се добили стабилни резултати при различитим условима осветљења издвајају се два канала: *R* из *RGB* (за беле) и *b* из *CIELAB* (за жуте линије). Оба канала се побољшавају *Top-hat* морфолошком операцијом, након чега се врши локална бинаризација тако што се централни пиксел пореди са околином у 4 смера. Пиксел постаје бео ако је довољно светлији у хоризонталном или вертикалном правцу, чиме се постиже отпорност на шум и промене осветљења. На крају се маске, настале издвајањем и побољшавањем канала, спајају у јединствену бинарну слику.

Завршна фаза детекције траке почиње хистограмом доње трећине фрејма, из кога се одређују почетне *x* координате леве и десне линије. На основу тих полазних тачака примењује се метода клизајућих прозора која дуж висине слике прикупља пикселе трака и затим их апроксимира полиномима другог реда. Из добијених полинома израчунава се радијус закривљености пута и одређује смер кривине поређењем усредњених нагиба при дну и на врху фрејма. Попречно одступање возила од центра траке

добија се као разлика између центра траке и оптичке осе камере. Да би резултати били стабилнији, континуирано се примењују медијан и усредњавање последњих 9, односно 7 израчунатих вредности. Резултати бинаризације и праћења линија клизајућим прозорима приказани су на слици 3.



Слика 3. Бинаризација и примена клизајућих прозора

На крају се детектована саобраћајна трака визуализује полигоном враћеним у оригиналну перспективу, уз испис радијуса, смера кривине и одступања возила од средине траке, као што је приказано на слици 4.

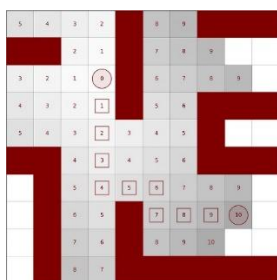


Слика 4. Детектована саобраћајна трака

3.2. Алгоритми за планирање оптималне путање

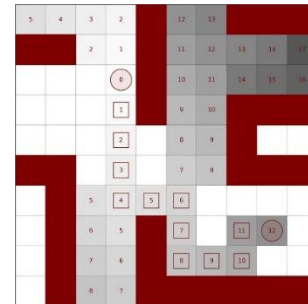
Алгоритми за планирање оптималне путање омогућавају возилу да безбедно стигне од старта до циља уз избегавање препрека. У раду су имплементирани и поређени: претрага у ширину - *BFS*, претрага у дубину - *DFS* и *A** алгоритам [7]. Упоредна анализа је вршена на примеру мреже са 10×10 ћелија са истим распоредом препрека, старта и циља и могућим кретањем горе, доле, лево, десно, приликом чега су прикупљени подаци о дужини и облику путање и броју посећених ћелија.

Претрага у ширину креће од старта и помоћу реда (*FIFO*) обрађује све на удаљености од једног корака, затим два, итд. *BFS* шири фронт у свим правцима и посећује велики број ћелија, што га чини неефикасним. Главна предност *BFS*-а је једноставност имплементације и гаранција проналаска најкраће путање. Приликом претраге у ширину на датој мрежи број истражених ћелија је 57, док дужина најкраћег пута износи 10 корака. Облик путање приказан је на слици 5.



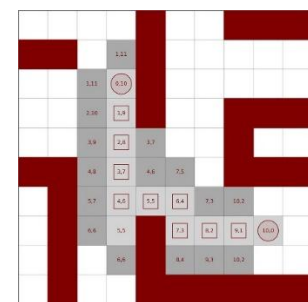
Слика 5. Претрага у ширину

Претрага у дубину истражује граф тако што иде што даље од старта дуж једног гранања, па се затим враћа уназад. Чворови се обрађују помоћу стека (*LIFO*), па алгоритам иде у дубину уместо слојевитог ширења које је карактеристично за *BFS*. Захтева мање меморије, али не гарантује најкраћу путању и у великој мери зависи од редоследа обиласка суседа. Уколико се деси да је прва грана која се посећује управо она која води до циља, резултат се постиже брзо. У супротном, алгоритам улази у слепе улице и враћа се, па пут није нужно оптималан, нити је број посећених ћелија мањи него код *BFS*-а. На слици 6 приказана је мрежа на којој је истражено укупно 44 ћелије, док пронађена путања има дужину од 12 корака.



Слика 6. Претрага у дубину

*A** алгоритам за планирање путање користи приоритетни ред у којем даје предност чворовима са мањом вредношћу $f = g + h$, где је g стварни трошак пута од старта, а h хеуристичка процена трошка до циља (нпр. Менхетн удаљеност). Захваљујући хеуристици, претрага се усмерава ка циљу и обилази знатно мање чворова него неинформисане методе. Када је хеуристика допуштима (не прецењује трошак), *A** је потпун и оптималан, тј. гарантује проналазак најкраће путање. На слици 7 је приказан резултат претраге по мрежи са Менхетн хеуристиком. Добијена је иста дужина пута као код *BFS*-а (10), али уз мање посећених чворова (12) и ужи фронт претраге (16). Полазећи од овог примера мреже са могућим кретањем у четири смера, *A** се издваја као најпогоднији избор за проналажење оптималне путање.



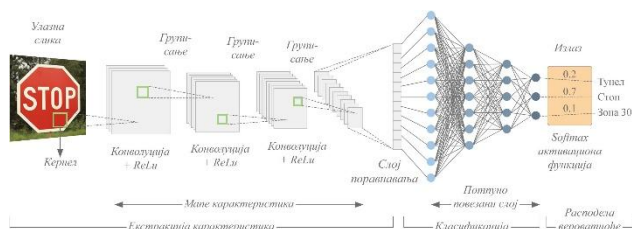
Слика 7. *A** алгоритам

3.3. Класификација саобраћајних знакова

Да би кретање било безбедно и у складу са прописима, перцептивни модул возила треба да поуздано препознаје саобраћајне знакове у реалном времену. У последњем делу рада креиран је *CNN* класификатор над скупом података *GTSRB* [8], који садржи преко 50 000 слика немачких саобраћајних знакова снимљених из различитих углова и при променљивим

условима осветљења. Имплементација је изведена у *Google Colab*-у уз *T4 GPU*.

Слике организоване у 43 класе су учитане, уједначене на димензије $30 \times 30 \times 3$ и подељене у односу 80:20 на тренинг и тест скуп. Ознаке су преведене у *one-hot* запис, а оштећени узорци прескочени уз испис поруке о грешци. Након припреме података, модел се креира употребом *CNN* архитектуре, чији је шематски приказ дат на слици 8.



Слика 8. Архитектура *CNN* модела

После компајлирања, модел се евалуира на тест скупу и приказују се графици тачности и губитка током епоха. Као метрика су примењени матрица конфузије и извештај о класификацији по класама (*Precision*, *Recall*, *F1*), чиме је омогућен увид у класе знакова које се најчешће међусобно замењују. За практичну проверу појединачних примера насумично се бира тест слика која се скалира на димензију 30×30 и пропушта кроз модел, након чега се добија предикција са исписаним називом класе. На слици 9 су приказани примери практичне провере модела.



Слика 9. Примери предикције саобраћајног знака

4. ЗАКЉУЧАК

У раду су истражене три целине система за аутономну вожњу: детекција саобраћајне траке, планирање путање (*BFS*, *DFS* и *A** алгоритам) и класификација знакова *CNN*-ом на *GTSRB* скупу. Резултати показују стабилну детекцију траке при умереним променама осветљења, већу ефикасност *A** у односу на *BFS* и *DFS*, и високу тачност *CNN* класификатора саобраћајних знакова.

Могућа унапређења обухватају развој модула за детекцију траке у ноћним условима и неповољним временским приликама, као и прилагођавање планирања путање динамичним препрекама. Поред тога, увођење корака локализације пре класификације повећало би робусност система и омогућило препознавање знакова на различитим позицијама у окружењу у реалном времену. За примену у Србији, потребно је прилагодити систем класификације прикупљањем локалних података и усклађивањем са важећим прописима и службеним писмом.

5. ЛИТЕРАТУРА

DOI: <https://doi.org/10.24867/34SA01Spanovic>
 DOI: <https://doi.org/10.24867/34SA01Spanovic>
 DOI: <https://doi.org/10.24867/34SA01Spanovic>
 DOI: <https://doi.org/10.24867/34SA01Spanovic>
 DOI: <https://doi.org/10.24867/34SA01Spanovic>
 DOI: <https://doi.org/10.24867/34SA01Spanovic>

[7] L. Anany, Introduction to the Design and Analysis of Algorithms (3rd Edition), Boston: Pearson, 2011.

[8] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing и C. Igel, „Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark,“ у International Conference on Neural Networks, 2013.

[9] „Traffic Signs Classification App,“ GitHub, 8.9.2020. URL: https://github.com/Spidy20/Traffic_Signs_WebApp (приступљено: 3.4.2025.)

Кратка биографија:



Дуња Шпановић рођена је у Сомбору 9.4.2000. Од 2019. студира на Факултету техничких наука у Новом Саду, смер Анимација у инжењерству, где је 2023. уписала мастер академске студије у истој области. Мастер рад је одбранила 2025. године. Контакт: spanovic2000@gmail.com