

UNIVERZITET U BEOGRADU  
ELEKTROTEHNIČKI FAKULTET

Jelena M. Kocić

**AUTONOMNO ODRŽANJE VOZILA U  
KOLOVOZNOJ TRACI ANALIZOM  
INFORMACIJA SA VIZUELNIH SENZORA  
KORIŠĆENJEM NEURALNE MREŽE**

doktorska disertacija

Beograd, 2020



UNIVERSITY OF BELGRADE  
SCHOOL OF ELECTRICAL ENGINEERING

Jelena M. Kocić

**AUTONOMOUS VEHICLE LANE KEEPING BY  
ANALYZING INFORMATION FROM VISUAL  
SENSORS USING A NEURAL NETWORK**

Doctoral Dissertation

Belgrade, 2020



Mentor:

dr Nenad Jovičić, vanredni profesor  
Univerzitet u Beogradu – Elektrotehnički fakultet

Članovi komisije:

dr Vujo Drndarević, redovni profesor u penziji  
Univerzitet u Beogradu – Elektrotehnički fakultet

dr Dragana Perić, viši naučni saradnik  
Institut Vlatacom, Beograd

dr Marko Barjaktarović, docent  
Univerzitet u Beogradu – Elektrotehnički fakultet

dr Goran Kvašček, vanredni profesor  
Univerzitet u Beogradu – Elektrotehnički fakultet

Datum odbrane: \_\_\_\_\_



# ZAHVALNICA

Zahvaljujem se svom mentoru, prof. dr Nenadu Jovičiću, za ohrabriranje u vezi izbora teme doktorske disertacije, za konstantnu podršku tokom istraživanja, teorijskog i eksperimentalnog rada. Posebno se zahvaljujem na insistiranju da se teorijska saznanja i zaključci osim u simuliranim uslovima verifikuju i u realnim uslovima realizacijom novog sistema za autonomnu vožnju, čime je osnovna ideja disertacije dobila i praktičnu realizaciju.

Zahvaljujem se prof. dr Vuji Drndareviću, svom mentoru tokom doktorskih studija, pre njegovog odlaska u penziju, za svu podršku, savete i vođenje kroz doktorske studije. Profesor me je naučio da prepoznam kvalitet i da mu težim, o važnosti razmišljanja na sistemskom nivou kada je reč o idejama, ali i o važnosti detalja kada je reč o implementaciji.

Zahvaljujem se Sumitu Šarmi, Kevinu Votsonu i Jariju Honkanenu za pruženu priliku da znanje i veštine koje sam stekla tokom svog istraživačkog rada na doktorskim studijama primenim u praksi, da projektujem nova rešenja mašinskog učenja i kompjuterske vizije i implementiram ih na visokotehnološki razvijenim realnim uređajima.

Zahvaljujem se dr Alekseju Makarovu za podršku i ohrabrenje da se paralelno sa građenjem inženjerske karijere bavim i naučno-istraživačkim radom. Zahvaljujem za pomoć u ranoj fazi moje karijere i doktorskih studija, da inovativna rešenja do kojih sam došla u praksi realizujući realne uređaje i sisteme objasnim naučnim metodama i publikujem na stručno-naučnim konferencijama.

Zahvaljujem se dr Miroslavu Laziću koji mi je prvi pružio šansu da uđem u svet profesionalnog razvoja elektronike, koji mi je pokazao da industrija i nauka idu jedno uz drugo i da je poželjno imati saznanja o oba sveta kako bi se stvarali krajnje inovativni proizvodi. Dr Lazić me je ličnim primerom inspirisao i podstakao da upišem doktorske studije.

Zahvaljujem se Zoranu Vasiljeviću Gukiju, svom razrednom starešini i profesoru informatike na tome što je prepoznao u meni talenat za informatiku i podstakao me da se razvijam u tom smeru. Profesor Guki je tokom srednjoškolskog školovanja usadio jake osnove programiranja i kritičkog razmišljanja ne samo meni već mnogim generacijama, stvorivši tako idealnu bazu za dalju nadogradnju. Takođe, zahvalna sam na životnim lekcijama koje je ovaj veliki čovek delio svojim učenicima stvarajući od dece ljude sa integritetom.

Zahvaljujem se svojim roditeljima Milki i Miodragu na безусловnoj podršci, poverenju i ohrabriranju da istrajem u donetim odlukama, svojim sestrama Jovani i Josipi na bezgraničnoj ljubavi i razumevanju. Konačno, zahvaljujem suprugu Mirku i svojoj deci Urošu i Mariji, koja su rođena i rasla tokom trajanja doktorskih studija, na podršci, strpljenju i ljubavi.





**Naslov doktorske disertacije:** Autonomno održanje vozila u kolovoznoj traci analizom informacija sa vizuelnih senzora korišćenjem neuralne mreže

**Sažetak:** Cilj disertacije je ostvarivanje autonomnog održanja vozila u kolovoznoj traci analizom informacija sa vizuelnih senzora korišćenjem projektovane duboke neuralne mreže, eng. *deep neural network* (DNN). DNN za učenje od-kraja-do-kraja na ulaz dovodi sliku sa kamere montirane na vozilu, a izlaz iz DNN je informacija o uglu okretanja upravljača vozila. Ova tehnika se još naziva i kloniranje ponašanja vozača, eng. *behavior cloning*.

Polazna hipoteza je da je moguće ostvariti autonomnu vožnju korišćenjem duboke neuralne mreže za učenje od-kraja-do-kraja koja je računarski manje zahtevna od do sada postojećih rešenja, pri čemu korišćenjem modela nove mreže, performanse autonomne vožnje ne degradiraju značajno. Osnovna prednost novog rešenja je omogućavanje implementacije projektovanog rešenja na autonomno vozilo sa ograničenim hardverskim performansama u smislu računarske snage i memorijskog kapaciteta. Razvijena je nova arhitektura DNN za učenje od-kraja-do-kraja za autonomnu vožnju koja je nazvana J-Net. U poređenju sa drugim poznatim modelima, PilotNet i AlexNet, J-Net model ima najmanji broj trenabilnih parametara, najmanji broj operacija nad čvorovima neuralne mreže i istrenirana J-Net mreža zauzima najmanje memorijskog prostora.

Verifikacija autonomne vožnje ostvarena je u simuliranim uslovima, korišćenjem simulatora autonomne vožnje otvorenog koda, i u realnim uslovima. Za verifikaciju u realnim uslovima, projektovan je sistem za autonomnu vožnju u laboratoriji za elektroniku Elektrotehničkog fakulteta Univerziteta u Beogradu. Verifikacije u simuliranim i u realnim uslovima pokazale su da je korišćenjem J-Net modela duboke neuralne mreže za učenje od-kraja-do-kraja moguće ostvariti uspešno održanje vozila u kolovoznoj traci analizom informacija sa vizuelnih senzora.

**Ključne reči:** autonomna vožnja, duboka neuralna mreža (DNN), duboko učenje, kamera, mašinsko učenje, robo-vozilo, simulator, sistem za autonomnu vožnju, učenje od-kraja-do-kraja.

**Naučna oblast:** Elektrotehnika i računarstvo

**Uža naučna oblast:** Elektronika

**UDK broj:** 621.3

**Doctorial Dissertation Title:** Autonomous vehicle lane keeping by analyzing information from visual sensors using a neural network

**Abstract:** The aim of the dissertation is to achieve autonomous lane keeping by analyzing information from visual sensors using the designed deep neural network (DNN). DNN for end-to-end learning has a camera image as an input and the steering angle of the vehicle as an output. This technique is also called behavior cloning.

The starting hypothesis was that it is possible to achieve autonomous driving using a deep neural network for end-to-end learning that is less computationally demanding than the existing solutions, whereby using a new network model, autonomous driving performance does not degrade significantly. The main advantage of the new solution is enabling the implementation of the designed solution on an autonomous vehicle with limited hardware performance in terms of computing power and memory capacity. A new DNN for end-to-end learning architecture for autonomous driving has been developed, it is called J-Net. Compared to other known models, PilotNet and AlexNet, the J-Net model has the least number of trainable parameters, the least number of operations of neural network nodes, and the trained J-Net network occupies the least memory space.

Verification of autonomous driving achieved in simulated conditions, using an open-source simulator for autonomous driving, and in real-world conditions. For the verification in real-world conditions, an autonomous driving system was designed and implemented in the Laboratory of Electronics at the School of Electrical Engineering, University of Belgrade. Verifications in both simulated and real-world conditions have shown that it is possible to achieve successful autonomous lane-keeping by analyzing information from visual sensors using the J-Net DNN model.

**Key words:** autonomous driving, deep neural network (DNN), deep learning, camera, machine learning, robo-vehicle, simulator, autonomous driving system, end-to-end learning.

**Scientific field:** Electrical Engineering and Computer Science

**Scientific subfield:** Electronics

**UDK number:** 621.3

# Sadržaj

1. Uvod .....	1
2. Autonomna vozila.....	4
2.1.  Podsistemi autonomnog vozila.....	5
2.2.  Senzori – kamera, radar i lidar .....	6
2.2.1.  Kamera.....	6
2.2.2.  Radar .....	7
2.2.3.  Lidar .....	8
2.3.  Podsystem za percepciju .....	9
2.4.  Planiranje i upravljanje.....	10
3. Duboko učenje .....	11
3.1.  Veštačka inteligencija, mašinsko učenje i duboko učenje .....	11
3.2.  Višeslojne neuralne mreže.....	14
3.2.1.  Perceptron .....	14
3.2.2.  Aktivaciona funkcija .....	15
3.2.3.  Funkcija gubitka.....	17
3.2.4.  Arhitektura višeslojnih neuralnih mreža .....	19
3.2.5.  Treniranje neuralne mreže koristeći propagaciju unazad .....	20
3.3.  Konvolucione neuralne mreže.....	21
3.4.  Evaluacija modela mašinskog učenja.....	22
3.5.  Razvoj dubokog učenja .....	23
4.  Primena tehnika dubokog učenja u autonomnim vozilima.....	25
4.1.  Fuzija informacija sa senzora .....	25
4.1.1.  3D detekcija objekata.....	26
4.1.2.  Detekcija i praćenje objekata u pokretu .....	26
4.2.  Lokalizacija i mapiranje .....	27
4.3.  Semantička segmentacija .....	28

4.4.	Učenje od-kraja-do-kraja ili kloniranje ponašanja vozača .....	29
4.4.1.	Primena modifikovane PilotNet arhitekture za učenje od-kraja-do-kraja .....	30
4.4.2.	Modifikacija i implementacija AlexNet arhitekture za učenje-od-kraja-do-kraja .....	32
5.	Opis problema i metodologije treniranja, verifikacije i evaluacije duboke neuralne mreže .....	34
5.1.	Postavka problema .....	34
5.2.	Opis simulatora autonomne vožnje .....	35
5.3.	Kreiranje baze podataka .....	37
5.3.1.	Uvećanje broja podataka .....	38
5.3.2.	Baza podataka .....	38
5.3.3.	Preprocesiranje podataka .....	39
5.4.	Treniranje duboke neuralne mreže za učenje od-kraja-do-kraja za autonomno održanje vozila u kolovoznoj traci .....	39
5.5.	Metode merenja odstupanja vozila od sredine kolovozne trake.....	41
6.	Projektovanje i implementacija nove arhitekture modela neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju .....	43
6.1.	Definisanje metodologije kreiranja arhitekture duboke neuralne mreže.....	43
6.2.	Analiza uticaja dubine neuralne mreže na performanse funkcionalnog modela.....	45
6.3.	Analiza uticaja dubine konvolucionog sloja i veličine kernela na performanse funkcionalnog modela .....	48
6.4.	Konačna arhitektura duboke neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju <sup>50</sup>	
7.	Verifikacija autonomne vožnje u simuliranim uslovima.....	52
7.1.	Treniranje duboke neuralne.....	52
7.2.	Uporedna analiza J-Net i drugih modela neuralnih mreža za autonomnu vožnju.....	54
7.2.1.	Računarska kompleksnost.....	54
7.2.2.	Evaluacija performansi tokom autonomne vožnje u simuliranim uslovima.....	56
7.3.	Evaluacija modela J-Net na novom delu kolovozne trake .....	60
8.	Verifikacija autonomne vožnje u realnim uslovima .....	61

8.1.	Opis sistema za autonomnu vožnju .....	61
8.2.	Samovozeća robotska platforma .....	63
8.3.	Zadata putanja za verifikaciju autonomne vožnje .....	64
8.4.	Sistem za automatsku akviziciju i laberiranje slika – kreiranje baze podataka.....	65
8.5.	Treniranje duboke neuralne mreže .....	69
8.6.	Autonomna vožnja u realnom vremenu .....	70
8.7.	Rezultati verifikacije autonomne vožnje u realnim uslovima na zadatoj kolovoznoj traci.	75
8.7.1.	Računarska kompleksnost modela dubokih neuralnih mreža .....	75
8.7.2.	Kvalitativni rezultati autonomne vožnje .....	75
8.7.3.	Kvantitativni rezultati autonomne vožnje .....	78
8.8.	Evaluacija modela J-Net na novoj kolovoznoj traci.....	79
9.	Zaključak .....	83
	Literatura.....	85
	Biografija autora .....	93

## SPISAK SLIKA

Slika 1.1 Autonomna vozila kroz istoriju: (a) Univerzitet Karnegi Melon, 1894; (b) Univerzitet Stenford na DARPA Grand Challenge takmičenju, 2005; (c) Daimler i Boš, pilot robo-taksi program, 2018.....	1
Slika 2.1 Nivoi autonomije vozila [14].....	4
Slika 2.2 Blok dijagram autonomnog vozila.....	5
Slika 2.3 Svetlosni spektar.....	6
Slika 2.4 Kamere u autonomnim vozilima. Levo – Flir Flea3, desno gore – ADAS kamera Mobileye i ZF S-Cam4, desno dole – stereo kamera Slightech MYNT EYE.....	7
Slika 2.5 Izgled radara.....	7
Slika 2.6 Lidar sa rotirajućom senzorskom glavom, Velodyne HDL-64E, HDL-32E i VLP-16.....	8
Slika 2.7 Lidar sa MEMS vibrirajućim mikroogledalima [36].....	9
Slika 3.1 Veštačka inteligencija, mašinsko učenje i duboko učenje.....	11
Slika 3.2 Klasično učenje i mašinsko učenje.....	11
Slika 3.3 Reprzentacije jednog sloja duboke neuralne mreže nastaju kombinovanjem više jednostavnijih reprzentacija iz prethodnog sloja duboke neuralne mreže.....	13
Slika 3.4 Duboka neuralna mreža.....	13
Slika 3.5 Tok primene metoda dubokog učenja: najpre se dizajnira duboka konvolucionna neuralna mreža, nakon čega se korišćenjem odgovarajuće baze podataka mreža trenira. Konačno, istreniran model neuralne mreže se u realnom vremenu koristi za predikciju definisani.....	14
Slika 3.6 Perceptron – jednoslojna neuralna mreža.....	15
Slika 3.7 Aktivacione funkcije.....	17
Slika 3.8 Izvodi aktivacionih funkcija.....	18
Slika 3.9 Arhitektura jedne troslojne neuralne mreže; (a) skalarna notacija; (b) vektorska notacija.....	20
Slika 3.10 Primer 2D konvolucije, ulazni podatak je (3×4), a kernel (2×2).....	22
Slika 4.1 Blok dijagram 3D detekcije objekata, slika modifikovana na osnovu [82].....	26
Slika 4.2 Sistem za percepciju korišćenjem više senzora, slika modifikovana na osnovu [83].....	27
Slika 4.3 Ilustracija DeepVO metode za vizuelnu odometriju od-kraja-do-kraja korišćenjem rekurentnih konvolucionih neuralnih mreža, slika modifikovana na osnovu [85].....	27
Slika 4.4 Mapiranje zauzetosti mreže, slika preuzeta iz [86].....	28
Slika 4.5 Potpuna konvolucionna neuralna mreža za semantičku segmentaciju na nivou piksela, slika modifikovana na osnovu [87].....	29
Slika 4.6 Blok dijagram sistema za autonomnu vožnju korišćenjem učenja od-kraja-do-kraja.....	29

Slika 4.7 Originalna PilotNet (naziva se i NVIDIA CNN) arhitektura, slika preuzeta iz [91].	31
Slika 4.8 Arihitektura reimplementirane PilotNet mreže.	31
Slika 4.9 Originalna AlexNet arhitektura, slika preuzeta iz [63].	32
Slika 4.10 Arihitektura modifikovane i reimplementirane AlexNet mreže.	33
Slika 5.1 Autonomna vožnja u realnom vremenu: na osnovu slike sa kamere koja je ulaz neuralne mreže, određuje se ugao okretanja upravljača.	34
Slika 5.2 Blok dijagram postupka za ostvarivanje autonomne vožnje.	35
Slika 5.3 Simulator autonomne vožnje, pogled iz vazduha na scenu u Unity okruženju [89].	36
Slika 5.4 Vozilo u okruženju simulatora na reprezentativnoj kolovoznoj traci, sa tri kamere postavljene na vrhu vozila. U donjem desnom uglu slike je prikaz slike koju vidi centralna kamera na vozilu.	36
Slika 5.5 Ulazni podaci snimljeni na različitim delovima kolovozne trake korišćenjem centralne kamere postavljene na vozilu.	37
Slika 5.6 Primer akvizicije podataka prikupljenih istovremeno u jednom frejmu sa tri kamere montirane na vozilu: (a) leva; (b) centralna; (c) desna kamera.	37
Slika 5.7 Ulazna slika nakon odsecanja.	39
Slika 5.8 Dijagram toka treniranja duboke neuralne mreže za učenje od-kraja-do-kraja.	40
Slika 5.9 Snimak jednog trenutka uspešne vožnje u simulatoru.	41
Slika 5.10 Merenje odstupanja vozila od centra kolovozne trake; (a) postoji odstupanje, vozilo se kreće bliže levoj ivici puta, (b) primer dijagrama relativog odstupanja kretanja vozila od centralne linije na čitavoj putanji.	41
Slika 5.11 Primeri neuspešne vožnje – skretanje vozila sa kolovozne trake.	42
Slika 6.1 Arhitektura prve eksperimentalne duboke neuralne mreže.	43
Slika 6.2 Grafički prikaz odstupanja automobila od sredine kolovozne merenjem srednje kvadratne i integralne apsolutne greške; (a) MSE i (b) IAE.	50
Slika 6.3 Arhitektura predložene duboke mreže za učenje od-kraja-do-kraja, J-Net.	51
Slika 7.1 Gubitak modela tokom treniranja i validacije za J-Net, PilotNet i AlexNet: (a) treniranje u 30 epoha; (b) treniranje u 6 epoha.	53
Slika 7.2 Poređenje arhitektura dubokih neuralnih mreža koje primenjuju učenje od-kraja-do-kraja za autonomnu vožnju:(a) AlexNet; (b) PilotNet; (c) J-Net.	55
Slika 7.3 Relativno odstupanje od centra trajektorije za jedan pun krug autonomne vožnje. Četiri glavne karakteristike puta: (a) uglavnom prav put oivičen bankinom, (b) krivine oivičene crvenobelim trakama, (c) most, (d) krivina oivičena prašinom (desno).	58
Slika 7.4 Histogram odstupanja od centra trajektorije za jedan pun krug autonomne vožnje.	59

Slika 8.1 Dva režima rada sistema za autonomnu vožnju.....	61
Slika 8.2 Osnovne komponente eksperimentalne postavke sistema za autonomnu vožnju: (1) kamera za lokalizaciju, (2) server računar, (3) robo-vozilo na definisanoj kolovoznoj traci i (4) računar na kome se vrši zaključivanje u realnom vremenu. ....	62
Slika 8.3 Samovozeća robotska platforma, robo-vozilo. ....	64
Slika 8.4 Zadana putanja za verifikaciju autonomne vožnje u realnim uslovima. ....	64
Slika 8.5 Izbor koordinata putanje za eksperimentalnu verifikaciju autonomne vožnje. ....	65
Slika 8.6 (a) Pogled na definisanu kolovoznu traku sa kamere za video nadzor; (b) Mapa koordinata kojom se definiše automatsko kretanje vozila. ....	65
Slika 8.7 Blok dijagram sistema za kreiranje baze podataka (režim automatskog kretanja). ....	66
Slika 8.8 Algoritam za lokalizaciju i automatsko kretanje po zadatoj mapi. ....	67
Slika 8.9 Primeri akviziranih slika za autonomnu vožnju u realnim uslovima.....	68
Slika 8.10 Blok dijagram treniranja neuralne mreže za autonomnu vožnju u realnim uslovima. ....	69
Slika 8.11 Treniranje duboke neuralne mreže, snimak ekrana. ....	69
Slika 8.12 Dijagrami gubitka pri treniranju neuralnih mreža u 30 epoha. ....	70
Slika 8.13 Blok dijagram sistema za autonomnu vožnju robo-vozila u realnom vremenu. ....	71
Slika 8.14 Dijagram toka podataka i naredbi tokom autonomne vožnje u realnom vremenu. ....	72
Slika 8.15 Algoritam zaključivanja na računaru za mašinsko učenje. ....	72
Slika 8.16 Algoritam na server računaru kojim se zadaje pomeraj robo-vozilu. ....	73
Slika 8.17 Faza zaključivanja - određivanje ugla na osnovu slike u realnom vremenu. ....	74
Slika 8.18 Jedan frejm video snimka autonomne vožnje po zadatoj putanji korišćenjem J-Net neuralne mreže [112]. ....	76
Slika 8.19 Trajektorija kretanja robo-vozila u realnom vremenu korišćenjem dubokih neuralnih mreža: plava – centar zadate trajektorije; zelena – PilotNet; crvena – J-Net; svetlo plava – AlexNet. ....	77
Slika 8.20 Nova putanja za autonomnu vožnju u realnim uslovima.....	79
Slika 8.21 Postavka sistema sa novom putanjom za autonomnu vožnju u realnim uslovima, [113]. ....	80
Slika 8.22 Izbor koordinata putanje za eksperimentalnu verifikaciju autonomne vožnje. ....	81
Slika 8.23 Trajektorija kretanja robo-vozila u realnom vremenu korišćenjem J-Net modela na novoj kolovoznoj traci.....	82



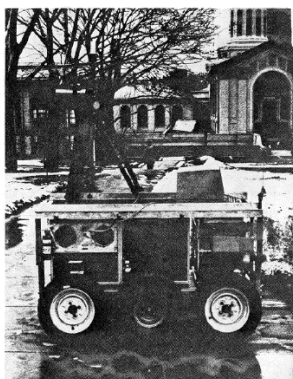
## SPISAK TABELA

<b>Tabela 5.1</b> Baza podataka parova slika-ugao za autonomnu vožnju u simulatoru. ....	38
<b>Tabela 6.1</b> Eksperimentalne arhitekture dubokih neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju testirane na simulatoru autonomne vožnje. ....	47
<b>Tabela 7.1</b> Kvantitativni opis mrežnih arhitektura: J-Net, PilotNet i AlexNet. ....	55
<b>Tabela 7.2</b> Veličine istreniranih modela: J-Net, PilotNet, i AlexNet. ....	56
<b>Tabela 7.3</b> Evaluacija autonomne vožnje korišćenjem modela AlexNet, PilotNet i J-Net. ....	57
<b>Tabela 7.4</b> Kvantitativna evaluacija autonomne vožnje korišćenje modela AlexNet, PilotNet i J-Net na platformi sa visokim performansama za simulator autonomne vožnje. ....	59
<b>Tabela 7.5</b> Evaluacija autonomne vožnje korišćenjem J-Net modela na novom delu kolovozne trake ..... .....	60
<b>Tabela 8.1</b> Baza podataka za treniranje duboke neuralne mreže za autonomnu vožnju u realnim uslovima. ....	68
<b>Tabela 8.2</b> Karakteristike treniranih modela u realnim uslovima, J-Net, PilotNet, and AlexNet. ....	75
<b>Tabela 8.3</b> Kvalitativna evaluacija autonomne vožnje korišćenjem modela AlexNet, PilotNet i J-Net. ....	77
<b>Tabela 8.4</b> Rezultati kvantitativne evaluacije autonomne vožnje korišćenjem modela AlexNet, PilotNet i J-Net u sistemu autonomne vožnje u realnim uslovima. ....	78
<b>Tabela 8.5</b> Rezultati kvantitativne evaluacije autonomne vožnje na novom modelu putanje ..... .....	81



# 1. Uvod

Prvi uspešni pokušaji razvoja autonomnih vozila dogodili su se 50-tih godina prošlog veka, a prvi prototipovi potpuno autonomnih vozila su razvijeni 1985-1986. godine od strane istraživača sa Univerziteta Karnegi Melon [1,2], slika 1.1(a) i 1987. godine od strane istraživača sa Univerziteta Bundeswehr Minhen [3]. Početkom ovog veka, praktično je demonstrirana uspešna autonomna vožnja u realnim uslovima, i to autonomna vožnja u ruralnim uslovima 2005-2006. godine [4,5], slika 1.1(b), i u urbanim uslovima 2007. godine [6]. Međutim, početkom druge decenije ovog veka, 2010-tih godina, dolazi do značajnog napretka u razvoju algoritama i tehnika veštačke inteligencije, odnosno mašinskog učenja, razvoja novih softverskih biblioteka za mašinsko učenje, napretka u razvoju hardvera koji omogućava paralelno izvršavanje operacija, kao što su na primer grafičke procesorske jedinice, i do generisanja velikog broja obimnih anotiranih baza podataka. Napredak u razvoju veštačke inteligencije je izazvao revoluciju u tehnici, u oblastima kao što su robotika, komunikacije, infrastruktura (takozvani pametni gradovi), potrošački proizvodi, društvene mreže, ali je uticao i na ostale oblasti ljudskog društva i humanih nauka, ostavivši traga na oblasti kao što su medicina, agronomija, ekonomija, i dr.



(a)



(b)



(c)

*Slika 1.1 Autonomna vozila kroz istoriju: (a) Univerzitet Karnegi Melon, 1894; (b) Univerzitet Stenford na DARPA Grand Challenge takmičenju, 2005; (c) Daimler i Boš, pilot robo-taksi program, 2018.*

Razvoj novih tehnika mašinskog učenja i njihova primena je od izuzetnog značaja za napredak u razvoju autonomnih vozila. Tehnike mašinskog učenja i dubokog učenja predstavljaju centralni deo inteligencije autonomnih vozila, preuzevši na sebe zadatke poput opažanja okruženja, detekcije i praćenja objekata, prepoznavanja prepreka, saobraćajnih znakova i svetlosne signalizacije, istovremene lokalizacije vozila i mapiranja okruženja, planiranja putanje i učenja autonomne vožnje od-kraja-do-kraja gde se neobrađeni signal sa senzora dovodi u algoritam dubokog učenja, a kao izlaz se dobijaju komande za upravljanje vozilom – tehnika koja se još naziva kloniranje ponašanja vozača. Cilj ovog doktorata je razvoj nove arhitekture i modela duboke neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju, čijom primenom se ostvaruje autonomno održavanje vozila u kolovoznoj traci analizom isključivo informacija sa vizuelnih senzora, odnosno kamera.

Osnovni zadatak algoritama mašinskog učenja je rešavanje konkretnog problema sa što je moguće većom tačnošću, što obično vodi do razvoja veoma kompleksnih i dubokih neuralnih mreža koje su računarski izuzetno zahtevne [7,8]. Kompleksni modeli dubokih neuralnih mreža se najčešće koriste za rešavanje problema iz domena kompjuterske vizije. Na primer, neke od dobro poznatih višeslojnih mrežnih arhitektura su VGGNet (16 do 19 slojeva) [9], GoogLeNet (22 sloja) [10], ResNet (152 sloja) [11], i slične arhitekture dubokih neuralnih mreža bazirane na ovim modelima. Konačno, napredak u razvoju konvolucionih neuralnih mreža i aplikacija, i eksperimentisanje sa kompleksnijim

arhitekturama su posledica dva faktora, konstruisanja i dostupnost velikih baza podataka i poboljšane računarske efikasnosti hardvera na kome se izvršavaju algoritmi mašinskog učenja.

Primena algoritama dubokog učenja i korišćenje istreniranih modela u realnom vremenu na namenskim ugradnim platformama sa ograničenom računarskom snagom i memorijskim prostorom zahteva drugačiji pristup [12]. Izvršavanje treniranja modela duboke neuralne mreže, kao i izvršavanje faze zaključivanja, odnosno predikcije gde se koristi istrenirani model neuralne mreže direktno zavisi od kapaciteta modela duboke neuralne mreže koja se koristi, odnosno od broja čvorova neuralne mreže, težinskih faktora dodeljenih svakom čvoru – parametrima modela, i broja operacija nad pojedinačnim čvorovima neuralne mreže. Stoga je za projektovanje duboke neuralne mreže kojom se rešava konkretan zadatak a koja se koristi na ugradnoj namenskoj platformi svrsishodno primeniti jednostavniji model nad kojim se vrši manji broj operacija za jednu predikciju, odnosno odluku u realnom vremenu [13]. Međutim, implementaciju jednostavnijeg modela mašinskog učenja koji postiže zadovoljavajuću tačnost je veoma zahtevno ostvariti, pogotovo kada su aplikacije vezane za kompjutersku viziju i podrazumevaju sliku visoke rezolucije kao ulazni podatak. Redukcija dubine neuralne mreže i broja čvorova obično dovodi do degradacije u tačnosti modela koji je projektovan. Stoga, tokom razvoja arhitekture duboke neuralne mreže za kompjutersku viziju pogodna za primenu na namenskim platformama, neophodno je pronaći rešenje koje će zadovoljiti oba kriterijuma: željenu tačnost i mogućnost primene modela na hardverskoj platformi sa ograničenim mogućnostima, kapacitetom procesora i veličinom raspoložive memorije. Iz ovog razloga, pri projektovanju arhitekture duboke neuralne mreže za autonomno održanje vozila u kolovoznoj traci, težilo se razvoju duboke neuralne mreže čijom će se upotrebom postizati zadovoljavajuća tačnost, a koja će istovremeno moći da radi u realnom vremenu unutar ograničenja platforme na kojoj se izvršava u smislu računarske snage i memorijskog prostora.

Kako bi se kreirala arhitektura neuralne mreže namenjena primeni na namenskoj platformi koja radi u realnom vremenu, pristup koji je korišćen u ovoj disertaciji nije redukcija broja parametara neke poznate arhitekture neuralne mreže koja se može koristiti za autonomnu vožnju, već razvoj nove arhitekture, principom kreiranja sloj po sloj neuralne mreže, do postizanja zadovoljavajućeg rešenja. Generalni princip projektovanja odgovarajućeg modela mrežne arhitekture koji je usvojen u ovom radu je da se počne sa relativno malim brojem slojeva i parametara, a da se potom povećava veličina slojeva ili da se dodaju novi slojevi do postizanja željene tačnosti. Pod novim slojevima dubokih neuralnih mreža, osim konvolucije i primene kernela različitih dimenzija nad konkretnom konvolucijom, podrazumevaju se i operacije kao što su udruživanje maksimizacijom (eng. *MaxPooling*), ravnajući sloj ili potpuno povezani sloj, kao i operacije za redukciju broja čvorova, poput tehnike izbacivanja (eng. *Dropout*). U radu je primenjena metodologija hijerarhijskog projektovanja duboke neuralne mreže kroz procese selekcije osnovnih elemenata arhitekture i specifičnih parametara implementacije u cilju zadovoljavanja unapred zadatih kriterijuma i performansi. Najpre su analizirane globalne karakteristike arhitekture neuralne mreže koje dominantno čine broj i tipovi slojeva, što ujedno predstavlja i dubinu neuralne mreže. Nakon toga se na sledećem hijerarhijskom nivou analiziraju uticaji veličine samih slojeva, konkretno broj mapa odlika svakog konvolucionog sloja. Na kraju se u najnižem hijerarhijskom nivou razmatra i uticaj veličine konvolucionih kernela.

Doprinosi predloženog pristupa su sledeći:

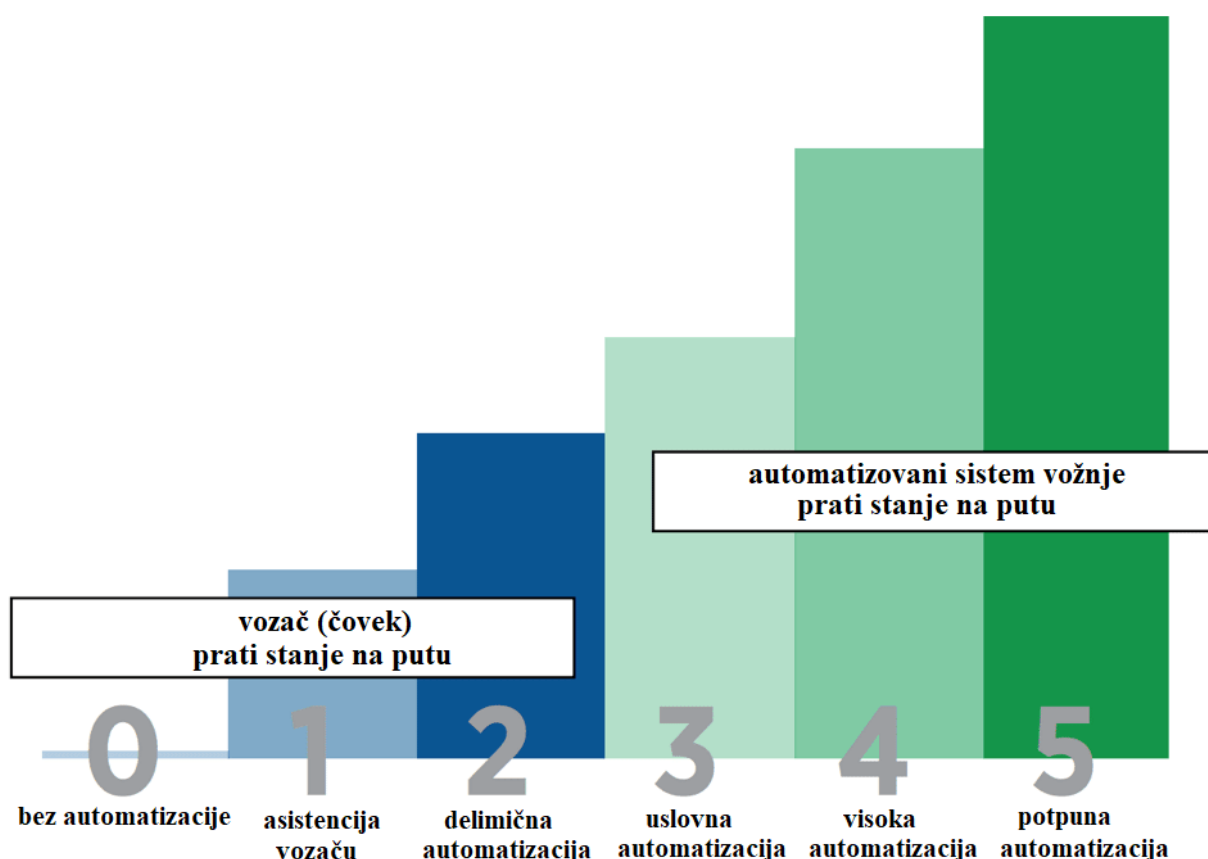
1. Razvijena je i implementirana nova originalna arhitektura duboke neuralne mreže za učenje od-kraja-do-kraja, koja u odnosu na referentne algoritme za upravljanje obezbeđuje značajno smanjenje potrebe za računarskom snagom i kompjuterskim resursima bez značajne degradacije performansi vožnje.

2. Opisana je originalna metodologija hijerarhijskog projektovanja duboke neuralne mreže kroz procese selekcije osnovnih elemenata arhitekture i specifičnih parametara implementacije u cilju zadovoljavanja unapred zadatih kriterijuma i performansi.
3. Razvijen je originalni laboratorijski sistem za eksperimentalnu verifikaciju algoritma upravljanja autonomnim vozilom.
4. Korišćenjem referentnog simulatora i razvijenog laboratorijskog sistema izvršena je verifikacija i poređenje razvijenog algoritma sa aktuelnim i referentnim algoritmima koji se koriste za upravljanje autonomnim vozilima i dokazana je validnost osnovne hipoteze istraživanja da je moguće projektovati novi algoritam za učenje od-kraja-do-kraja za ostvarivanje autonomne vožnje zasnovan na upotrebi duboke neuralne mreže, koji u odnosu na referentne algoritme za upravljanje ima značajno manje potrebe za računarskom snagom i kompjuterskim resursima.

U drugom poglavlju ovog rada je predstavljen sistem autonomnog vozila i autonomne vožnje, dat je opis senzora autonomnih vozila, podsistema za percepciju, planiranje i kontrolu nad vozilom. U trećem poglavlju dat je pregled osnovnih pojmova iz domena veštačke inteligencije, mašinskog učenja i dubokog učenja sa akcentom na duboke konvolucione neuralne mreže. U četvrtom poglavlju dati su pregled i analiza algoritama dubokog učenja primenjivih u sistemu za percepciju autonomnih vozila. U petom poglavlju opisani su problem i metodologija treniranja, verifikacije i evaluacije duboke neuralne mreže. Centralni deo disertacije, šesto poglavlje opisuje postupak projektovanja i implementacije novog modela arhitekture duboke neuralne mreže za učenje od-kraja-do-kraja koja je nazvana J-Net. U sedmom poglavlju je predstavljen postupak verifikacije korišćenja J-Net modela za autonomno održanje vozila u kolovoznoj traci u simuliranim uslovima. Rezultati korišćenja J-Net modela su upoređeni sa još dva modela, PilotNet i AlexNet, koji su implementirani s ciljem da se uporede performanse modela tokom autonomne vožnje. U osmom poglavlju prikazani su rezultati verifikacije u realnim uslovima emulacijom sistema za autonomnu vožnju, kao i opis realizovanog sistema za autonomnu vožnju. Zaključak je dat u poslednjem devetom poglavlju.

## 2. Autonomna vozila

Pod autonomnim vozilom se smatra vozilo koje delimično ili potpuno samostalno obavlja vožnju sa ili bez prisustva čoveka na mestu vozača. Postoji pet nivoa autonomije vozila koji se definišu standardom Udruženja inženjera automobilske industrije (eng. *Society of Automotive Engineers*), SAE J3016\_201806 [14], slika 2.1. Motorna vozila bez automatizacije definišu se kao nulti nivo autonomije. Nivoi autonomije jedan i dva predstavljaju automatsku asistenciju vozaču i delimičnu automatizaciju, respektivno. Najviši nivo autonomije vozila koja se danas nalaze u komercijalnoj upotrebi su vozila sa nivoom autonomije tri, i on se naziva uslovna automatizacija. Kod uslovne automatizacije autonomno vozilo može preuzeti vožnju u određenim scenarijima, npr. vožnja autoputom ili vožnja u uslovima usporenog saobraćaja. Autonomna vozila sa nivoom autonomije četiri predstavljaju visoko automatizovana vozila, vozila koja su u mogućnosti da ostvare potpunu autonomnu vožnju. Za vozila sa autonomijom nivoa četiri nije neophodno prisustvo vozača, ali da bi se uspostavila vožnja, neophodno je da budu ispunjeni svi preduslovi. Primeri primene vozila sa nivoom autonomije četiri su lokalna taksi vozila bez vozača koja se kreću u određenom poznatom okruženju, npr. naselju ili kampusu [15]. Pod autonomnim vozilima nivoa pet se podrazumevaju potpuno autonomna vozila (eng. *full autonomous vehicle*) u kojima nije potrebno prisustvo vozača i vozilo je osposobljeno da vozi u svim uslovima. Razvoj autonomnih vozila nivoa četiri i pet je u toku i, mada su pojedine kompanije i istraživački centri dobili dozvolu za testiranje ovakvih vozila na konkretnim lokacijama u više različitih država, potpuno autonomno vozilo još uvek nije razvijeno u smislu komercijalne proizvodnje i upotrebe. Predikcije su da će se komercijalna potpuno autonomna vozila pojaviti na tržištu u narednih deset godina [16], [17].



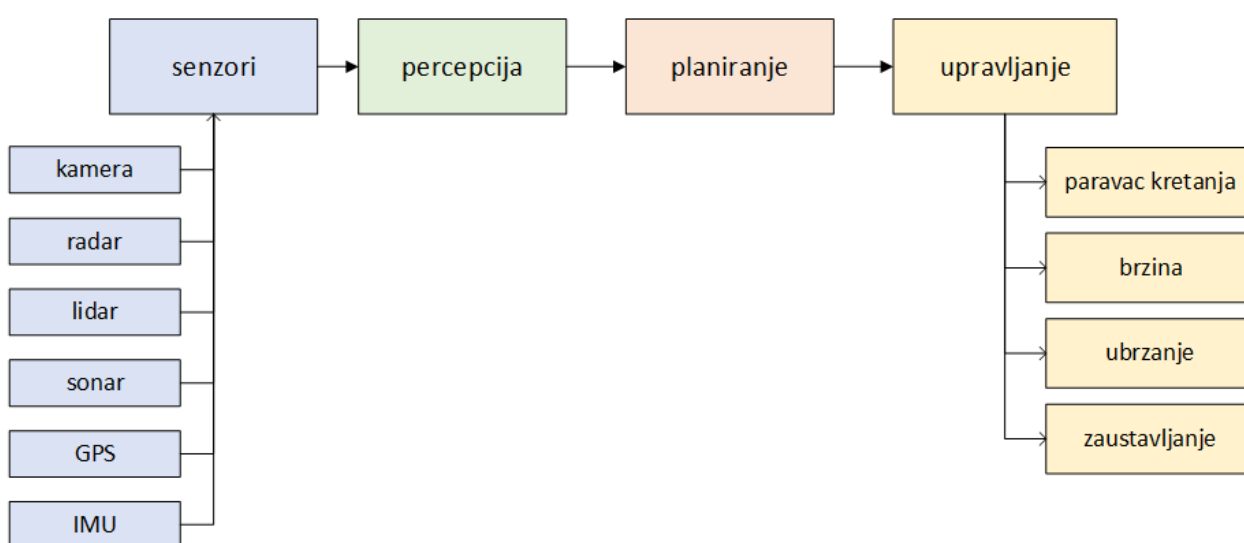
Slika 2.1 Nivoi autonomije vozila [14].

Postoje dva pristupa u procesu razvoja potpuno autonomnih vozila, postizanja autonomije nivoa pet. U prvom pristupu proizvođači automobila i istraživačko-razvojni centri postepeno nadograđuju autonomiju vozila idući od nivoa autonomije 1 ka ciljanoj autonomiji nivoa 5. Većina proizvođača automobila se opredelila za ovaj pristup, na primer svi modeli automobilskog brenda Tesla koji trenutno imaju autonomiju nivoa dva [18], kao i pojedini modeli automobilskih brendova kao što su BMW (model iNext) [19], Audi (model A8L) [20] koji su proizvedeni sa nivoom autonomije 3 se nalaze u komercijalnoj upotrebi. Drugi pristup podrazumeva direktan razvoj autonomnih vozila nivoa 4 i 5, odnosno potpuno autonomnih vozila. Ovakav pristup je pre karakterističan za istraživačko-razvojne centre visokotehnoloških kompanija koje su fokusirane na razvoj softverskih rešenja. Primeri kompanija koje trenutno testiraju svoje prototipove potpuno autonomnih vozila, nivo autonomije pet, su Waymo LCC (ogranak kompanija Alphabet Inc. i Google, Inc.) i Uber Technologies, Inc. [21].

U paraleli sa razvojem klasičnih autonomnih vozila za prevoz putnika i dobara, kao što su automobili i kamioni, razvijaju se i autonomna vozila za prevoz dobara koja mogu biti poput automobila ili kombija, ali i robo-vozila kojima je moguće ostvariti dostavu paketa, ili autonomno operisanje unutar određenog okruženja, [22]-[24].

## 2.1. Podsistemi autonomnog vozila

Autonomno vozilo čine četiri osnovna bloka: senzori, podsistem za percepciju, podsistem za planiranje putanje i podsistem za upravljanje vozilom. Blok dijagram sistema autonomnog vozila prikazan je na slici 2.2. Autonomno vozilo opaža svet oko sebe korišćenjem na desetine senzora različitih tipova koji su montirani na vozilu ili se nalaze unutar vozila: kamera, radar, lidar, sonar, globalni pozicioni sistem (GPS), inercijalna merna jedinica, eng. *inertial measurement unit* (IMU). Informacije prikupljene pomoću senzora se procesiraju u podsistemu za percepciju, čije komponente kombinuju informacije sa senzora i nakon procesiranja ih pretvaraju u informacije sa značenjem. Podsistem za planiranje koristi izlaz iz bloka za percepciju i vrši planiranje ponašanja vozila uključujući planiranje putanje, odnosno daljeg kretanja vozila. Podsistem za upravljanje osigurava da vozilo prati putanju i ponašanje predviđeno sistemom za planiranje i šalje upravljačke komande vozilu. Ovo podrazumeva upravljanje sistemom za kočenje, upravljačem, radom motora, odnosno pravcem kretanja, brzinom, ubrzanjem i zaustavljanjem vozila.



Slika 2.2 Blok dijagram autonomnog vozila.

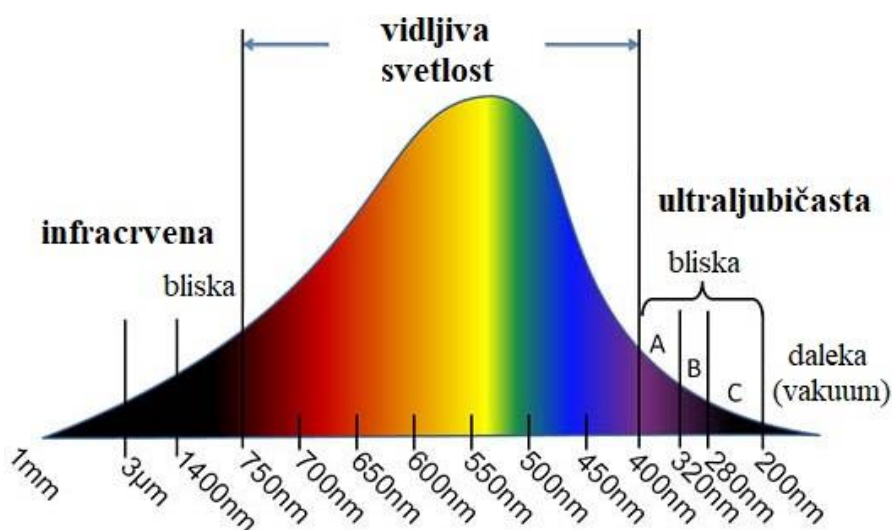
## 2.2. Senzori – kamera, radar i lidar

Jedna od osnovnih motivacija za razvoj autonomnih vozila je smanjenje broja saobraćajnih nesreća. Osnovna ideja je bila da se razvije autonomno vozilo koje će biti pouzdanije od čoveka na mestu vozača s obzirom na to da je u većini slučajeva ljudski faktor glavni uzrok saobraćajnih nesreća. Kako bi se ostvario ovaj zadatak, autonomna vozila treba ne samo da imitiraju ljudsko ponašanje, odnosno vozača, već da ostvare bolji učinak i vožnju koja će rezultirati s manjim brojem saobraćajnih nesreća i kolizija. Stoga, autonomna vozila koriste veliki broj senzora različitih tipova i kombinuju informacije sa ovih senzora u smislene zaključke, [25]. Pouzdanost senzora koji su deo autonomnih vozila i sistema za fuziju informacija sa senzora predstavljaju osnovu u ostvarivanju potpuno autonomne vožnje.

Senzori koji mogu biti deo autonomnog vozila su kamera, lidar, radar, sonar, globalni sistem za pozicioniranje, inercijalna merna jedinica i odometrija točkova. Senzori u autonomnim vozilima se koriste za prikupljanje podataka koji se dalje analiziraju unutar centralnog računara u autonomnom vozilu i koriste se za kontrolu upravljača, sistema za kočenje i kontrolu brzine vozila. Osim informacija sa senzora koji se nalaze montirani na vozilu i predstavljaju deo autonomnih vozila, dodatne informacije od značaja za autonomnu vožnju se mogu dobiti i putem internet mreže iz skladišnih kapaciteta smeštenih u oblaku, eng. *cloud computing*, a koje obično sadrže mape okruženja u kome se nalazi vozilo, ili se dodatne informacije mogu dobiti od drugih vozila putem vozilo-do-vozilo konekcije. Tri osnovna tipa senzora koji se koriste u autonomnim vozilima su kamera, radar i lidar.

### 2.2.1. Kamera

Percepcija okruženja autonomnog vozila se ostvaruje korišćenjem više različitih tipova senzora i senzorskih sistema. Međutim, kamere koje rade u domenu vidljivog dela svetlosnog spektra, slika 2.3, su osnovni i neizostavni tip senzora koji se koristi u autonomnim vozilima. S obzirom na to da je pravilna percepcija okruženja vozila najvažnija i polazna tačka u sistemu autonomnog vozila, kamera kao vizuelni senzor predstavlja primarni resurs u saznavanju okruženja u kome se vrši vožnja i okruženja sa kojim vozilo interaguje. Autonomna vozila koja se danas razvijaju imaju na desetine različitih kamera postavljenih na vozilo. Primeri izgleda nekih od kamera koje se danas koriste u sklopu autonomnih vozila prikazani su na slici 2.4.



Slika 2.3 Svetlosni spektar.



Kamere su široko rasprostranjeni senzori koji omogućavaju autonomnom vozilu da vizualizuje svoje okruženje. Prednost korišćenja kamera su dostupnost i cena koja je značajno niža od drugih tipova senzora koji se takođe mogu koristiti za vizuelizaciju, poput radara i lidara. Kamere najnovije generacije sadrže milione piksela u jednom frejmu, pri čemu se procesira 30 do 60 frejmova u sekundi. Ovo dovodi do informacije od više megabajta podataka koju je potrebno obraditi u realnom vremenu što utiče na podizanje zahteva za hardver koji procesira informacije sa kamere – u kontekstu računarske snage i memorijskog kapaciteta. Stoga, osnovni nedostatak kamera je to što visoka rezolucija slika zahteva korišćenje računara visokih performansi za obradu slika dobijenih sa kamere u realnom vremenu.



*Slika 2.4 Kamere u autonomnim vozilima. Levo – Flir Flea3, desno gore – ADAS kamera Mobileye i ZF S-Cam4, desno dole – stereo kamera Slightech MYNT EYE.*

Primene upotrebe kamera u autonomnim vozilima su brojne [26]-[30]. Primena uključuje percepciju, semantičku segmentaciju, autonomnu vožnju od-kraja-do-kraja i mnoge druge. Kamere se takođe mogu koristiti unutar vozila za čovek-mašina interakciju [31].

### **2.2.2. Radar**

Radar je tip senzora integrisan u autonomno vozilo i koristi se u svim nivoima autonomnog vozila, za zadatke poput prilagodljivog tempomata, upozorenja na sudar sa objektom koji se nalazi u mrtvom uglu, izbegavanje sudara, pa do naprednih zadataka kao što su prepoznavanje objekata i praćenje. Iako je radar zrela tehnologija, u toku su stalna poboljšanja, pogotovo za primenu u autonomnim vozilima [32]-[33]. Dok ostali senzori mere brzinu izračunavanjem razlike između dva merenja, radar koristi Doplerov efekat kako bi direktno merio brzinu. Doplerov efekat je važan za fuziju informacija sa senzora zato što daje informaciju o brzini kao nezavisnom parametru, čime osigurava da algoritmi fuzije brže konvergiraju.



*Slika 2.5 Izgled radara.*

Radar dugog dometa je mikrotalasni radar koji radi na 77 GHz, ima nisku rezoluciju i može da meri brzinu i detektuje vozila i prepreke do udaljenosti od 200 m. Radar kratkog ili srednjeg dometa je dovoljno razvijena i pristupačna tehnologija koja radi u opsezima od 24 GHz i 76 GHz. Ovakav senzor može da detektuje brzinu i udaljenost uočenog objekta. Međutim, široki snopovi talasa i dugačke talasne dužine ograničavaju rezoluciju radara i proizvode kompleksne povratne signale. Primer izgleda radara koji se koristi u autonomnim vozilima dat je na slici 2.5.

Radar se može koristiti za lokalizaciju generisanjem radarskih mapa, može da detektuje objekte ispod vozila, kao i da uoči zgrade i objekte koji bi drugim tipovima senzora bili zanemareni. Od svih senzora koji se koriste u autonomnim vozilima, na radar najmanje utiču vremenske prilike kao što su kiša ili magla. Radar može imati širok ugao vidljivosti, oko 150 stepeni, ili dug domet, preko 200 m. Iako je radar efikasniji od kamere i lidara u uslovima loših vremenskih prilika, radar ima manju ugaonu tačnost, nižu rezoluciju i generiše manje podataka od lidara. Prednost radara je to što je, za razliku od kamere i lidara, potrebno malo računarske snage za obradu informacija sa radara.

### 2.2.3. Lidar

Lidar, eng. *Light Detection and Ranging* (LiDAR), je tip senzora koji koristi infracrveni laserski snop za određivanje rastojanja od senzora do objekata koje posmatra. Većina trenutnih lidara koristi opseg talasne dužine od 900 nm. Neki od lidara koriste i duže talasne dužine od 900 nm čime postižu bolji rad u uslovima kiše i magle. Najčešći tip trenutno upotrebljivanih lidara u autonomnim vozilima su lidari sa rotacionom senzorskom glavom koji skenira objekte laserskim snopom duž projektovanog ugla vidljivosti. Laser pulsira, a impulsi se reflektuju na objektima. Ove refleksije generišu oblak tačaka, eng. *point cloud*, koji može da predstavlja objekte. Na primer, lidar visoke definicije sa rotirajućom senzorskom glavom koji sadrži 64 poluprovodnička lasera i omogućava efikasno generisanje trodimenzionalnih mapa sa visokim nivoom detalja je predstavljen u [34]. Primeri izgleda lidara sa rotirajućom senzorskom glavom su prikazani na slici 2.6.



Slika 2.6 Lidar sa rotirajućom senzorskom glavom, Velodyne HDL-64E, HDL-32E i VLP-16.

Lidar ima značajno veću prostornu rezoluciju od radara zato što proizvodi fokusirani laserski snop, veći broj skeniranih slojeva u vertikalnom pravcu i veću gustinu tačaka u svakom sloju. Lidar sa rotacionim stožerom ne može da meri brzinu objekata direktno i mora da se oslanja na različite pozicije objekta između dva ili više skeniranja. Takođe, na performanse lidara mogu uticati vremenski uslovi i zaprljanost senzora.

Drugi tip lidara su lidari sa mikroelektromehaničkim sistemima, eng. *Micro-Electro-Mechanical System* (MEMS), sa vibrirajućim mikroogledalima [35]. Umesto da mehanički pomeraju laserski snop kao kod lidara sa rotirajućom senzorskom glavom, lidari sa MEMS vibrirajućim mikroogledalima dele jedan laserski snop na višestruke talasne dužine, gde se fazni odnos između talasnih dužina može izmeriti i tako promeniti smer laserskog snopa. Primer izgleda lidara sa MEMS vibrirajućim ogledalom je prikazan na slici 2.7.



*Slika 2.7 Lidar sa MEMS vibrirajućim mikroogledalima [36].*

Prednosti lidara su visoka rezolucija koja je izuzetno značajna za identifikaciju objekata i mogućnost direktnog merenja brzine skeniranih objekata kod lidara sa MEMS vibrirajućim ogledalima. Lidari mogu da mapiraju statičko okruženje, kao i da detektuju objekte u pokretu, pešake ili životinje. Trenutno limitirajući faktori za upotrebu lidara u autonomnim vozilima su visoka cena i relativno velike dimenzije uređaja. Međutim, nova rešenja imaju tendenciju da smanje veličinu i cenu ovog senzora. Novi pravci u razvoju lidara u oblasti autonomnih vozila su predstavljeni u literaturi [37]-[41].

### **2.3. Podsistem za percepciju**

Podsistem za percepciju se sastoji od softverskih komponenti koje procesiraju informacije sa senzora. Zadaci podsistema za percepciju su detekcija objekata, drugih vozila ili pešaka na putu, detekcija i praćenje objekata u pokretu i semantička segmentacija puta. Komponente podsistema za percepciju kombinuju informacije dobijene sa različitih senzora, uglavnom sirove podatke, i pretvaraju ih u informacije od značaja. U ovom podsistemu se vrši analiza okruženja u kome se nalazi vozilo. Dominantne tehnike, metode i algoritmi koje koristi podsistem za percepciju pripadaju domenu kompjuterske vizije [42].

Sistem za percepciju se može grubo podeliti na dve celine: podsistem za detekciju i podsistem za lokalizaciju. Podsistem za lokalizaciju je odgovoran za korišćenje podataka sa senzora i mape kako bi zaključio koja je tačna lokacija vozila. Podsistem za detekciju je odgovoran za razumevanje okruženja u kome se nalazi vozilo. Ovaj podsistem se sastoji od softverskih komponenti kao što su detekcija kolovozne trake, detekcija i klasifikacija saobraćajnih znakova i svetlosne signalizacije, prepoznavanje i praćenje objekata i detekcija slobodnog prostora. Svaka komponenta sistema za percepciju se oslanja na različitu grupu senzora. Za prepoznavanje objekata se koriste kamera, lidar i radar, dok je primarni senzor za lokalizaciju lidar uz dodatak podataka sa mape. Sistem za percepciju prosleđuje podatke dobijene od podsistema za detekciju i lokalizaciju ka sistemu za planiranje. Podsistem za planiranje odlučuje koju sledeću akciju će vozilo preuzeti.

## 2.4. Planiranje i upravljanje

Podsistem za planiranje koristi informacije sa izlaza podsistema za percepciju kako bi planirali putanju i ponašanje vozila, uključujući kratkoročan i dugoročan opseg planiranja putanje. Komponente sistema za planiranje su:

- planiranje rute – određivanje putanje između dve tačke na mapi na višem nivou,
- predikcija – identifikuje koju akciju drugi objekti na putu mogu da ostvare,
- planiranje ponašanja – odlučuje koju akciju vozilo treba da preuzme,
- planiranje trajektorije – koje delove precizne putanje će vozilo pratiti.

Planiranje rute je komponenta odgovorna za odluke na visokom nivou u vezi rute vozila između dve tačke na mapi. Odnosi se na određivanje ulica, puteva i autoputeva koje vozilo može koristiti između ove dve tačke na mapi. Predikcija se bavi predviđanjem mogućih akcija drugih objekata na putu, kao što je predviđanje trajektorije drugih vozila na putu ili ponašanja pešaka. Planiranje ponašanja je komponenta koja definiše akcije koje će samo vozilo preuzeti u svakom vremenskom trenutku, što može biti zaustavljanje automobila kada je uključeno crveno svetlo na svetlosnoj signalizaciji, menjanje traka kojima se kreće vozilo, ubrzavanje vozila, skretanje. Na osnovu definisanog željenog ponašanja vozila u datom trenutku, planiranje trajektorije utvrđuje najbolju putanju za izvršavanje takvog ponašanja.

Tradicionalni pristupi u planiranju trajektorije su tehnike upravljanja bazirane na optimizaciji, koje redukuju definiciju problema tako da je on podložan optimizaciji. U ovom slučaju, definiše se više pretpostavki, nakon čega se generišu hiljade mogućih trajektorija, i konačno koristi se objektivna funkcija za odlučivanje koja od generisanih trajektorija bira najbolje rešenje za date uslove na putu. Međutim, korišćenjem tehnika dubokog učenja i učenja podsticanjem, eng. *reinforcement learning*, postiže se bolja robusnost u planiranju trajektorije i generalno sistem bolje reaguje u nepredviđenim situacijama, kada postoji nedostatak u kalibraciji senzora ili nedostatak prethodnih informacija na mapi [43].

Podsistem za upravljanje osigurava da vozilo prati put koji je definisao podsistem za planiranje i šalje upravljačke komande ka vozilu. Podsistem za upravljanje se sastoji od softverskih komponenti koje osiguravaju da vozilo prati putanju definisanu podsistemom za planiranje. Komponente ovog podsistema su proporcionalni-integralni-diferencijalni (PID) softverski regulatori, regulatori modela prediktivne kontrole, eng. *model predictive control* (MPC) softverski regulatori i drugi regulatori ili kotnroleri [44]. Podsistem za upravljanje šalje komande za ubrzanje, kočenje i upravljanje pravcem vozila. Ovim se kompletira tok informacija od senzora do akcije čime se omogućava autonomna vožnja datog vozila.

## 3. Duboko učenje

### 3.1. Veštačka inteligencija, mašinsko učenje i duboko učenje

Veštačka inteligencija je nastala 50-tih godina prošlog veka sa ciljem da se kompjuteri nauče da misle. Veštačka inteligencija se definiše kao napor da se automatizuju intelektualni zadaci koje inače ljudi obavljaju. Takođe, ona se može definisati kao proučavanje inteligentnih agenata koji čine bilo koji uređaj koji percipira svoje okruženje i preduzima akcije koje maksimizuju šanse za uspeh u datom zadatku. Kolokvijalno, ovaj termin se koristi kada mašina imitira kognitivne funkcije kao što su učenje i rešavanje problema.

Veštačka inteligencija je oblast koja objedinjuje mašinsko učenje i duboko učenje, pri čemu takođe uključuje i zadatke za koje nije potrebno kakvo učenje. Mašinsko učenje je podskup veštačke inteligencije, dok je duboko učenje podskup mašinskog učenja, slika 3.1. Na primer, prvi programi za igranje šaha su uključivali samo fiksno napisana pravila od strane programera i oni ne predstavljaju mašinsko učenje. Simbolična veštačka inteligencija je vrsta veštačke inteligencije gde se teži postizanju inteligencije na nivou čoveka pisanjem ekstremno velikog skupa fiksnih pravila za manipulaciju znanjem. Ovo je bila dominantna paradigma od 50-tih do 80-tih godina prošlog veka, pri čemu je 80-tih doživela vrhunac razvojem ekspertskih sistema.



Slika 3.1 Veštačka inteligencija, mašinsko učenje i duboko učenje.

Artur Semjuel, pionir u oblasti veštačke inteligencije i kompjuterskih igara, je definisao mašinsko učenje kao oblast studiranja koja daje kompjuterima mogućnost da uče iako nisu eksplicitno programirani. On je isprogramirao igru dame (eng. *checkers*). Mašinsko učenje (eng. *machine learning* – ML) je oblast veštačke inteligencije koja se zasniva na kompjuterima koji uče o okruženju iz podataka, umesto da su zasnovani na pravilima postavljenim od strane programera, slika 3.2, [45].



Slika 3.2 Klasično učenje i mašinsko učenje.

Mašinsko učenje je nastalo iz pitanja mogu li kompjuteri ići van ljudskog znanja kako se određeni problem rešava, da sami nauče kako da izvršavaju dat zadatak. U klasičnom programiranju, uključujući i simboličku veštačku inteligenciju, ljudi daju programu ulaze u vidu pravila i podatke koji treba da budu procesuirani na osnovu datih pravila, a kao izlaz očekuje se odgovor. Sa druge strane, kod mašinskog učenja, čovek daje mašini i podatke i odgovore, a kao izlaz očekuju se pravila. Ova pravila se dalje mogu primeniti na nove podatke i tako se dobija originalan odgovor generisan na osnovu naučenih pravila.

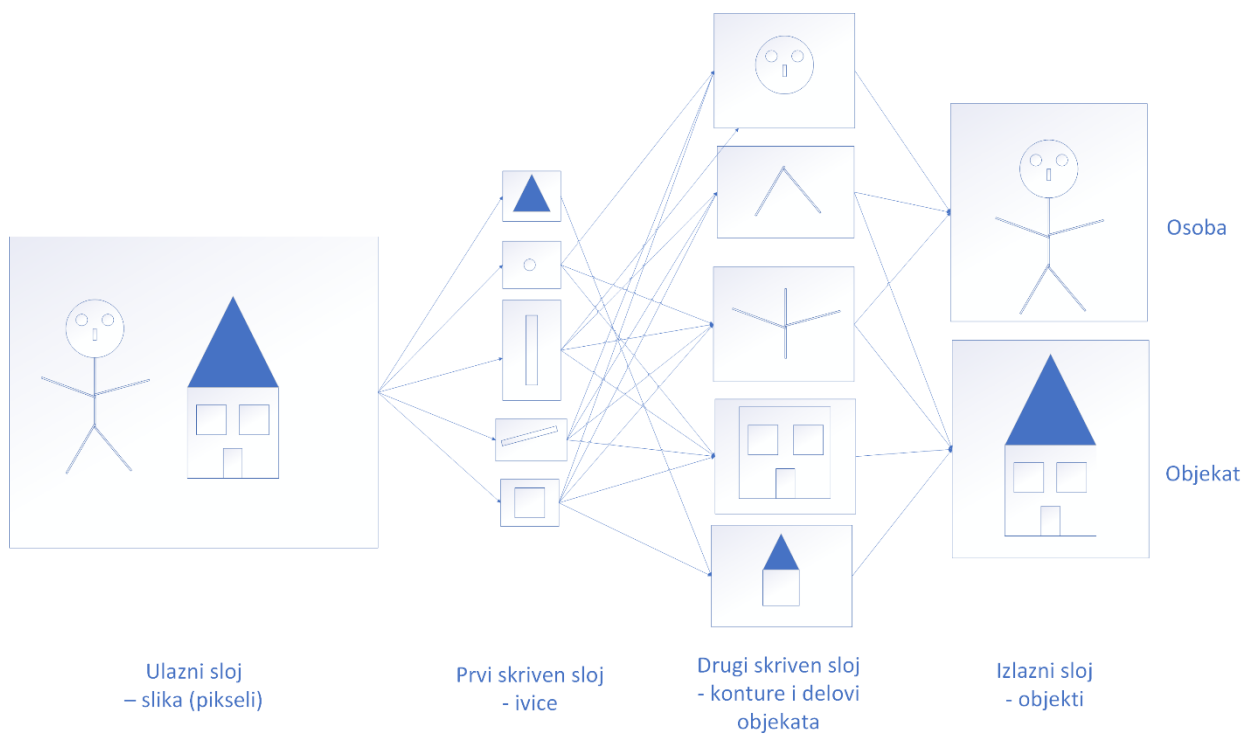
Sistem mašinskog učenja je treniran, a ne eksplicitno programiran. Ovakvom sistemu se predstave brojni primeri relevantni za zadatak, gde sistem mašinskog učenja pronalazi statističku strukturu u ovim primerima koja dovodi do toga da sistem zaključi pravila za automatizaciju zadatka. Mašinsko učenje je usko povezano sa matematičkom statistikom, ali se razlikuje od statistike na nekoliko važnih načina. Nasuprot statistici, mašinsko učenje teži da sadrži velike i kompleksne skupove podataka, kao što su milioni slika, pri čemu svaka ima nekoliko hiljada piksela. Za velike i kompleksne skupove podataka klasična statistika, npr. Bajsova analiza, bi bila nepraktična. Kao rezultat, mašinsko učenje, i pre svega duboko učenje, se sastoje od komparativno malo matematičke teorije i više su inženjerski orijentisane tehnike. Ovo je praktična disciplina u kojoj se ideje češće dokazuju empirijski nego teorijski.

Zadatak mašinskog učenja je da otkrije pravila po kojima će izvršiti zadatak obrade ulaznog podatka, tako što se algoritmu mašinskog učenja daju primeri ulaznih podataka i očekivanog odgovora. U ovom kontekstu elemente neophodne za ostvarivanje uspešnog modela mašinskog učenja možemo podeliti u tri kategorije:

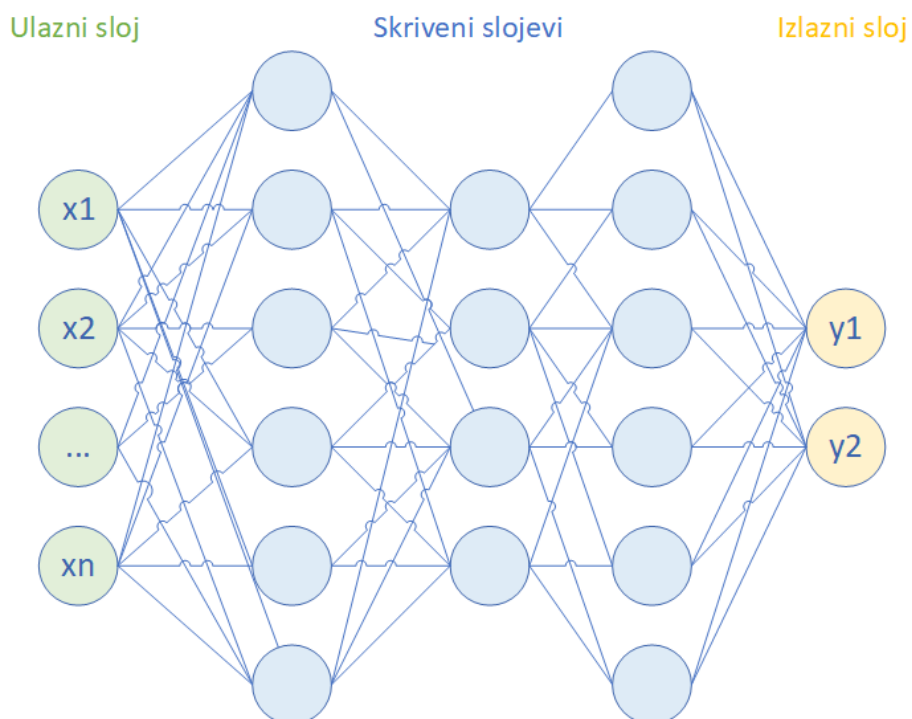
- Ulazni podaci – na primer zvučni fajlovi koji sadrže ljudski govor, za zadatak prepoznavanja govora; ili slike za zadatak prepoznavanja objekata na slici.
- Primeri očekivanog izlaza – u primeru prepoznavanja govora izlaz može biti transkript zvučnih fajlova; u slučaju prepoznavanja objekata izlaz mogu biti kategorije „pas“, „mačka“, „osoba“, „kuća“, itd.
- Mera kvaliteta uspešnosti algoritma – ovo je neophodno kako bismo odredili razliku između trenutnog izlaza projektovanog algoritma i očekivanog izlaza. Ovakva mera služi kao povratna sprega kako bi se prilagodio i fino podesio sam algoritam. Prilagođavanje algoritma se naziva učenje, [45].

Model mašinskog učenja transformiše ulazne podatke u izlaz sa određenim značenjem. Ova transformacija se uči tako što se model izloži poznatim primerima ulaza i izlaza. Stoga, centralni problem mašinskog i dubokog učenja je kako da se smisleno transformišu podaci, da se nauče korisne reprezentacije ulaznih podataka. Dubokim učenjem se smatra deo šire porodice metoda mašinskog učenja baziranih na učenju reprezentacije podataka [45-48]. Reprezentacije jednog sloja duboke neuralne mreže se kreiraju kao rezultat više jednostavnijih reprezentacija iz prethodnog sloja duboke neuralne mreže, slika 3.3.

Osnovni gradivni element dubokih neuralnih mreža su konvolucione mreže [49]. Konvolucione neuralne mreže (eng. *convolutional neural network* – CNN) su neuralne mreže za procesiranje podataka bazirane na povezivanju čvorova koji se nazivaju neuroni, slika 3.4. Konvolucione neuralne mreže su dizajnirane za procesiranje podataka koji se mogu predstaviti u višestrukim nizovima, npr. slika, govor, audio spektrogram i video. CNN se najčešće koriste za analizu vizuelnih signala, pre svega slike sa kamere, ali i drugih vizuelnih senzora, npr. lidara.



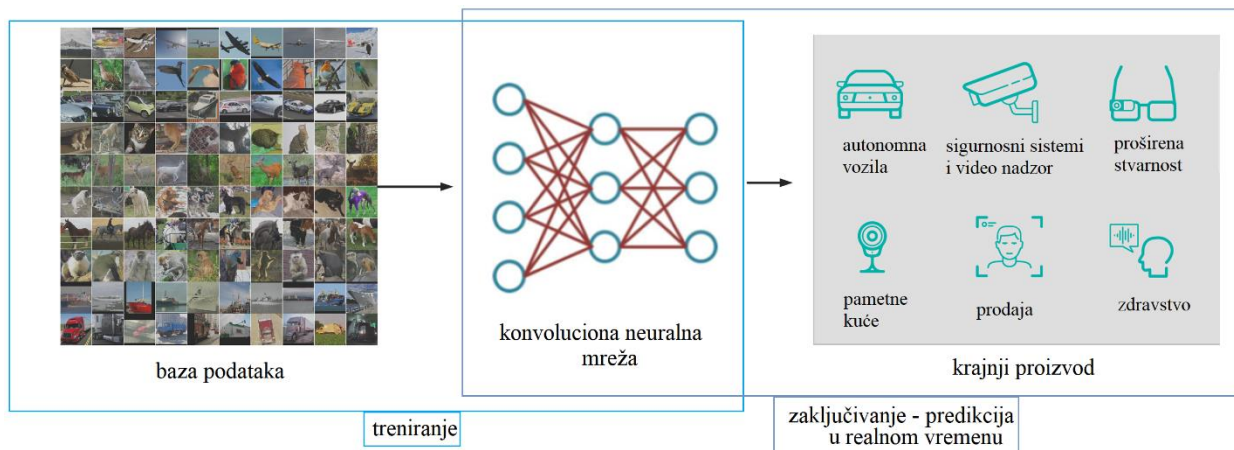
*Slika 3.3 Reprzentacije jednog sloja duboke neuralne mreže nastaju kombinovanjem više jednostavnijih reprzentacija iz prethodnog sloja duboke neuralne mreže.*



*Slika 3.4 Duboka neuralna mreža.*



Duboko učenje za kompjutersku viziju ima veliki uticaj na razvoj različitih komercijalnih i industrijskih sistema i proizvoda, u domenima kao što su: automobilska industrija, bezbednost i nadzor, proširena stvarnost, pametne kuće, zdravstvo, industrija video igara. Generalni tok razvoja i primene tehnike dubokog učenja je prikazan na slici 3.5 i podrazumeva tri faze: (1) projektovanje neuralne mreže, čime se dobija netrenirani model neuralne mreže; (2) treniranje neuralne mreže, gde se isprojektovani model neuralne mreže trenira korišćenjem baze podataka koja je kreirana u skladu sa problemom koji se rešava; i (3) zaključivanje, odnosno izvršavanje predikcija u realnom vremenu, na osnovu ulaznog signala sa senzora uređaja na kome se vrši mašinsko učenje a korišćenjem istreniranog modela neuralne mreže koji je dobijen u prethodnoj fazi.



Slika 3.5 Tok primene metoda dubokog učenja: najpre se dizajnira duboka konvoluciona neuralna mreža, nakon čega se korišćenjem odgovarajuće baze podataka mreža trenira. Konačno, istreniran model neuralne mreže se u realnom vremenu koristi za predikciju definisani.

Konvolucione neuralne mreže su među prvim modelima dubokog učenja koji su pronašli praktičnu realizaciju i dobar učinak pri rešavanju važnih komercijalnih aplikacija. Jedan od prvih primera praktične primene konvolucionih neuralnih mreža je iz 1990. godine, rešenje razvijeno od strane istraživačke grupe kompanije AT&T, Inc, gde se CNN koristi za optičko prepoznavanje znakova i proveru čitanja adresa na poštanskim pošiljkama [7]. Nakon ovoga, više rešenja za optičko prepoznavanje znakova i prepoznavanje rukopisa je razvijeno bazirajući se na konvolucionim neuralnim mrežama [50], dok su najnovije primene CNN-a za kompjutersku viziju izuzetno brojne i bave se rešavanjem problema iz različitih domena [51-54].

## 3.2. Višeslojne neuralne mreže

### 3.2.1. Perceptron

Kod jednoslojnih mreža, ulazni signali se direktno mapiraju na izlazni signal korišćenjem generalizovane varijacije linearne funkcije. Ovakva jednostavna instanca neuralne mreže se naziva perceptron [47-48]. Perceptron se sastoji od jednog ulaznog sloja i jednog izlaznog čvora. Osnovna arhitektura perceptrona je prikazana na slici 3.6. Ukoliko jednu instancu za treniranje označimo sa  $(\mathbf{X}, y)$ , gde se svaki  $\mathbf{X} = [x_1, x_2, \dots, x_n]$  sastoji od  $n$  ulaznih varijabli, a  $y \in \{-1, 1\}$  je uočena vrednost binarne varijable. Ovde se termin uočena vrednost odnosi na činjenicu da je ova vrednost data kao deo podatka za treniranje, a naš cilj je da predvidimo klasu varijable za nove ulazne podatke za koje je  $y$  nepoznato.



Ulazni sloj se sastoji od  $n$  čvorova koji sadrže  $n$  odlika ulaznog signala, a svakom od ovih čvorova se dodeljuje težinski faktor  $\mathbf{W} = [\omega_1, \omega_2, \dots, \omega_n]$ . Na izlaznom čvoru se računa linearna funkcija  $\mathbf{W} \cdot \mathbf{X} = \sum_{i=1}^n \omega_i x_i$ , a zatim se znak ove realne vrednosti (+1 ili -1) koristi kao predikcija ulaznog podatka  $\mathbf{X}$ . Dakle, predikcija  $\hat{y}$  se izračunava na sledeći način:

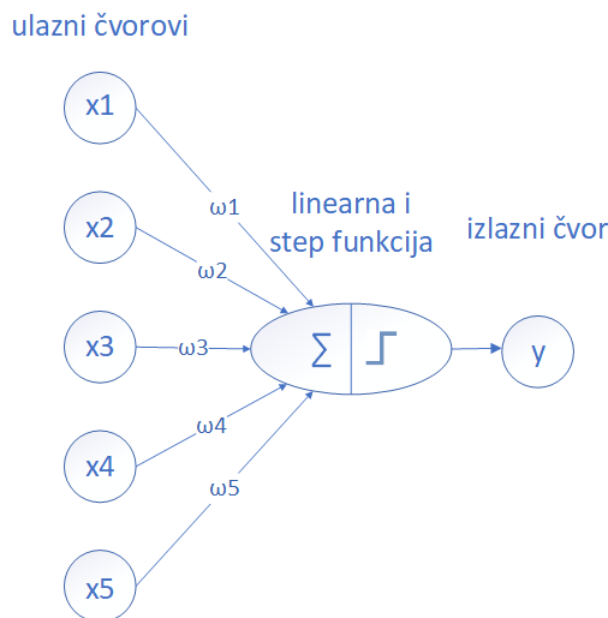
$$\hat{y} = \text{sign}\{\mathbf{W} \cdot \mathbf{X}\} = \text{sign}\left\{\sum_{i=1}^n \omega_i x_i\right\} \quad (3.1)$$

Ovako mapirana izlazna vrednost je pogodna za binarnu klasifikaciju ulaznog signala. Greška se izračunava po formuli:

$$E(\mathbf{X}) = y - \hat{y} \quad (3.2)$$

i može imati jedino vrednost iz skupa  $\{-2, 0, 2\}$ . Kada je greška za  $E(\mathbf{X})$  različita od nule, težinski faktori neuralne mreže se koriguju u negativnom smeru gradijenta greške.

Prethodno opisan postupak je sličan tradicionalnim modelima mašinskog učenja. Međutim, interpretacija perceptrona kao računarske jedinice je korisna iz razloga što omogućava da se višestruke jedinice spoje i tako kreiraju daleko moćnije modele u odnosu na modele tradicionalnog mašinskog učenja.



Slika 3.6 Perceptron – jednoslojna neuralna mreža.

Na slici 3.6 je pokazana arhitektura perceptrona gde se karakteristike jednog ulaznog sloja transformišu u izlazni čvor. Ovde se karakteristike ulaznog sloja množe sa težinskim faktorima  $\omega_i$ , nakon čega se zbirna vrednost ovih proizvoda konvertuje u labelu klase korišćenjem funkcije znaka. U ovom slučaju funkcija znaka se naziva aktivaciona funkcija. Iako se perceptron sastoji od dva sloja, ulaznog i izlaznog, s obzirom da se u ulaznom sloju ne vrši nikakvo izračunavanje, već se samo prenose karakteristike ulaznog signala, ulazni sloj se ne računa u ukupnom broju slojeva neuralnih mreža. Stoga, možemo reći da je perceptron jednoslojna neuralna mreža.

### 3.2.2. Aktivaciona funkcija

Aktivaciona funkcija čvora neuralne mreže definiše izlaz tog čvora na osnovu ulaza ili skupa ulaza. Izbor aktivacione funkcije je važan deo dizajna neuralnih mreža. Za perceptron je korišćena funkcija znaka kao aktivaciona funkcija s obzirom da je izlaz iz ove neuralne mreže binarna klasifikacija. Međutim, važnost nelinearnih aktivacionih funkcija je značajna kada se radi o

višeslojnim arhitekturama. Različite nelinearne funkcije se mogu koristiti u različitim slojevima neuralne mreže. Koristićemo notaciju  $\Phi$  da označimo aktivacionu funkciju:

$$\hat{y} = \Phi(W \cdot X) \quad (3.3)$$

Neuron računa dve funkcije unutar čvora, sumu proizvoda vrednosti ulaznih čvorova i njihovih težinskih faktora, i aktivacionu funkciju. Izlaz iz neurona je uvek postaktivaciona vrednost, mada se i preaktivacione vrednosti često koriste u različitim analizama, kao što je slučaj kod algoritma za propagaciju unazad.

Osnovna aktivaciona funkcija  $\Phi(\cdot)$  je linearna aktivacija (eng. *identity*), slika 3.7(a):

$$\Phi(v) = v, \quad (3.4)$$

Linearna aktivaciona funkcija se često koristi u izlaznom čvoru, kada je cilj realna vrednost. Klasične aktivacione funkcije koje se koriste kod neuralnih mreža su funkcija znaka, *sigmoid* i hiperbolička tangenta funkcija:

$$\Phi(v) = \text{sign}(v), \quad (3.5)$$

funkcija znaka, slika 3.7(b).

$$\Phi(v) = \frac{1}{1 + e^{-v}}, \quad (3.6)$$

*sigmoid* funkcija, slika 3.7(c).

$$\Phi(v) = \frac{e^{2v} - 1}{e^{2v} + 1}, \quad (3.7)$$

*tanh* funkcija, slika 3.7(d).

Dok se funkcija znaka koristi za mapiranje binarnog izlaza u vreme predikcije, njena nediferencijabilnost onemogućava korišćenje ove funkcije za kreiranje funkcije gubitka u vreme treniranja. Sigmoid aktivaciona funkcija daje vrednost (0,1) što je od značaja kod izračunavanja koja je potrebno interpretirati kao verovatnoće. *Tanh* funkcija ima sličan oblik *sigmoid* funkciji, osim što je horizontalno skalirana a vertikalno translirana i skalirana, pa je izlaz ove funkcije [-1, 1]. *Tanh* i *sigmoid* aktivacione funkcije su povezane na sledeći način:

$$\tanh(v) = 2 \cdot \text{sigmoid}(2v) - 1. \quad (3.8)$$

*Tanh* funkcija se koristi radije nego *sigmoid* aktivaciona funkcija kada se želi da izlazni rezultat bude i pozitivan i negativan. Takođe, *tanh* je centralizovana funkcija sa većim gradijentom u odnosu na *sigmoid*, što je čini lakšom za treniranje. *Sigmoid* i *tanh* funkcija su u ranijim godinama bile osnovne aktivacione funkcije koje su uvodile nelinearnost u neuralne mreže. Međutim, u poslednjih nekoliko godina, prednost uzimaju nove aktivacione funkcije ReLU (eng. *Rectified Linear Unit*) i *hard tanh*:

$$\Phi(v) = \max\{v, 0\}, \quad (3.9)$$

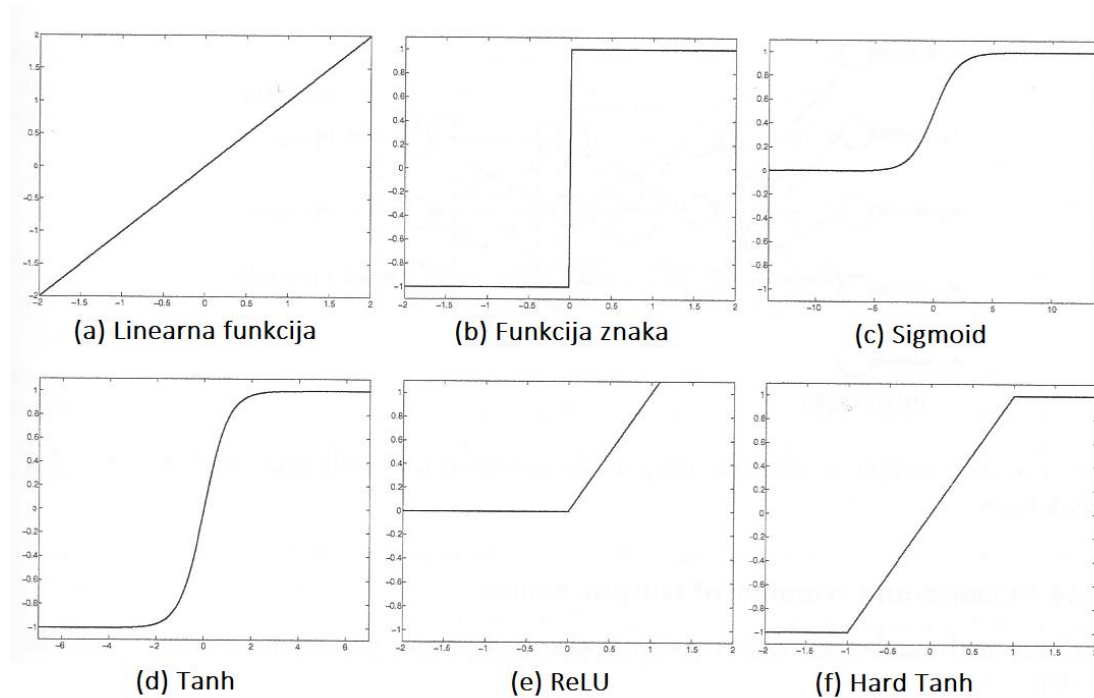
ReLU funkcija, slika 3.7(e).

$$\Phi(v) = \max\{\min[v, 1], -1\}, \quad (3.10)$$

*hard tanh* funkcija, slika 3.7(f).

ReLU i *hard tanh* aktivacione funkcije su uglavnom zamenile *sigmoid* i *tanh* aktivacione funkcije u modernim neuralnim mrežama stoga što je korišćenjem novih aktivacionih funkcija trening neuralnih mreža pojednostavljen. Grafički prikaz svih pomenutih aktivacionih funkcija je prikazan na slici 3.7. Može se uočiti da su sve prikazane aktivacione funkcije monotone. Nelinearne aktivacione funkcije su veoma korisne kod višeslojnih neuralnih mreža, jer omogućavaju kreiranje

složenih kompozicija različitih tipova funkcija. Ukoliko bi neuralna mreža koristila samo linearne aktivatore, to ne bi omogućilo bolje modelovanje od korišćenja jednoslojne linearne mreže.



Slika 3.7 Aktivacione funkcije.

### 3.2.3. Funkcija gubitka

U matematičkoj optimizaciji i teoriji odlučivanja, funkcija gubitka (eng. *loss function*) ili funkcija troška (eng. *cost function*) je funkcija koja preslikava događaj ili vrednosti jedne ili više promenljivih na realan broj intuitivno predstavljajući „trošak“ povezan sa ovim preslikavanjem. Optimizacijom se nastoji da se minimalizuje funkcija gubitka. Objektivna funkcija je ili funkcija gubitka ili njena negativna vrednost, funkcija dobitka (eng. *reward function*), u čijem slučaju je potrebno maksimizovati ovu funkciju.

Izbor funkcije gubitka je ključan u definisanju izlaza na način koji najviše odgovara praktičnoj aplikaciji za koju se primenjuje. Na primer, metoda najmanjih kvadrata sa numeričkim izlazima zahteva jednostavnu kvadratnu funkciju gubitka u obliku  $(y - \hat{y})^2$  za jednu instancu treninga sa ciljanim izlazom  $y$  i predikcijom  $\hat{y}$  koristeći linearnu aktivaciju:

$$L = \max\{0, 1 - y \cdot \hat{y}\}. \quad (3.11)$$

Za predikcije više klasa najčešće se koristi *softmax* funkcija. *Softmax* funkcija je funkcija koja uzima vektor sa  $K$  realnih brojeva i normalizuje ga u distribuciju verovatnoća, pri čemu ova distribucija sadrži  $K$  verovatnoća proporcionalnih eksponencijalnim vrednostima ulaznih brojeva. Stoga, pre primene *softmax* funkcije vektor može sadržati negativne brojeve ili brojeve veće od jedan, i njihov zbir ne mora biti jednak broju jedan. Međutim, nakon primene *softmax* funkcije svaki element vektora će uzimati jednu od vrednosti unutar intervala  $(0, 1)$  i zbir elemenata vektora će iznositi  $1$ . *Softmax* se često koristi u neuralnim mrežama kako bi se mapirao nenormalizovani izlaz mreže sa distribucijom verovatnoća predviđenih izlaznih klasa. Standardna *softmax* funkcija  $\sigma : \mathbb{R}^K \rightarrow \mathbb{R}^K$  se definiše formulom:

$$\sigma(\mathbf{y})_i = \frac{e^{y_i}}{\sum_{j=1}^K e^{y_j}} \quad (3.12)$$

za  $i = 1, \dots, K$  i  $\mathbf{y} = (y_1, \dots, y_K) \in \mathbb{R}^K$ .

Rezultat *softmax* funkcije je probabilistički. Za probabilističke predikcije se koriste dva različita tipa funkcije gubitka, zavisno od toga da li su predikcije binarnog tipa ili je u pitanju višestruka kategorizacija: logistička regresija i gubitak unakrsne entropije (eng. *cross-entropy loss*).

Logistička regresija se koristi u slučaju binarnih predikcija. Pretpostavimo da posmatrani izlaz  $y$  može uzeti vrednosti  $\{-1, 1\}$ , i da je predikcija  $\hat{y}$  arbitrarna numerička vrednost linearne aktivacione funkcije. U ovom slučaju, funkcija gubitka za jednu instancu posmatrane vrednosti  $y$  i zaista dobijene predikcije  $\hat{y}$  korišćenjem linearne aktivacije, se definiše kao:

$$L = \log(1 + e^{-y \cdot \hat{y}}) \quad (3.13)$$

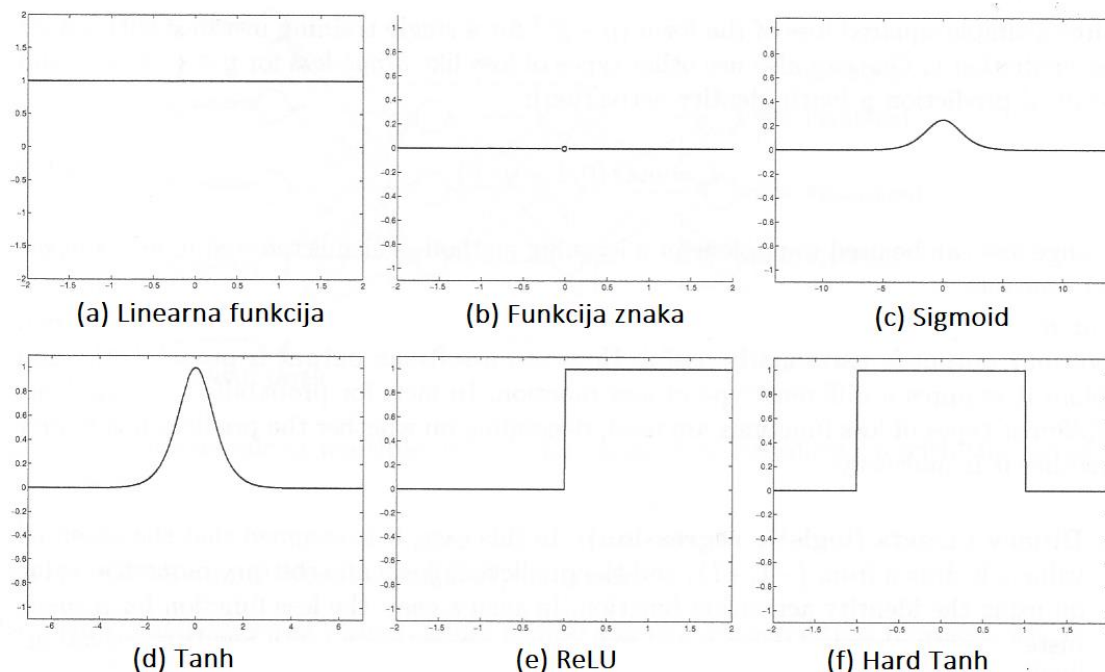
Kada je u pitanju klasifikacija više različitih mogućih vrednosti izlaza, definišemo  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_k$  kao verovatnoće  $k$  klasa, dok je  $r$ -ta klasa tačna klasa (eng. *ground truth*). U ovom slučaju funkcija gubitka se definiše kao:

$$L = -\log(\hat{y}_r) \quad (3.14)$$

Gubitak unakrsne entropije implementira multinominalnu logističku regresiju, i naziva se gubitak unakrsne entropije. Takođe, binarna logistička regresija je identična multinominalnoj logističkoj regresiji kada se definiše  $k = 2$ .

Odabir odgovarajuće aktivacione funkcije i funkcije gubitka zavisi od konkretne aplikacije modela neuralne mreže. Takođe izbor aktivacione funkcije i funkcije gubitka je uzajamno zavisna. Za diskretne izlaze, uobičajeno je da se koristi *softmax* aktivaciona funkcija sa funkcijom gubitka unakrsne entropije. Za izlaze predstavljene realnim vrednostima, uobičajeno je da se koristi linearna aktivacija sa kvadratnom funkcijom gubitka.

Na slici 3.8. prikazani su izvodi aktivacionih funkcija. Izvod linearne aktivacione funkcije je 1 za sve vrednosti ulaznog signala, slika 3.8(a). Izvod funkcije sign( $v$ ) je 0 za svako  $v$  različito od 0. U tački  $v = 0$  funkcija nije diferencijabilna, slika 3.8(b), pa se ova funkcija retko koristi kao funkcija gubitka.



Slika 3.8 Izvodi aktivacionih funkcija.

Kod *sigmoid* aktivacione funkcije, izračunavanje izvoda funkcije je izuzetno jednostavno. Ako definišemo  $o$  kao rezultat aktivacione funkcije sa argumentom  $v$ :

$$o = \frac{1}{1 + e^{-v}} \quad (3.15)$$

izvod ove aktivacione funkcije je:

$$\frac{\partial o}{\partial v} = \frac{e^{-v}}{(1 + e^{-v})^2}, \quad (3.16)$$

što je prikazano na slici 3.8(c). Ključno je to što se *sigmoid* može napisati pogodnije kao:

$$\frac{\partial o}{\partial v} = o(1 - o). \quad (3.17)$$

Slično važi i za *tanh* aktivacionu funkciju, slika 3.8(d):

$$o = \frac{e^{2v} - 1}{e^{2v} + 1} \quad (3.18)$$

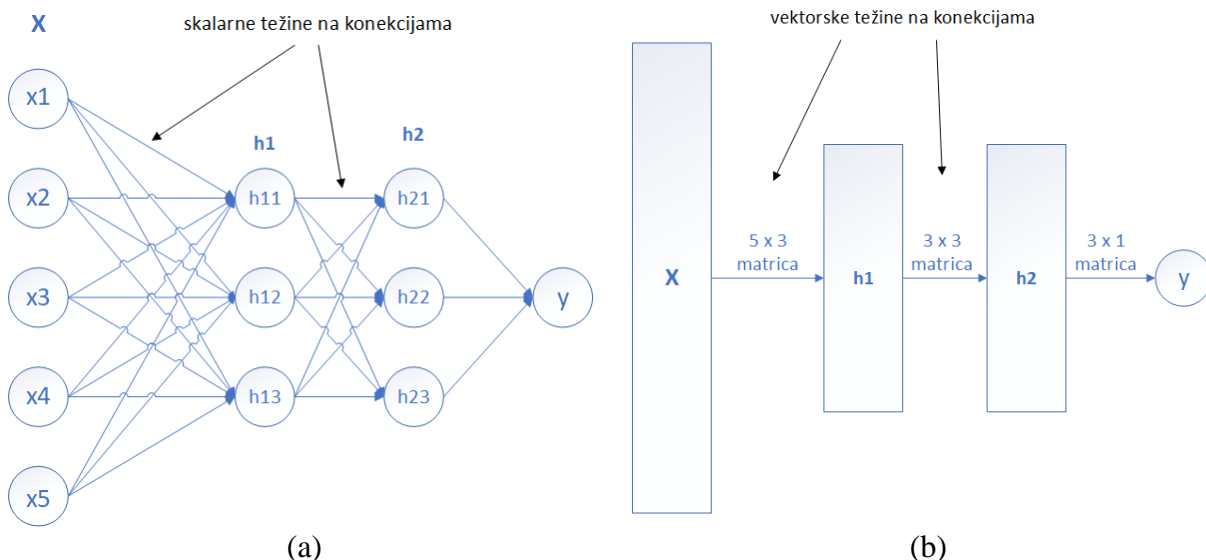
$$\frac{\partial o}{\partial v} = \frac{4 \cdot e^{2v}}{(e^{2v} + 1)^2} = 1 - o^2. \quad (3.19)$$

Izvod ReLU aktivacione funkcije je 1 za nenegativne vrednosti i 0 za negativne vrednosti ulaznog signala, slika 3.8(e). Kod *hard tanh* funkcije, izvod je 1 za ulazni signal od  $[-1, 1]$  i 0 za ostale vrednosti ulaznog signala, slika 3.8(f).

### 3.2.4. Arhitektura višeslojnih neuralnih mreža

Višeslojne neuralne mreže se sastoje od više od jednog sloja. Ove mreže se sastoje od ulaznog sloja, dodatnih skrivenih slojeva između ulaza i izlaza i izlaznog sloja, slika 3.4. Specifična arhitektura višeslojnih neuralnih mreža se naziva i mreža sa propagacijom unapred (eng. *feed-forward*). Osnovna arhitektura mreže sa propagacijom unapred pretpostavlja da su svi čvorovi u jednom sloju povezani sa svim čvorovima u narednom sloju. Ovim je arhitektura neuralne mreže gotovo u potpunosti definisana, kada imamo broj slojeva i broj ulaznih i izlaznih čvorova. Za potpunu definiciju arhitekture ovakve neuralne mreže potrebno je još odrediti funkciju gubitka koja se optimizuje u izlaznom sloju.

Na slici 3.9 prikazana je osnovna arhitektura jedne neuralne mreže sa propagacijom unapred koja se sastoji iz tri sloja. Na slici 3.9(a) je prikazana skalarna notacija arhitekture, a na slici 3.9(b) vektorska notacija predstavljene arhitekture. Ukoliko se neuralna mreža sastoji od  $p_1, p_2, \dots, p_k$  čvorova u svakom sloju  $k$ , vektorska reprezentacija njihovih izlaza,  $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k$ , ima dimenziju  $p_1, p_2, \dots, p_k$ . Stoga možemo reći da se broj čvorova u jednom sloju smatra dimenzijom tog sloja. Iako se svaki čvor sastoji od skalarne promenljive, najčešća reprezentacija jednog sloja neuralne mreže je vektor, a konekcije između dva sloja se predstavljaju matricama, kao što je prikazano na slici 3.9(b).



Slika 3.9 Arhitektura jedne troslojne neuralne mreže; (a) skalarna notacija; (b) vektorska notacija.

Težinski faktori na konekcijama između ulaznog i prvog skrivenog sloja su predstavljeni matricom  $\mathbf{W}_1$  dimenzije  $d \times p_1$ , dok su težinski faktori između sloja  $r$  i sloja  $(r + 1)$  predstavljeni matricom  $\mathbf{W}_r$  dimenzije  $p_r \times p_{r+1}$ . Ukoliko se izlazni sloj sastoji iz  $o$  čvorova, onda je matrica  $\mathbf{W}_{k+1}$  dimenzije  $p_k \times p_o$ . Ulazni vektor  $\mathbf{x}$  dimenzije  $d$  se transformiše u izlazni rezultat korišćenjem rekursivnih jednačina 3.20, 3.21 i 3.22.

Transformacija od ulaza do prvog skrivenog sloja:

$$\mathbf{h}_1 = \Phi(\mathbf{W}_1^T \mathbf{x}) \quad (3.20)$$

Transformacija od skrivenog do skrivenog sloja:

$$\mathbf{h}_p = \Phi(\mathbf{W}_{p+1}^T \mathbf{h}_p), \quad \forall p \in \{1, \dots, k-1\} \quad (3.21)$$

Transformacija od skrivenog sloja do izlaza:

$$o = \Phi(\mathbf{W}_{k+1}^T \mathbf{h}_k) \quad (3.22)$$

gde se aktivacione funkcije primenjuju nad svakim elementom unutar vektora. Međutim, neke aktivacione funkcije, poput *softmax* funkcije, već imaju argumente vektora.

### 3.2.5. Treniranje neuralne mreže koristeći propagaciju unazad

Kod jednoslojnih neuralnih mreža, proces treniranja je jednostavan s obzirom na to da se funkcija gubitka može računati direktno na osnovu vrednosti težinskih faktora što omogućava jednostavno izračunavanje gradijenta. U slučaju višeslojnih mreža, funkcija gubitka je kompleksna funkcija koja obuhvata funkcije gubitka računane nad težinskim faktorima u prethodnim slojevima. Gradijent ove funkcije gubitka se računa korišćenjem algoritma za propagaciju unazad.

Algoritam propagacije unazad koristi izvod složene funkcije (eng. *chain rule*), jednačina 3.23, koji računa grešku gradijenta u slučaju sume proizvoda lokalnih gradijenta kroz različite puteve od čvora do izlaza:

$$(f \circ g)' = (f' \circ g) \cdot g' \quad (3.23)$$

Algoritam propagacije unazad se sastoji iz dve faze: faza unapred i faza unazad. U fazi unapred se izračunavaju vrednosti izlaza i lokalni izvodi po svakom čvoru, dok se u fazi unazad akumuliraju proizvodi lokalnih vrednosti kroz sve putanje od jednog čvora do izlaza.

### 3.3. Konvolucione neuralne mreže

Konvolucione neuralne mreže su specijalizovane neuralne mreže za procesiranje podataka koji se mogu predstaviti od matricnom strukturom, npr. vremenska serija podataka koja se predstavlja 1-D vektor, slika koja je 2-D matrica piksela, ili lidar podaci koji predstavljaju 3-D matrica piksela. Konvolucija je specijalizovana linerana operacija. Konvolucione neuralne mreže su neuralne mreže koje koriste konvoluciju umesto množenja matrica na minimum jednom sloju neuralne mreže [46].

Konvolucija je matematička operacija dve funkcije,  $f$  i  $g$ , kojom se generiše treća funkcija ( $f * g$ ) koja odražava modifikaciju koju je druga funkcija načinila nad prvom, jednačina 3.24. Termin konvolucija se odnosi i na rezultujuću funkciju i na proces njenog izračunavanja.

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau \quad (3.24)$$

Definišimo funkciju  $s(t)$  kao rezultat konvolucije nad funkcijom  $x(a)$  koja predstavlja ulazni podatak i  $\omega(a)$  koja predstavlja funkciju težinskih faktora čvorova mreže koja se u ovom slučaju naziva kernel. Rezultat ove konvolucije se još naziva i mapa odlika (eng. *feature map*). Tada konvoluciju  $s(t)$  možemo prikazati jednačinama 3.25 i 3.26.

$$s(t) = \int_{-\infty}^{\infty} x(a)\omega(t - a) da \quad (3.25)$$

$$s(t) = (x * \omega)(t) \quad (3.26)$$

Kako se ulazni podaci sa kojima radimo mogu diskretizovati, konvoluciona funkcija nad ovakvim ulaznim podatkom se može prikazati jednačinom 3.27.

$$s(t) = (x * \omega)(t) = \sum_{n=-\infty}^{\infty} x(a)\omega(t - a) \quad (3.27)$$

Kada govorimo o multidimenzionalnim nizovima, broj podataka sa kojima radimo je konačan, stoga se konvolucija nad dvodimenzionalnim ulaznim podatkom korišćenjem kernela dimenzije ( $n \times m$ ) može prikazati jednačinom 3.28.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (3.28)$$

Konvolucija je kumulativna operacija, tako da se jednačina 3.28 može napisati i u formi jednačine 3.29. Ova jednačina se najčešće koristi za implementaciju biblioteka mašinskog učenja zato što ima manje varijacija u opsegu validnih rešenja za  $m$  i  $n$ .

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (3.29)$$

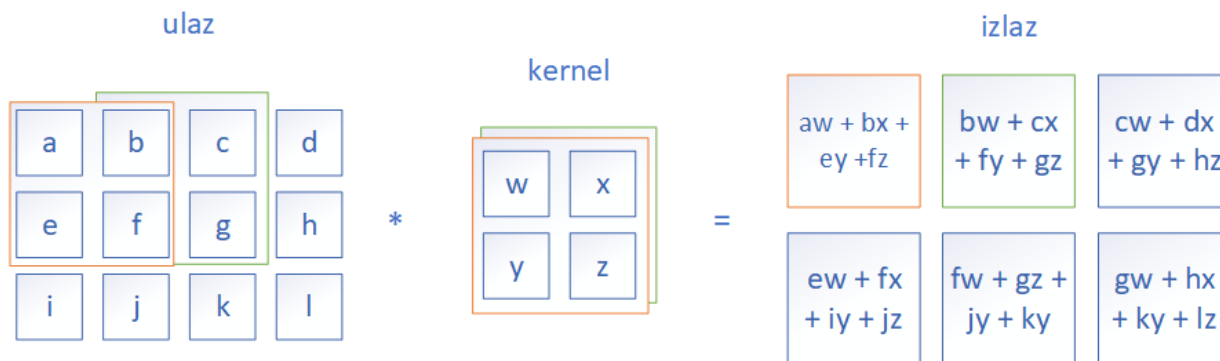
Takođe, u praksi se često koristi i kros-korelaciona funkcija, koja je identična konvolucionoj funkciji, osim što je kernel izokrenut u odnosu na ulaz, jednačina 3.30. Zapravo, mnoge biblioteke mašinskog učenja implementiraju kros-korelaciju, ali je nazivaju konvolucijom.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (3.30)$$

Konvolucione neuralne mreže koriste tri važna principa koji ih diferenciraju od tradicionalnih neuralnih mreža: proređene interakcije, deljenje parametara i ekvivalentne reprezentacije. Kod tradicionalnih neuralnih mreža slojevi koriste množenje matrica parametara sa posebnim parametrima

koji opisuju interakciju između svakog ulaza i izlaza nekog čvora. Sa druge strane, kod konvolucionih mreža se koriste ređe interakcije. Kernel se bira tako da bude manji od ulaznog signala. Ukoliko procesiramo sliku, ulazni podatak može imati hiljade ili milione piksela, ali mi možemo detektovati male značajne osobine kao što su na primer ivice sa kernelima koji se sastoje od svega nekoliko desetina ili stotina piksela. Ovo znači da treba da sačuvamo manje parametara, što dovodi do redukcije u zahtevima za memorijom za čuvanje modela, kao i do povećanja statističke efikasnosti modela. Ovo takođe znači da izračunavanje izlaza zahteva manji broj operacija. Ova poboljšanja u efikasnosti su veoma značajna.

Na slici 3.10. prikazana je 2D konvolucija. Izlaz je ograničen na pozicije gde kernel leži u potpunosti unutar slike, što se još naziva i „validna“ konvolucija.



Slika 3.10 Primer 2D konvolucije, ulazni podatak je  $(3 \times 4)$ , a kernel  $(2 \times 2)$ .

Deljenje parametara se odnosi na to da se isti parametar koristi za više od jedne funkcije u modelu. U tradicionalnim neuralnim mrežama, svaki element težinske matrice se koristi tačno jednom za računanje izlaza iz sloja. Za razliku od toga, u konvolucionim neuralnim mrežama, svaki član kernela se koristi na svakoj poziciji ulaza, osim možda nekih krajnjih piksela na slici, zavisno od odluka o dizajnu vezano za granične piksele. Deljenje parametara znači da, umesto da se uče posebni setovi parametara za svaku lokaciju, ovde se uči samo jedan set parametara. Ovo i dalje ne utiče na vreme procesiranja,  $O(k \times n)$ , ali redukuje zahteve za memorijskim prostorom modela na  $k$  parametara. Pri čemu je  $k$  nekoliko redova veličine manje od  $m$ . Iz ovog razloga konvolucija je značajno efikasnija od gustih matricnih množenja u smislu memorijskih zahteva i statističke efikasnosti.

Tipičan sloj konvolucione mreže se sastoji iz tri faze. U prvoj fazi, nad slojem konvolucione neuralne mreže izvršavaju se konvolucije u paraleli i stvara se set linearnih aktivacija. U drugoj fazi, svaka linearna aktivacija prolazi kroz nelinearnu aktivacionu funkciju, npr. ReLU. Ova faza se naziva i faza detektora. U trećoj fazi, koristi se funkcija udruživanja (eng. *pooling*) kako bi se dalje modifikovao izlaz sloja.

Funkcija udruživanja zamenjuje izlaz mreže na određenim lokacijama sa zbirnom statistikom bliskih izlaza. Na primer, za udruživanje maksimizacijom operacija vraća maksimalni izlaz u okviru određenog broja suseda, uokvirenog pravougaonikom.

### 3.4. Evaluacija modela mašinskog učenja

U mašinskom učenju, cilj je postići dobru generalizaciju modela, odnosno da se model uspešno izvršava nad podacima koji nisu viđeni ranije, tokom postupka treniranja modela. U protivnom, ukoliko je trenirani model previše dobro prilagođen podacima kojima je treniran, može doći do stanja



previše prilagođenog modela, eng. *overfitting*, [55-58], što znači da trenirani model neće imati dobar učinak nad potpuno novim ulaznim podacima.

Jedan od načina da se spreči *overfitting* je tehnika ranog zaustavljanja, eng. *early stopping*, [58-60], kojom se ograničava broj epoha za treniranje neuralnih mreža. Tokom treniranja dubokih neuralnih mreža, jedan ciklus prolaska kroz kompletan set podataka naziva se epoha. Obično se koristi više epoha tokom treniranja dubokih neuralnih mreža. Jedna epoha predstavlja operaciju u kojoj se čitav set podataka prosleđuje na ulaz neuralnoj mreži jednom. Kada se čitav set podataka prosleđuje neuralnoj mreži više puta, odnosno kada se koristi više epoha, veći broj težinskih faktora čvorova neuralnih mreža biva promenjen. Premali broj epoha može dovesti do problema nedovoljnog prilagođenja modela, eng. *underfitting* [61], a preveliki broj epoha može dovesti do previše prilagođenog modela, *overfitting*. Stoga je potrebno, najčešće eksperimentalnim putem, utvrditi optimalni broj epoha kojim se trenira konkretan model duboke neuralne mreže.

Kako bi se postigla raznolikost seta podataka kojima se trenira model, a i u cilju uvećanja seta podataka, pristupa se tehnici uvećanja podataka, eng. *data augmentation*, [62]. Stoga, tehnika uvećanja podataka se koristi sa ciljem poboljšanja performansi modela. Tehnikom uvećanja podataka se veštački kreiraju varijacije nad postojećim skupom podataka. Ukoliko su podaci slike, uvećanje podataka se postiže različitim tehnikama transformacije, zumiranja slike, rotiranja slike za nekoliko stepeni, mirorovanje slike ili isecanje. Primenom tehnike uvećanja podataka postiže se bogatiji, raznovrsniji set podataka koji se dalje mogu koristiti za klasifikaciju slika, detekciju objekata ili semantičku segmentaciju.

### 3.5. Razvoj dubokog učenja

Konvolucione neuralne mreže i propagacija unazad su poznate još od 1989. godine [49], međutim tek od 2012. godine koncept dubokog učenja doživljava konkretnu primenu i naglu ekspanziju u razvoju novih algoritama i primena [63]. S obzirom da je polje dubokog učenja više eksperimentalnog nego teorijskog aspekta, napredak u razvoju hardvera i kreiranje novih baza podataka doprineli su i razvoju novih algoritama dubokog učenja.

Kada je reč o napretku u razvoju hardvera, centralne procesorske jedinice (CPU) su od 1990. do 2010. postale 5000 puta brže. Dalje, tokom 2000-tih godina je došlo do razvoja brzih čipova sa velikom mogućnošću paralelizacije procesa, grafičke procesorske jedinice (GPU), kako bi se postigla bolja grafika za video igre. Ovo je omogućilo renderovanje kompleksnih 3D scena u realnom vremenu. Rad na razvoju hardvera za industriju video igara direktno je doprineo razvoju mašinskog učenja i dubokog učenja. Duboke neuralne mreže koje se sastoje od velikog broja matričnog množenja se mogu veoma efikasno paralelizovati. Današnji GPU-ovi, kao npr. NVIDIA TITAN X mogu da dostignu i do 6,6 TFLOPS-ova, tj. 6,6 triliona *float32* operacija u sekundi. Trenutni razvoj hardvera za duboko učenje ide i izvan grafičkih procesorskih jedinica. Na primer, kompanija Google je razvila tenzorsku procesorsku jedinicu (eng. *Tensor Processor Unit* – TPU), novi čip dizajn na kome se duboke neuralne mreže mogu pokretati 10 puta brže sa značajno boljom energetsom efikasnošću u odnosu na trenutno najbolje GPU jedinice.

Razvojem interneta dolazi se do bolje dostupnosti podataka i razvoja novih baza podataka za duboko učenje. Razvoj interneta je omogućio da se skupe i distribuiraju veoma veliki skupovi podataka, baze podataka slika, video snimaka, kao i prirodnog jezika. Jedna od osnovnih baza podataka koja je doprinela naglom razvoju dubokog učenja je ImageNet baza podataka [64], koja se sastoji od 1,4 miliona slika koje su ručno označene, tj. anotirane, u 1000 kategorija. Značajan doprinos razvoju konvolucionih neuralnih mreža i arhitektura dubokog učenja je nastao kao rezultat „ImageNet Large Scale Visual Recognition“ godišnjeg takmičenja koje se organizuje počevši od 2010. godine [65]. Arhitekture dubokih neuralnih mreža koje su pobeđivale tokom godina na ovom takmičenju

predstavljaju poslednju reč tehnologije razvoja neuralnih mreža i dubokog učenja, pri čemu su ove arhitekture postale gradivni blokovi i inspiracija za nova rešenja. Neke od najpoznatijih arhitektura dubokih neuralnih mreža koje su pobeđivale na ImageNet takmičenju su AlexNet razvijen od strane „SuperVision“ grupe sa Univerziteta u Torontu [63], VGG-16 model razvijen od strane tima VGG (Visual Geometry Group) sa Univerziteta Oksford [9], GoogLeNet razvijen od strane istraživača iz kompanije Google [10] koji uvode uvodne module (eng. *inception moduls*), Residual Neural Network (ResNet) razvijen od strane istraživača iz kompanije Microsoft [11], gde ova neuralna mreža ima čak 152 sloja, i ReNet razvijena od strane istraživača iz Politehnike Milano i Univerziteta u Montrealu [66]. Neka od novih otkrića u oblasti dubokog učenja su automatizovano mašinsko učenje [67], treniranje dubokih neuralnih mreža sa sintetičkim podacima [68], sinteza video-na-video [69], igranje igre Go gde se koristi učenje podsticanjem (eng. *reinforcement learning*) [70,71] i učenje od-kraja-do-kraja [72–76].

## 4. Primena tehnika dubokog učenja u autonomnim vozilima

Osnove oblasti autonomne vožnje za koje se koriste algoritmi mašinskog učenja, i konkretnije dubokog učenja odgovaraju na sledeća pitanja:

- gde sam ja? – lokalizacija i mapiranje,
- gde i ko su svi drugi? – razumevanje scene,
- kako da stignem od tačke A do tačke B? – planiranje kretanja.

Lokalizacija i mapiranje predstavlja mogućnost vozila da lokalizuje sebe u prostoru. Razumevanje scene se odnosi na opažanje, odnosno percepciju scene oko vozila, detekciju i klasifikaciju objekata i bića na sceni, uključujući saobraćajnu signalizaciju, objekte pored i na putu, pešake, životinje. Planiranje kretanja odgovara na pitanje kako da vozilo dođe od trenutne do željene destinacije, uključujući rutiranje putanje i planiranje trajektorije.

Većina algoritama dubokog učenja koji se koriste u autonomnim vozilima su direktno vezani za rešavanje problema unutar sistema za percepciju. Ovi algoritmi uključuju detekciju i praćenje objekata, fuziju informacija sa senzora, lokalizaciju i mapiranje, semantičku segmentaciju i učenje od-kraja-do-kraja. Na primer, u [77] je opisana tehnika kojom se ulazna slika preslikava na mali broj ključnih elemenata percepcije koji se direktno odnose na uslove na putu, odnosno trenutni saobraćaj. Ovakva reprezentacija pruža skup kompaktnih, ali kompleksnih opisa scene kako bi se jednostavnim kontrolerom omogućilo autonomno kretanje. U gruboj podeli, autonomna vožnja zasnovana na viziji se može podeliti na dve glavne paradigme: posredni pristupi percepciji koji raščlanjavanju čitavu scenu kako bi se donela odluka o vožnji i druga tehnika koja je bazirana na kloniranju ponašanja vozača korišćenjem učenja od-kraja-do-kraja.

Osnovne odlike metoda u okviru sistema za percepciju autonomnih vozila su: čuvanje podataka koji identifikuju skup statičkih objekata na oblasti dela puta koji se posmatra, prikupljanje podataka sa senzora montiranih na vozilu sa scene koja uključuje deo puta po kome se kreće vozilo, određivanje na osnovu prikupljenih podataka sa senzora i sačuvanih podataka da li je objekat na putu statički ili nestatički, praćenje nestatičkih objekata u pokretu, kao i detekcija statičkih objekata [78]. Važan deo sistema za percepciju čine algoritmi za obradu informacija sa senzora i fuzija informacija sa senzora. Sistematski pregled komponenti sistema za percepciju dat je u nekoliko preglednih radova [79-81].

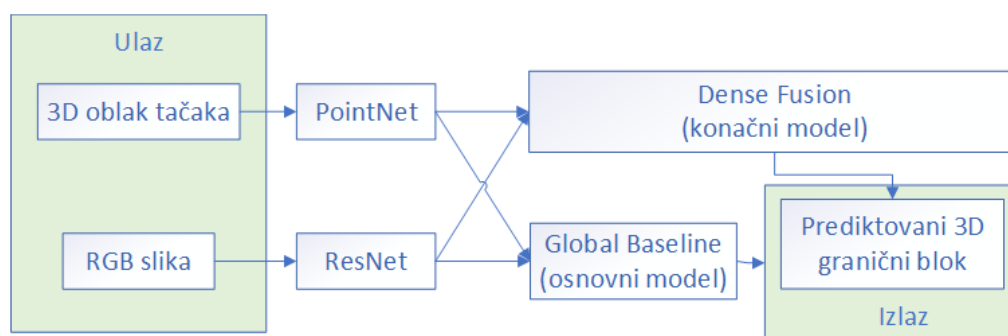
### 4.1. Fuzija informacija sa senzora

Fuzija informacija sa senzora je tehnika u kojoj se kombinuju podaci dobijeni sa različitih izvora kako bi se kreirala koherentna informacija. Informacija dobijena nakon fuzionisanja informacija sa različitih senzora je tačnija nego što bi to bio slučaj da su informacije sa različitih senzora pojedinačno korišćene. Ovo je od posebnog značaja kada se kombinuju informacije sa različitih tipova senzora [25]. Na primer, na autonomnom vozilu je od interesa imati kameru kako bi se klonirala ljudska vizija, odnosno ono što vozač vidi. Međutim, informacija o udaljenosti objekta koji se vidi kamerom je najbolje dobiti pomoću senzora kao što su radar ili lidar. Iz ovog razloga fuzija informacija sa kamere sa informacijama sa radara ili lidara su od velike važnosti, jer su ove informacije komplementarne. Sa druge strane, kombinovanje informacija sa radara i lidara može da omogući pouzdaniju informaciju o udaljenosti objekata ispred vozila ili razdaljinu od objekata u okruženju.

### 4.1.1. 3D detekcija objekata

Trenutni trend u razvoju autonomnih vozila je povećana upotreba lidara. Fuzija informacija sa kamere i lidara predstavlja optimalno rešenje u pogledu kompleksnosti hardvera sistema, jer se samo dva tipa senzora koriste, i pokrivenosti sistema, gde se kamera koristi za viziju, a lidar za detekciju objekata, što su komplementarne osobine. Ovde se slika sa kamere fuzioniše sa 3D oblakom podataka, i kao rezultat 3D kontura pretpostavke i njena verovatnoća su predviđeni. Primer jednog ovakvog rešenja je *PointFusion* mreža [82].

Novi pristup u rešavanju fuzije informacija sa senzora korišćenjem neuralnih mreža podrazumeva da se informacija sa svakog tipa senzora tretira različitom neuralnom mrežom, nakon čega se integrišu rezultujuće reprezentacije u novu neuralnu mrežu gde se vrši fuzija informacija na visokom nivou, slika 4.1. Osnovna prednost ovakvog rešenja je izbegavanje manje tačnih ulaznih predikcija, tako što će se ulazni signali najpre individualno procesirati na niskom nivou, nakon čega se vrši procesiranje na visokom nivou.

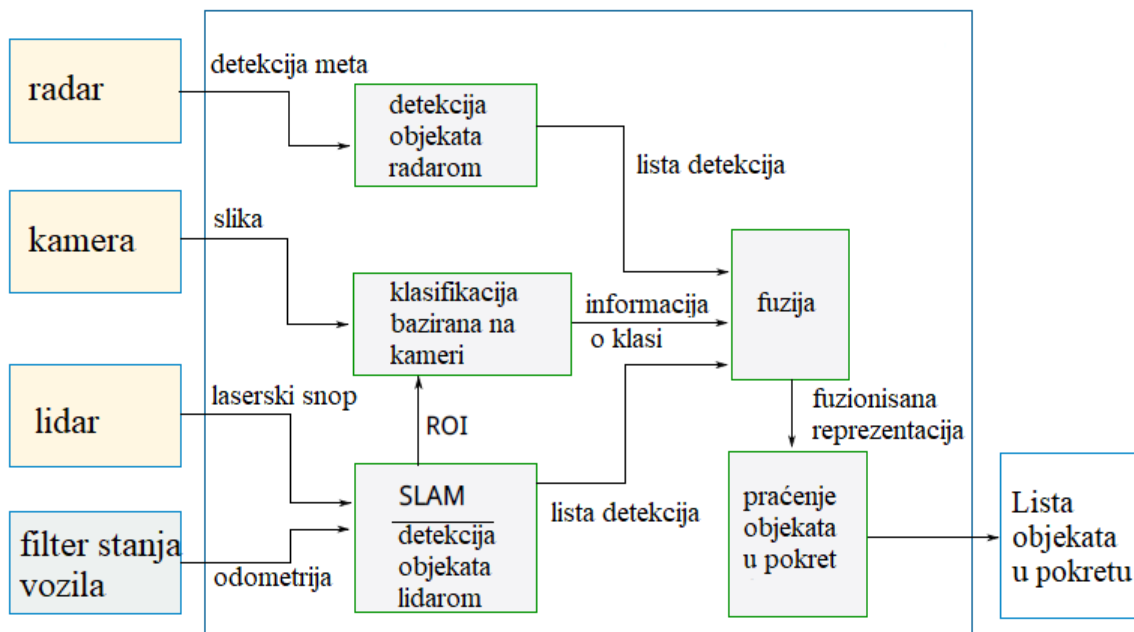


Slika 4.1 Blok dijagram 3D detekcije objekata, slika modifikovana na osnovu [82].

### 4.1.2. Detekcija i praćenje objekata u pokretu

Detekcija i praćenje objekata u pokretu je jedan od najizazovnijih aspekata u domenu autonomnih vozila. Kako je rešavanje ovog problema od velikog značaja za autonomnu vožnju, pouzdanost i performanse rešenja su od izuzetnog značaja. Stoga, za detekciju i praćenje objekata u pokretu koriste se svi postojeći senzori montirani na vozilu. Najčešće se radi o fuzionisanju informacija sa kamere, radara i lidara.

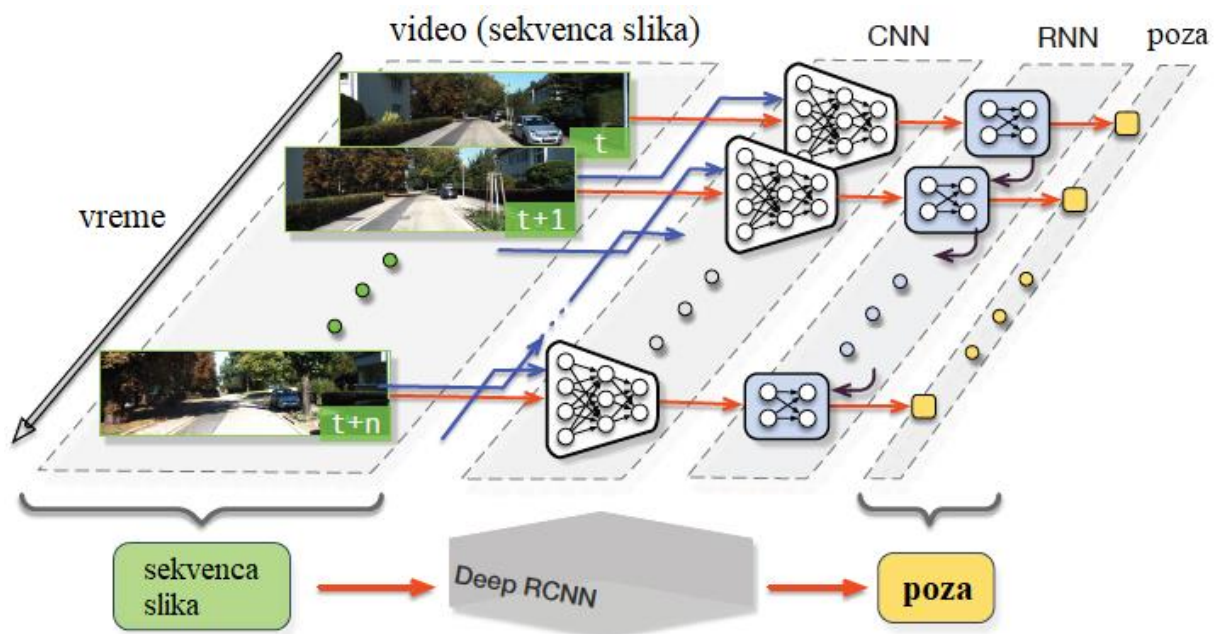
Jedan od pristupa koji se koristi je detekcija objekata u pokretu korišćenjem radara i lidara, nakon čega se region od interesa detektovan lidarom prosleđuje u klasifikator kamere, a zatim se sve informacije zajedno fuzionišu. Informacije dobijene nakon fuzije informacija sa ovih senzora se prosleđuju u blok za praćenje, i dodaju listi objekata koji se kreću. Korišćenjem klasifikacije objekata sa više tipova senzora se povećava tačnost definisanog modela okruženja. Blok dijagram sistema za detekciju i praćenje objekata u pokretu korišćenjem više senzora dat je na slici 4.2, [83].



Slika 4.2 Sistem za percepciju korišćenjem više senzora, slika modifikovana na osnovu [83].

## 4.2. Lokalizacija i mapiranje

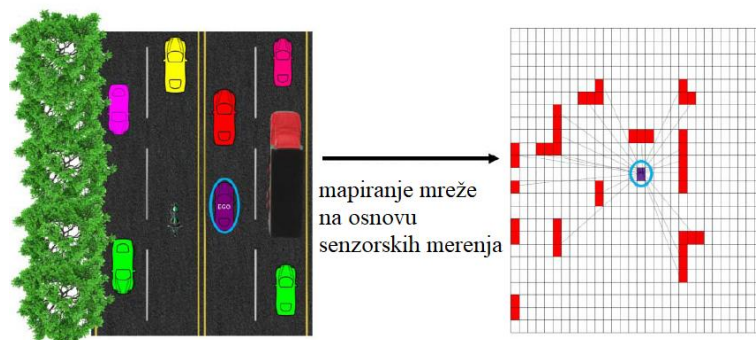
Tradicionalni pristup simultane lokalizacije i mapiranja (eng. *Simultaneous Localization and Mapping* – SLAM) [85], detektuje određene karakteristike u sceni i prati ove karakteristike u vremenu kroz svaki frejm. Iz razlike u pokretu ovih karakteristika SLAM algoritam vrši procenu lokalizacije i orijentaciju vozila, odnosno kamere.



Slika 4.3 Ilustracija DeepVO metode za vizuelnu odometriju od-kraja-do-kraja korišćenjem rekurentnih konvolucionih neuralnih mreža, slika modifikovana na osnovu [85].

Za razliku od tradicionalnog pristupa, od 2015. godine se koristi duboko učenje od-kraja-do-kraja za stereo ili pojedinačne kamere za vizuelnu odometriju baziranu na dubokom učenju. U [85] je opisana metoda gde se uzima vremenska sekvenca slika, nakon čega se svaka od ovih slika dovodi na ulaz konvolucione neuralne mreže čime se ekstraktuju glavne karakteristike svake slike, slika 4.3. Dalje se koristi rekurentna neuralna mreža za praćenje trajektorije kretanja tokom vremena. Ovim se dobija rešenje od-kraja-do-kraja za vezu slika-poza. Korišćenjem dubokog učenja se uklanja više modula tradicionalne SLAM tehnike, jer se primenom modela mašinskog učenja može pronaći rešenje od-kraja-do-kraja za definisanje trajektorije vozila.

Mapiranje zauzetosti mreže se koristi za navigaciju i lokalizaciju autonomnih vozila u dinamičkom okruženju, slika 4.4. Za rešavanje ovog problema se obično koristi fuzija informacija sa kamere i lidara. Koristi se fuzija informacija sa senzora jer su informacije dobijene pomoću kamere i lidara međusobno komplementarne. Kamera vrši akviziciju dvodimenzionalne slike visoke rezolucije u boji, dok se pomoću lidara dobija trodimenzioni oblak tačaka koje predstavljaju objekat na mapi. Uobičajeni pristup pri mapiranju zauzetosti mreže je da se sve mrežne ćelije nezavisno mapiraju. Međutim, novi trendovi vode do toga da se koriste superpikseli u mapi zauzetosti, tako da se ćelije mape koje su zauzete ne preskaču, [86].

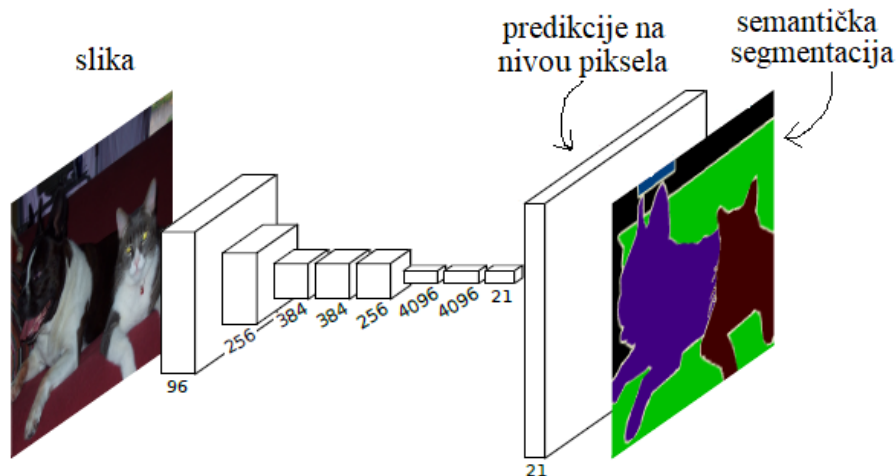


Slika 4.4 Mapiranje zauzetosti mreže, slika preuzeta iz [86].

### 4.3. Semantička segmentacija

Semantička segmentacija predstavlja problem kompjuterske vizije koji podrazumeva uzimanje podataka sa kamere, npr. dvodimenzionalna slika, i njihovo pretvaranje u masku sa regionima od interesa. Semantička segmentacija punog piksela je metoda gde se svakom pikselu na slici dodeljuje indentifikacija klase zavisno od toga kom objektu od interesa pripada. Primenom tradicionalnih metoda kompjuterske vizije je moguće pronaći karakteristike poput ivica, linije i krive, ili gradijenta, ali se tek sa primenom semantičke segmentacije vrši razumevanje slike na nivou piksela. Semantičkom segmentacijom se grupišu slične slike u isti objekat od interesa što je od velikog značaja za zadatak razumevanja scene na delu puta kojim se kreće vozilo u domenu autonomnih vozila.

Primer potpuno konvolucione neuralne mreže koja se koristi za semantičku segmentaciju na nivou piskela dat je na slici 4.5, [87]. Konvolucione neuralne mreže trenirane od-kraja-do-kraja metodom piksel-po-piksel predstavljaju poslednju reč tehnike kada se semantička segmentacija koristi za razumevanje scene [87]- [90].

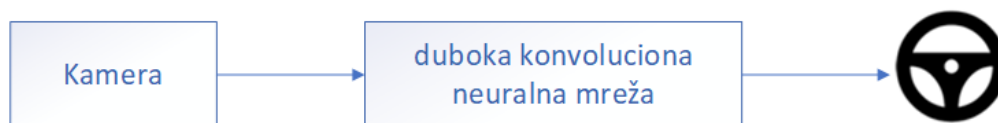


Slika 4.5 Potpuna konvoluciona neuralna mreža za semantičku segmentaciju na nivou piksela, slika modifikovana na osnovu [87].

#### 4.4. Učenje od-kraja-do-kraja ili kloniranje ponašanja vozača

Generalno, svaki sistem koji se zasniva na učenju može se podeliti u više faza za procesiranje podataka. Učenje od-kraja-do-kraja zamenjuje sve ove faze za procesiranje podataka sa samo jednom neuralnom mrežom čiji su ulaz podaci sa senzora a izlaz krajnje komande tražene od strane sistema. Drugim rečima, učenje od-kraja-do-kraja predstavlja proces treniranja neuralne mreže od početka do kraja bez ikakve ljudske interakcije ili uključenja u proces treniranja. Svrha učenja od-kraja-do-kraja je da sistem samostalno nauči interne reprezentacije neophodne za procesiranje, kao što je to na primer detekcija korisnih karakteristika puta u sistemu autonomne vožnje, pri čemu je čitav proces učenja baziran samo na ulaznom signalu.

Duboke neuralne mreže od-kraja-do-kraja za autonomnu vožnju koriste sirove informacije sa senzora koji su montirani na vozilu, obrađuju ih korišćenjem duboke neuralne mreže i kao rezultat ove obrade upravljaju vozilom tako što vrše kontrolu nad nekim od upravljačkih komponenti. Na slici 4.6. prikazan je blok dijagram sistema za autonomnu vožnju korišćenjem učenja od-kraja-do-kraja gde se za ulaz u duboku neuralnu mrežu koristi signal sa kamere, slika, odnosno piksel kao ulazni signal, a izlaz iz duboke neuralne mreže je ugao upravljača. Korišćenje učenja od-kraja-do-kraja u autonomnim vozilima se naziva još i imitaciono učenje ili kloniranje ponašanja vozača.



Slika 4.6 Blok dijagram sistema za autonomnu vožnju korišćenjem učenja od-kraja-do-kraja.

Poznata rešenja za učenje od-kraja-do-kraja za autonomnu vožnju su opisana u [91–95]. Poznata rešenja za autonomnu vožnju koja koriste učenje od-kraja-do-kraja su uglavnom razvijena za komercijalna vozila, gde se model mašinskog učenja koji se koristi u realnom vremenu za zaključivanje nalazi na kompjuteru visokih performansi smeštenom uglavnom u prtljažniku automobila. Do sada realizovana rešenja za autonomnu vožnju korišćenjem učenja od-kraja-do-kraja uglavnom koriste veoma složene arhitekture neuralnih mreža koje su računarski izuzetno zahtevne, na primer primena ResNet50 arhitekture u [93]. Metoda koja predstavlja poslednju reč tehnike kada je u pitanju imitaciono učenje, odnosno kloniranje ponašanja vozača prikazana je u [96]. U ovoj



metodi se primenjuju sintetizovani podaci koji stvaraju situacije poput sudara ili silaska s puta. Umesto da se vrši imitacija svih podataka, ovde se negativni scenariji dodatno kažnjavaju tokom treniranja korišćene neuralne mreže.

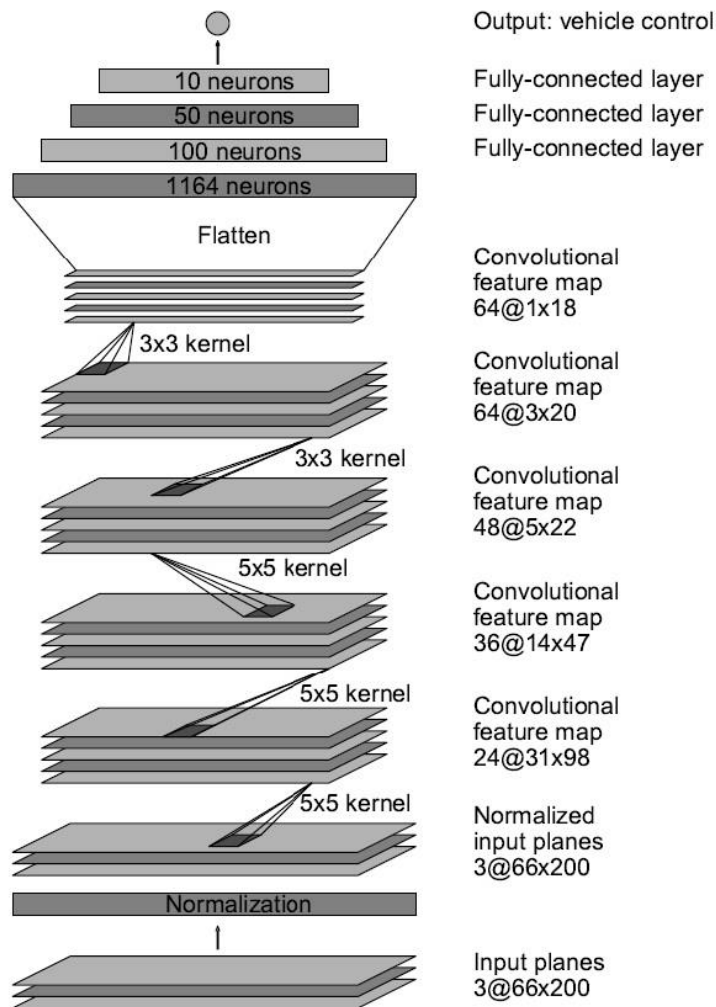
Prethodno razvoju originalne neuralne mreže za autonomno održanje vozila u kolovoznoj traci korišćenjem informacija sa vizuelnih senzora, odnosno za razvoj duboke neuralne mreže za učenje od-kraja-do kraja, koja će biti efikasna sa stanovišta potrošnje energije, korišćenja računarskih resursa procesora na kome se izvršava i smanjenog memorijskog kapaciteta, prikupljeno je najpre reimplementaciji poznatih rešenja za učenje od-kraja-do-kraja. Odabrane su dve mrežne arhitekture za reimplementaciju i poređenje sa novim originalnim rešenjem koje je predmet ovog rada, PilotNet, mrežna arhitektura kreirana sa istovetnom namerom autonomne vožnje i AlexNet, mrežna arhitektura inicijalno kreirana sa ciljem prepoznavanja objekata. Arhitektura AlexNet-a je najpre modifikovana a zatim i implementirana, kako bi ovako izmenjena mreža bila pogodna za autonomnu vožnju učenjem od-kraja-do-kraja.

#### **4.4.1. Primena modifikovane PilotNet arhitekture za učenje od-kraja-do-kraja**

PilotNet, model arhitekture duboke neuralne mreže, koji se takođe naziva i NVIDIA CNN, je mrežna arhitektura razvijena od strane istraživača iz NVIDIA korporacije i predstavlja duboku neuralnu mrežu za autonomnu vožnju korišćenjem učenja od-kraja-do-kraja [91,92]. Arhitektura PilotNet mreže se sastoji od normalizujućeg sloja, praćenog sa pet konvolucionih slojeva, nakon čega slede četiri potpuno povezana sloja. Originalna PilotNet arhitektura prikazana je na slici 4.7.

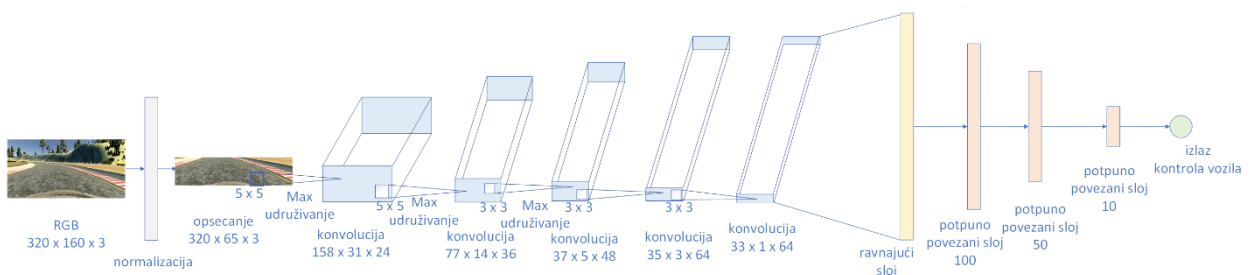
Za potrebe kvalitativnog i kvantitativnog poređenja nove neuralne mrežne arhitekture koja koristi učenje od-kraja-do-kraja, u ovom radu je izvršena modifikacija poznate arhitekture neuralne mreže PilotNet. Razlika između originalne arhitekture PilotNet-a i modifikovane verzije PilotNet arhitekture je pre svega u veličini ulaznog signala, pri čemu se razlike dalje propagiraju kroz sve slojeve duboke neuralne mreže. Sama struktura slojeva i njihov raspored je ostao nepromenjen u odnosu na originalnu arhitekturu. U originalnoj mreži ulazna slika je veličine  $200 \times 66$  piksela, dok je u ovom radu veličina ulaza  $320 \times 160$  piksela, a nakon odsecanja dela slike ulaz u sam model neuralne mreže iznosi  $320 \times 65$  piksela. Ova razlika u ulaznim signalima je dovela i do razlike u ukupnom broju trenerabilnih parametrara mreže. Na primer, u ravnajućem sloju, nakon petog konvolucionog sloja, u originalnoj PilotNet arhitekturi je bilo 1164 neurona, dok je u predloženoj implementaciji 2112 neurona.





Slika 4.7 Originalna PilotNet (naziva se i NVIDIA CNN) arhitektura, slika preuzeta iz [91].

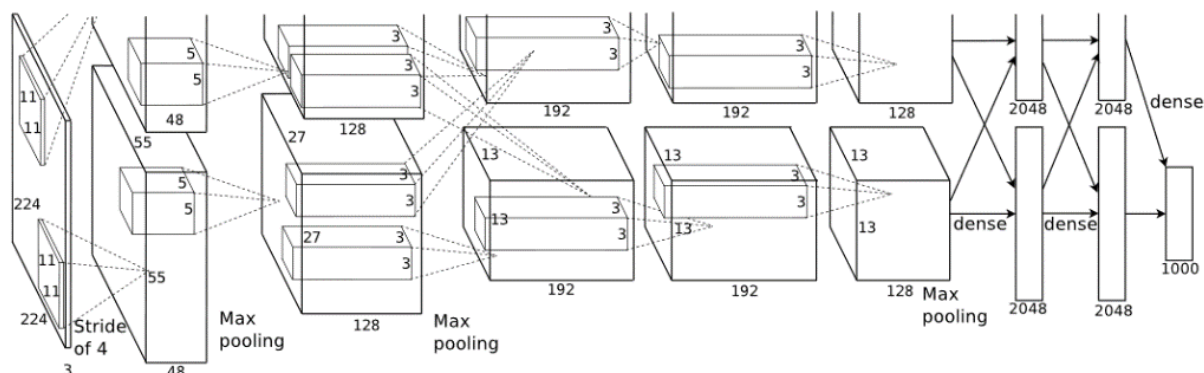
Izuev razlike u broju parametara po svakom sloju, koja je uslovljena razlikama u veličini ulaznog podatka, predložena implementacija PilotNet mrežne arhitekture odgovara originalnom modelu. Arhitektura reimplementirane PilotNet mreže je prikazana na slici 4.8.



Slika 4.8 Arhitektura reimplementirane PilotNet mreže.

#### 4.4.2. Modifikacija i implementacija AlexNet arhitekture za učenje-od-kraja-do-kraja

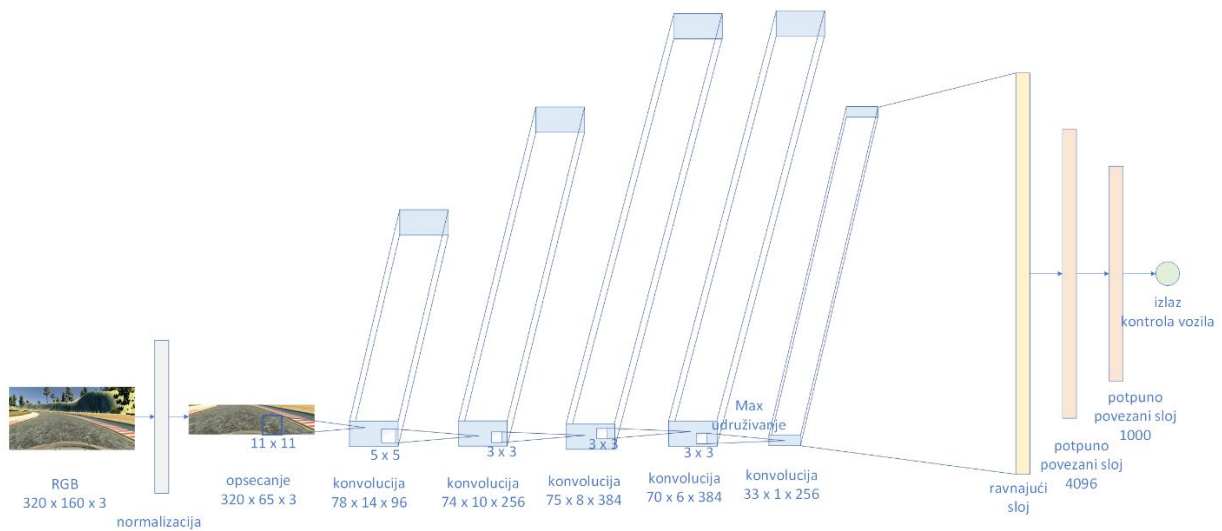
AlexNet je prva poznata višeslojna konvoluciona neuralna mreža [63]. Arhitektura AlexNet mreže se sastoji od pet konvolucionih slojeva praćenih slojevima za udruživanje maksimizacijom, ReLU aktivacionih funkcija, tri potpuno povezana sloja i jednog sloja sa primenom tehnike izbacivanja. Originalna arhitektura AlexNet mreže je prikazana na slici 4.9. Korišćenjem AlexNet mreže su ostvareni najbolji rezultati na *ImageNet* takmičenju ILSVRC-2012 [64]. Projekat *ImageNet* je velika vizuelna baza podataka dizajnirana za upotrebu u istraživanju softvera za vizuelno prepoznavanje objekata.



Slika 4.9 Originalna AlexNet arhitektura, slika preuzeta iz [63].

U ovom radu, AlexNet mrežna arhitektura je modifikovana i reimplemenirana tako da odgovara učenju od-kraja-do-kraja za autonomnu vožnju, slika 4.10. U originalnoj arhitekturi AlexNet mreže postoje dva paralelna procesa. Razlog za ovo je istorijski, prouzrokovan nedovoljnim hardverskim performansama računarskih sistema u to vreme, i rešenje je bilo da se konvolucije izvršavaju u dve paralelne grane na dve nezavisne grafičke procesorske jedinice. S obzirom na to da je na raspolaganju bilo dovoljno hardverskih resursa, za treniranje predložene varijante AlexNet mreže je korišćena jedna GPU jedinica, pa su se i konvolucije odvijale u jednoj grani. Ovo je prva razlika u odnosu na originalnu arhitekturu.

U ovom radu veličina ulaznog signala je  $320 \times 160$  piksela, a nakon odsecanja slike  $320 \times 65$  piksela, dok je u originalnoj AlexNet arhitekturi ulazni signal veličine  $224 \times 224$  piksela. Razlika u veličinama ulaznih slojeva je uticala na to da je bilo neophodno izostaviti tri sloja sa udruživanjem maksimizacijom, s obzirom da nije bilo dovoljno piksela u predloženoj implementaciji nad kojim bi se izvršavalo udruživanje. Izostavljen je prvi sloj udruživanja iz originalne arhitekture i slojevi sa udruživanjem maksimizacijom nakon četvrtog konvolucionog sloja sa 384 odlika i poslednjeg konvolucionog sloja sa 256 odlika.



Slika 4.10 Arihtektura modifikovane i reimplementirane AlexNet mreže.

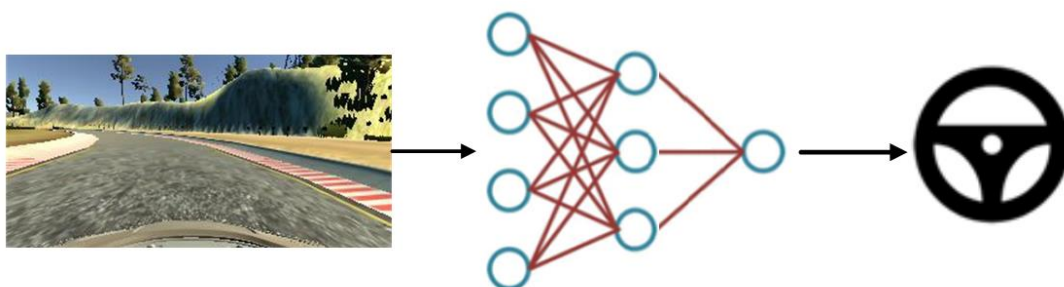
Razlika u veličini ulaznih slojeva je uticala i na razliku u ukupnom broju parametara mreže koji se treniraju, odnosno težinskih faktora između čvorova neuralne mreže. U ovom radu, smanjen je broj parametara za treniranje, tako da originalna AlexNet arhitektura ima 63 miliona parametara a naša implementacija 42 miliona parametara koji se treniraju. Poslednja i važna razlika između originalne arhitekture i naše implementacije je u poslednjem sloju neuralne mreže. Poslednji sloj originalne AlexNet mreže ima 1000 čvorova u izlaznom sloju, s obzirom na to da je zadatak ImageNet takmičenja bio da se klasifikuje 1000 različitih objekata, pa je stoga originalna arhitektura imala 1000 izlaznih čvorova. U našoj aplikaciji, postoji samo jedan izlazni čvor: predikcija ugla okretanja upravljača, odnosno volana.

## 5. Opis problema i metodologije treniranja, verifikacije i evaluacije duboke neuralne mreže

U procesu projektovanja nove arhitekture duboke neuralne mreže korišćen je eksperimentalni pristup. Kako bi se projektovani modeli neuralnih mreža validirali i kako bi se izmerile performanse svake eksperimentalne arhitekture, korišćen je simulator autonomne vožnje. U ovom poglavlju je opisana postavka problema, metodologija postupka za ostvarivanje autonomne vožnje, simulator autonomne vožnje, baza podataka i opis procesa treniranja i evaluacije neuralnih mreža za opisani problem koji se rešava.

### 5.1. Postavka problema

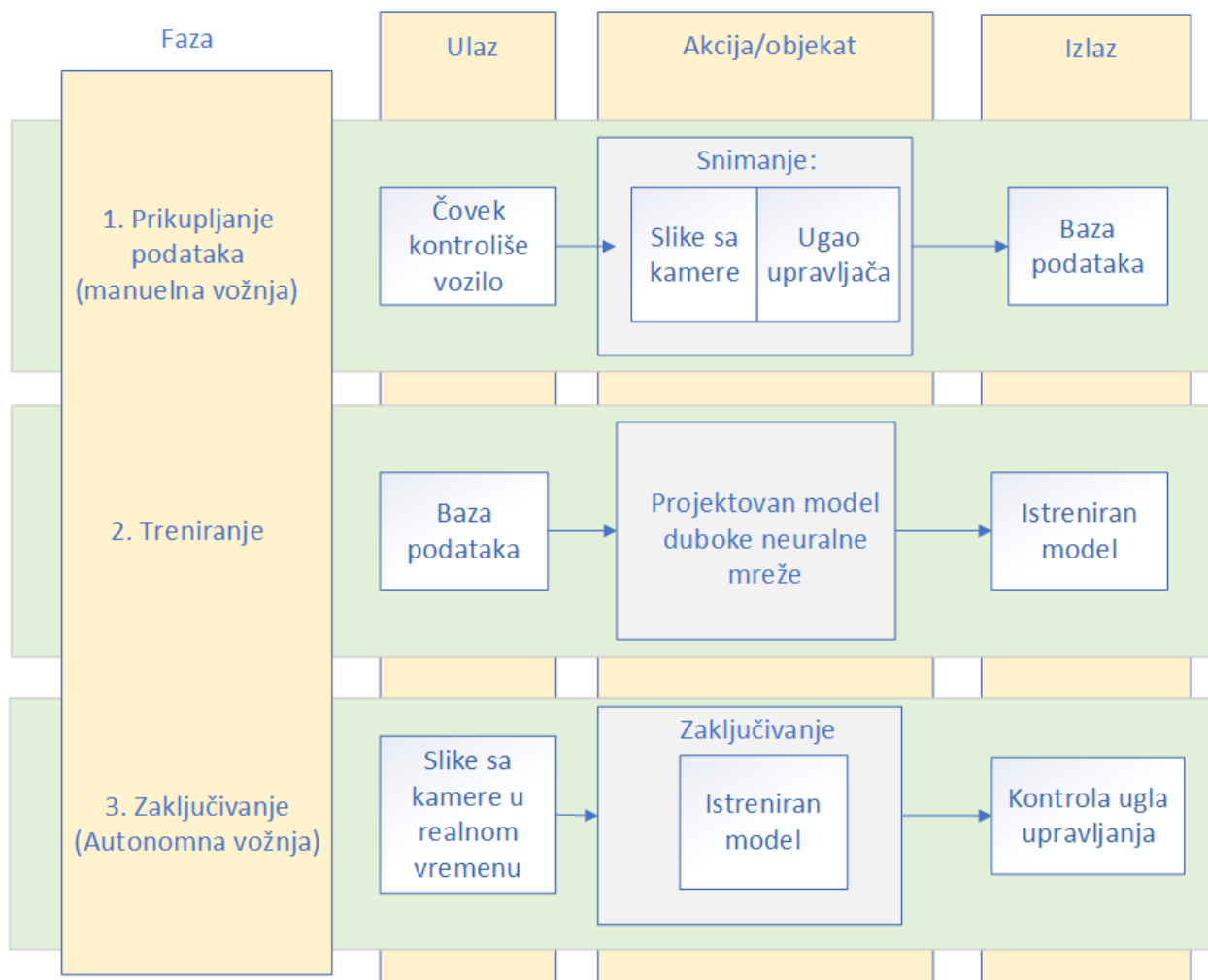
Za autonomno održanje vozila u kolovoznoj traci može se koristiti tehnika učenja od-kraja-do-kraja za autonomnu vožnju. Ulaz u duboku neuralnu mrežu za autonomnu vožnju je slika, pikseli, a izlaz je kontrola vozila, u našem slučaju ugao okretanja upravljača automobila. Učenje od-kraja-do-kraja se primenjuje na ovu mrežu sa ciljem kontrolisanja vozila samo na osnovu ulaznog signala sa kamere tokom rada u realnom vremenu, slika 5.1. Inspiracija za korišćenje samo kamere kao jedinog senzora je bazirana na činjenici da je čoveku kao vozaču čulo vida primarno čulo za upravljanje vozilom. Stoga, osnovna ideja je da se klonira ponašanje čoveka kao vozača i da se samo na osnovu slika sa kamere postavljene na vozilu donosi odluka o pravcu kretanja vozila, odnosno o uglu okretanja upravljača.



*Slika 5.1 Autonomna vožnja u realnom vremenu: na osnovu slike sa kamere koja je ulaz neuralne mreže, određuje se ugao okretanja upravljača.*

Postupak rešavanja opisanog problema se sastoji iz tri faze: prikupljanje podataka kojima će se trenirati isprojektovana neuralna mreža, razvoj neuralne mreže i postupak treniranja, i poslednje, autonomna vožnja kada se u realnom vremenu vrše predikcija, odnosno izdaje komanda za ugao okretanja upravljača samo na osnovu trenutne slike sa kamere. Detaljan blok dijagram svih faza u postupku ostvarivanja autonomne vožnje korišćenjem učenja od-kraja-do-kraja prikazan je na slici 5.2.

Kako bi se izvršila akvizicija podataka potrebnih za treniranje modela duboke neuralne mreže, vozač-čovek upravlja vozilom dok se istovremeno snimaju slike sa kamere i ugao pod kojim je okrenut upravljač. Par slika-ugao prikupljen u datom trenutku predstavlja jedan podatak. Autonomna vožnja se može vršiti u simulatoru autonomne vožnje ili u realnim uslovima. Simulator koji je korišćen u ovom radu je simulator razvijen u Unity okruženju od strane kompanije Udacity, Inc. [97]. U slučaju korišćenja simulatora autonomne vožnje, vozilom se upravlja u manuelnom (trening) režimu vožnje od strane čoveka tako što koristi tastaturu, miša ili džojstik kao upravljač za vozilo, pri čemu se baza podataka automatski kreira.



Slika 5.2 Blok dijagram postupka za ostvarivanje autonomne vožnje.

Podaci akvizirani tokom manualnog režima vožnje su slike sa kamere i ugao okretanja upravljača za svaki frejm. Slike se koriste kao set karakteristika,  $X$ , a merenja ugla okretanja upravljača kao set labela,  $Y$ . Brzina vozila je fiksirana radi pojednostavljenja problema. Podaci prikupljeni korišćenjem ovog pristupa su upotrebljeni za treniranje neuralne mreže koja uči da vozi bazirano samo na ulaznim podacima, bez ikakve ljudske interakcije. Ova tehnika se stoga naziva i kloniranje ponašanja vozača, ili imitaciono učenje. Duboka neuralna mreža za autonomnu vožnju trenirana na ovoj bazi podataka predviđa ugao okretanja volana. Trenirani model se koristi u realnom vremenu za autonomnu vožnju u istom okruženju simulatora. Ocena uspešne vožnje na reprezentativnoj putanji je održanje vozila u kolovoznoj traci tokom čitavog toka autonomne vožnje.

## 5.2. Opis simulatora autonomne vožnje

Platforma koja je korišćena za kolekciju baze podataka kojom se trenira model neuralne mreže, kao i za samu autonomnu vožnju gde su evaluirane performanse implementiranih modela neuralnih mreža za učenje od-kraja-do-kraja za autonomnu vožnju je simulator autonomnog vozila, softver sa otvorenim kodom kreiran od strane kompanije Udacity, Inc. u Unity razvojnom okruženju [97].

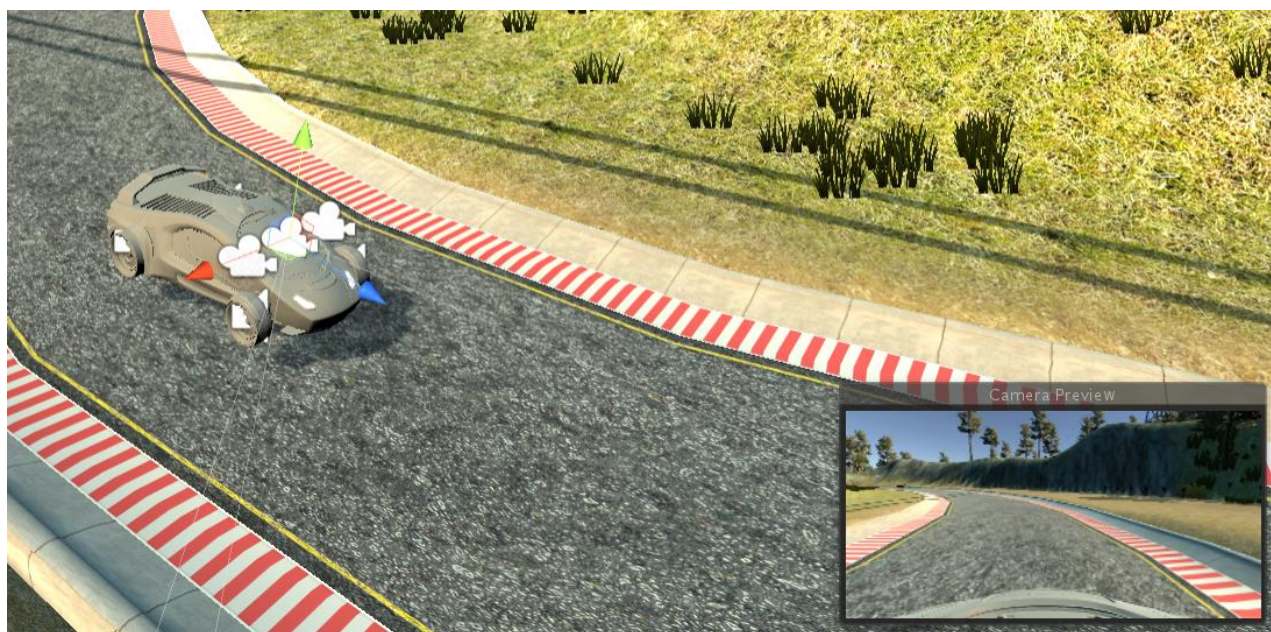
Reprezentativna putanja za autonomnu vožnju se koristi za kreiranje baze podataka. Izgled putanje za autonomnu vožnju u simulatoru, pogled odozgo, prikazan je na slici 5.3.





*Slika 5.3 Simulator autonomne vožnje, pogled iz vazduha na scenu u Unity okruženju [89].*

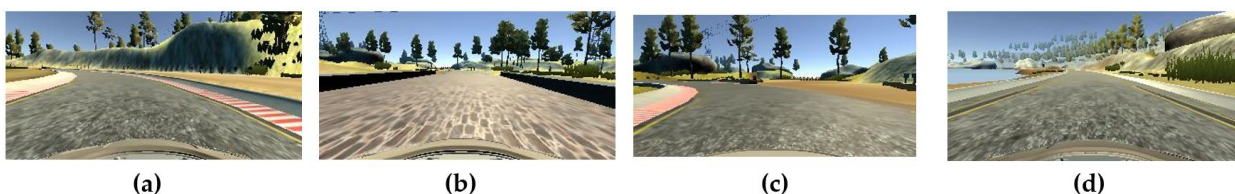
Dok se vozilo kreće u manuelnom režimu rada, slike sa kamera postavljenih na krov vozila koje istovremeno snimaju put ispred vozila uparene sa uglom pod kojim je okrenut upravljač u tom trenutku, slika 5.4. Ovaj par podataka slika-ugao se snima u određen direktorijum. U svakom frejmu se snimaju podaci sa sve tri kamere uparene sa uglom okretanja upravljača za taj frejm. Ista reprezentativna putanja je korišćena za vožnju u autonomnom režimu rada simulatora, gde se odluka o uglu okretanja upravljača donosi u realnom vremenu samo na osnovu slike sa kamere u tom trenutku.



*Slika 5.4 Vozilo u okruženju simulatora na reprezentativnoj kolovoznoj traci, sa tri kamere postavljene na vrhu vozila. U donjem desnom uglu slike je prikaz slike koju vidi centralna kamera na vozilu.*

Različite odlike reprezentativne putanje predstavljaju izazov za treniranje mreže za učenje od kraja-do-kraja za autonomnu vožnju. Na primer, model mora da nauči kako da se ponaša u ostrim krivinama, na različitim teksturama puta, na putu sa različitim ivicama i graničnicima. Najzahtevniji deo ove reprezentativne putanje su tri krivine pod pravim uglom. Ove krivine se nalaze odmah nakon prelaska mosta, što se može videti na slici 5.3. Kolovozna traka je definisana sa crveno-belom trakama, ivičnjacima, ili samo prašinom između trake i ostatka okruženja simulatora.

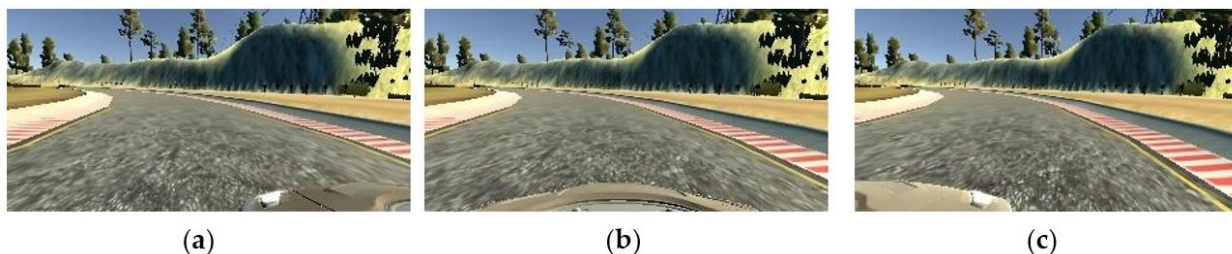
Primeri slika koje se prikupljaju u postupku kreiranja baze podataka dati su na slici 5.5. Deo puta gde je kolovozna traka definisana sa crveno-belim prugama prikazana je na slici 5.5(a). Različite teksture na putu se mogu videti na slici 5.5(b) koja predstavlja most preko jezera. Na samom mostu kolovozna traka je prekrivena kockama, dok se na drugim delovima kolovozne trake nalazi asfalt. Granica puta na mostu je nizak zid. Deo puta oivičen samo prašinom (na jednoj strani) i deo oivičen bankinama prikazan je na slici 5.5(c) i 5.5(d), respektivno. Ove različite karakteristike puta u okruženju utiču na veću opštost modela koji vodi do uspešne autonomne vožnje u različitim scenarijima.



*Slika 5.5 Ulazni podaci snimljeni na različitim delovima kolovozne trake korišćenjem centralne kamere postavljene na vozilu.*

### 5.3. Kreiranje baze podataka

Prikupljanje podataka se vrši u manuelnom režimu vožnje u simulatoru. Slike se akviziraju sa sve tri kamere montirane na vozilu dok se vozilo kreće reprezentativnom putanjom, odnosno kolovoznom trakom, slika 5.4. Na kraju vožnje u manuelnom režimu, slike se snimaju zajedno sa tabelom u kojoj su uparene informacije slika-ugao okretanja upravljača za svaki frejm. Primer slika koje su prikupljene u jednom frejmu simultano sa tri kamere montirane na vozilu je prikazana na slici 5.6. Tri kamere se koriste za potrebe treniranja. Tokom postupka prikupljanja podataka, u svakom frejmu, slike sa sve tri kamere su uparene sa identičnim uglom okretanja volana. Mala razlika u uglu vidljivosti ovih slika za sve tri kamere, dovodi do veće generalizacije modela, što je osnovni motiv za korišćenje slike sa tri kamere u postupku treniranja neuralne mreže. Kasnije, u fazi zaključivanja, tokom same autonomne vožnje, koristiće se samo slika sa centralne kamere.



*Slika 5.6 Primer akvizicije podataka prikupljenih istovremeno u jednom frejmu sa tri kamere montirane na vozilu: (a) leva; (b) centralna; (c) desna kamera.*



Kako bi se prikupili kvalitetni uzorci, koji će omogućiti modelu da nauči tačne odlike, vozilo se kreće u manuelnom režimu rada na isti način kako se očekuje da vozilo kreće u režimu autonomne vožnje. Tokom prikupljanja podataka, cilj je bio da se vozilo kreće sredinom kolovozne trake. Posebna pažnja je posvećena vožnji u krivinama pod pravim uglom. Od izuzetnog značaja je da model nauči kako da se ponaša u krivinama.

### 5.3.1. Uvećanje broja podataka

Tokom postupka prikupljanja podataka, nekoliko krugova vožnje u manuelnom režimu je snimljeno, pri čemu su prikupljeni podaci sa sve tri kamere. Međutim, čak i nakon nekoliko krugova, prikupljena baza podataka nije sadržala dovoljno uzoraka. Iz tog razloga, pristupilo se tehničari za uvećanje broja podataka, eng. *data augmentation*. Jedna od najvažnijih karakteristika reprezentativne putanje su krivine. Jednostavan metod za dupliranje baze podataka sa stavljanjem fokusa na krivine je bilo uvećanje podataka tako što su slike okrenute po vertikalnoj osi, a merenje ugla okretanja upravljača je pomnoženo sa brojem  $-1$ .

Uz dupliranje broja uzoraka korišćenih za treniranje, uvećanjem broja podataka okretanjem slike je obezbeđena dodatna vrednost konačnom modelu. Kako je reprezentativna kolovozna traka zatvorena putanja, mnoge krivine su u istom smeru. U ovom radu, većina krivina je bila na levo. Ukoliko bismo koristili samo podatke koje smo prikupili na ovakvoj putanji, model bi naučio da okreće u većini slučajeva, iako bi tačna informacija bila pravo. Uvećanje podataka manipulacijom nad slikom, u ovom radu okretanjem slike po vertikalnoj osi, i invertovanjem vrednosti ugla okretanja upravljača, obezbeđuje balansiranu bazu podataka pri čemu je modelu omogućeno da jednako nauči ponašanje u krivinama u levo i u desno.

### 5.3.2. Baza podataka

Ukupan broj akviziranih uzoraka je 34.290 slika rezolucije  $320 \times 160 \times 3$ . Broj podataka u kreiranoj bazi predstavlja slike dobijene sa sve tri kamere, odnosno prikupljene su informacije u 11,430 vremenskih trenutaka. Svaka snimljena slika ima 320 piksela visine, 160 piksela širine i tri kanala dubine – RGB slika u boji. Prosečna veličina memorije svake slike je oko 13,5 KB. Svaka slika je uparena sa odgovarajućom vrednošću ugla okretanja upravljača koji je normalizovan u opsegu između  $-1$  i  $1$ . Nakon primene tehnike uvećanja podataka, ukupan broj uzoraka u bazi podataka je 68.580. Podaci su podeljeni na set za treniranje i set za validaciju, pri čemu je 80% podataka odabrano za trening, 54.864 uzoraka i 20% podataka za validaciju, 13.716 uzoraka, kao što je prikazano u tabeli 5.1.

**Tabela 5.1** Baza podataka parova slika-ugao za autonomnu vožnju u simulatoru.

	Treniranje	Validacija	Ukupno
Broj uzoraka	54.864	13.716	68.580
Procenat od ukupne baze podataka	80%	20%	100%

Testiranje je izvršeno tokom faze zaključivanja u realnom vremenu u režimu autonomne vožnje. U fazi zaključivanja, slike samo sa centralne kamere koje se akviziraju u realnim vremenu se kontinuirano šalju na ulaz istreniranog modela koji se koristi za kontrolu ugla okretanja upravljača. Evaluacija uspešne autonomne vožnje je izvršena posmatranjem da li vozilo uspešno vrši održanje u kolovoznoj traci, odnosno ne skreće van predefinisane putanje. Ukoliko vozilo skrene van puta, smatra se da nije postignuta uspešna autonomna vožnja.



### 5.3.3. Pretprocesiranje podataka

Za treniranje duboke neuralne mreže, vrši se akvizicija slika sa sve tri kamere – centralne, leve i desne, slika 5.6. Upotreba tri kamere za prikupljanje informacija za treniranje duboke neuralne mreže za autonomnu vožnju primenjeno je i u drugim radovima [91]. Sve tri slike su snimljene u sličnoj sceni simulatora, ali snimljene sa neznatno različitim pozicija. Prednost korišćenja tri kamere umesto jedne centralne kamere je što se dobija tri puta više podataka za jedan frejm, kao i bolje performanse za upravljanje vozilom kada je potrebno da se ono vrati na centar kolovozne trake. Kako je ugao okretanja upravljača uparen tačno sa centralnom kamerom, neophodno je postaviti ofset za levu i desnu kameru. Ovaj korekcionni faktor se takođe može smatrati hiperparametrom koji se može podešavati u procesu treniranja.

Odsecanje delova ulazne slike je izvršeno kako bi se uklonili delovi slike koji nemaju vredne informacije za autonomnu vožnju. Stoga su uklonjeni delovi slike na kojima se vidi nebo, kao i donji deo slike na kojoj se nalazi prednji deo vozila. Primer slike nakon odsecanja navedenih delova prikazan je na slici 5.7.



Slika 5.7 Ulazna slika nakon odsecanja.

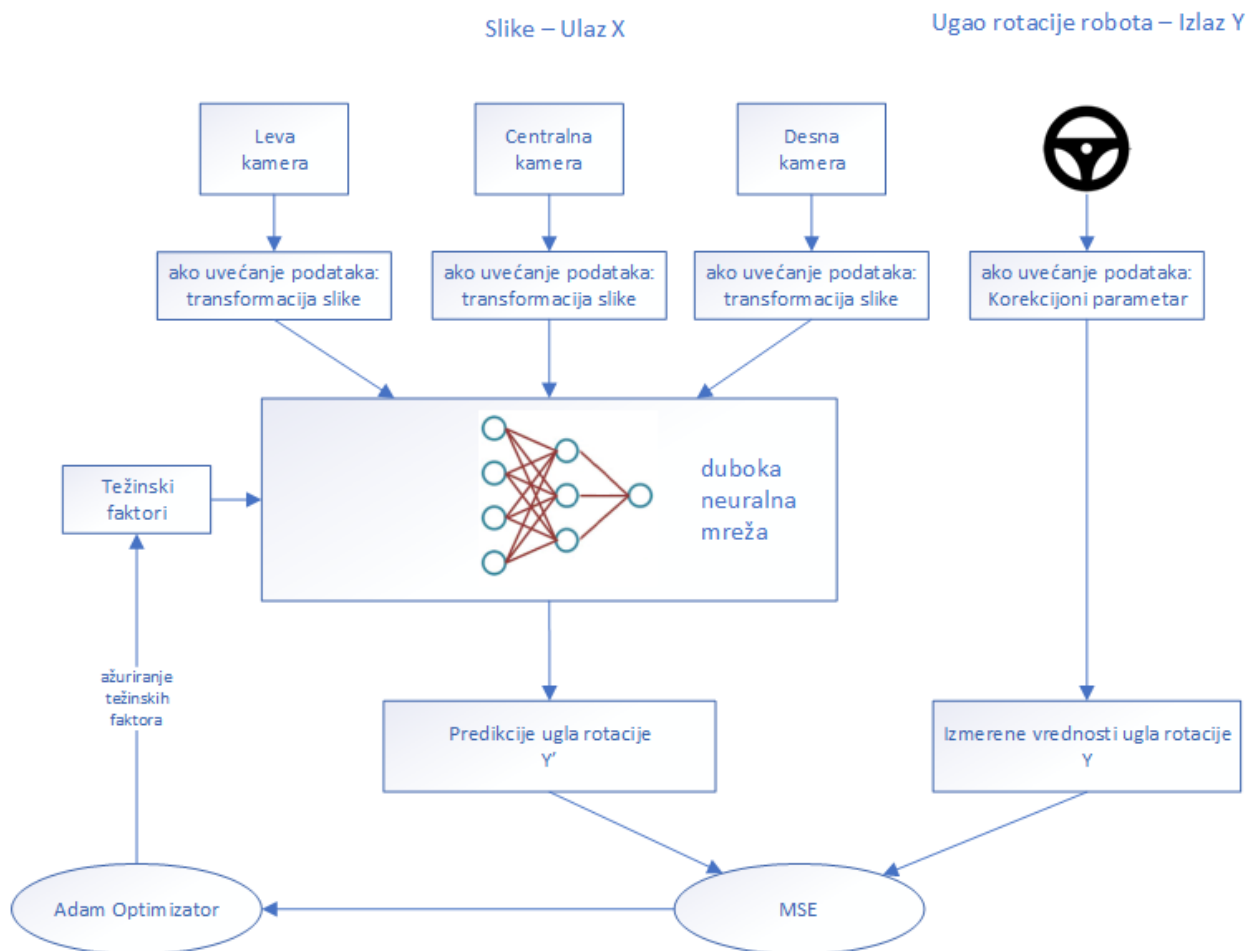
Slike su normalizovane tako što je vrednost svih piksela na slici podeljena sa brojem 255, što je maksimalna vrednost koju piksel na slici može imati. Ovako se vrednosti piksela svake slike normalizuju u opsegu između 0 i 1. Centriranje podataka po sredini se vrši oduzimanjem vrednosti 0.5 za svaki piksel:

$$x_{norm} = \frac{x}{255} - 0.5 \quad (5.1)$$

što rezultira time da se vrednosti svih piksela slike sada nalaze u opsegu  $-0.5 < x_{norm} < 0.5$ .

## 5.4. Treniranje duboke neuralne mreže za učenje od-kraja-do-kraja za autonomno održanje vozila u kolovoznoj traci

Tokom treniranja, slike sa centralne, leve i desne kamere montirane na vozilu se dovode na ulaz modela duboke neuralne mreže, slika 5.8. Korišćenjem tri kamere, triplira se broj podataka za treniranje mreže i pomaže se modelu da bolje upravlja vozilom u slučaju kada ono skrene sa centralne linije kolovozne trake. Slike sa centralne kamere se uzimaju bez izmena, dok se za slike sa leve i desne kamere primenjuje korekcionni parametar na vrednost ugla okretanja upravljača. Nakon finog podešavanja hiperparametara tokom postupka treniranja mreže, vrednost ovog korekcionnog faktora je postavljena na 0.22, vrednost određena eksperimentalnom metodom, koje se dodaje ili oduzima merenjima ugla okretanja upravljača za slike sa leve i desne kamere, respektivno. Dodatno, u pretprocesiranje podataka takođe spada i odsecanje slika i normalizacija i centriranje podatka, pri čemu se koristi Lambda sloj iz Keras biblioteke.



Slika 5.8 Dijagram toka treniranja duboke neuralne mreže za učenje od-kraja-do-kraja.

Svi modeli su trenirani korišćenjem baze podataka koja je prethodno kreirana. Za funkciju gubitka koristi se srednje kvadratna greška (eng. *Mean Squared Error* – MSE) kako bi se minimalizovala greška između predikcija za ugao okretanja upravljača i tačne vrednosti ovog ugla. MSE je funkcija gubitka koja najviše odgovara regresivnim mrežama [45-47]. Ova funkcija gubitka uzima srednju vrednost kvadrata razlike između originalnih vrednosti merenja i prediktovanih vrednosti, jednačina (6.2). Prednost MSE metode je jednostavnije računanje gradijenta. Računanje kvadrata greške dovodi do efekta da su veće greške izraženije od manjih, pa se i sam model fokusira na ispravljanje većih grešaka.

$$MSE = \frac{1}{N} \sum_{j=1}^N (y_j - \hat{y}_j)^2, \quad (5.2)$$

gde je  $\hat{y}_j$  vektor od  $N$  vrednosti predikcija, a  $y_j$  vektor koji predstavlja  $N$  tačnih vrednosti za data merenja.

Tokom procesa treniranja mreže, Adam optimizator [98] je primenjen na sva tri modela. Adam optimizacija je jedna od najefektnijih optimizacionih metoda za treniranje dubokih neuralnih mreža. Empirijski rezultati demonstriraju da Adam optimizacija daje dobre rezultate u praksi i ova optimizacija predstavlja dominantan izbor kod korišćenja metoda stohastičke optimizacije. Sve tri mreže su trenirane korišćenjem MSE funkcije gubitka i Adam optimizatora.

## 5.5. Metode merenja odstupanja vozila od sredine kolovozne trake

Osnovna ocena autonomnog održanja vozila u kolovoznoj korišćenjem modela duboke konvolucione neuralne mreže je uspešna ili neuspešna vožnja. Pod uspešnom vožnjom se podrazumeva da je automobil prešao čitavu putanju bez da je skrenuo sa iste, da došao u polaznu tačku i dalje nastavio da se autonomno kreće istom putanjom. Na slici 5.9 se nalazi prikaz vozila u simulatoru tokom uspešne vožnje.



Slika 5.9 Snimak jednog trenutka uspešne vožnje u simulatoru.

Kao mera kvaliteta uspešne vožnje meri se odstupanje vozila od sredine kolovozne trake, odnosno u simuliranim uslovima, meri se broj piksela od centralne linije vozila do sredine kolovozne trake. Na slici 5.10(a), je prikazano vozilo na kolovoznoj traci gde je crvenim strelicama označeno odstupanje od sredine kolovozne trake, vozilo kreće bliže jednoj liniji puta. Na slici 5.10(b) je prikazan primer dijagrama relativnog odstupanja kretanja vozila od centralne linije na čitavoj putanji – više detalja o merenju relativnog odstupanja biće predstavljeno u poglavlju 7.2.2.



(a)



(b)

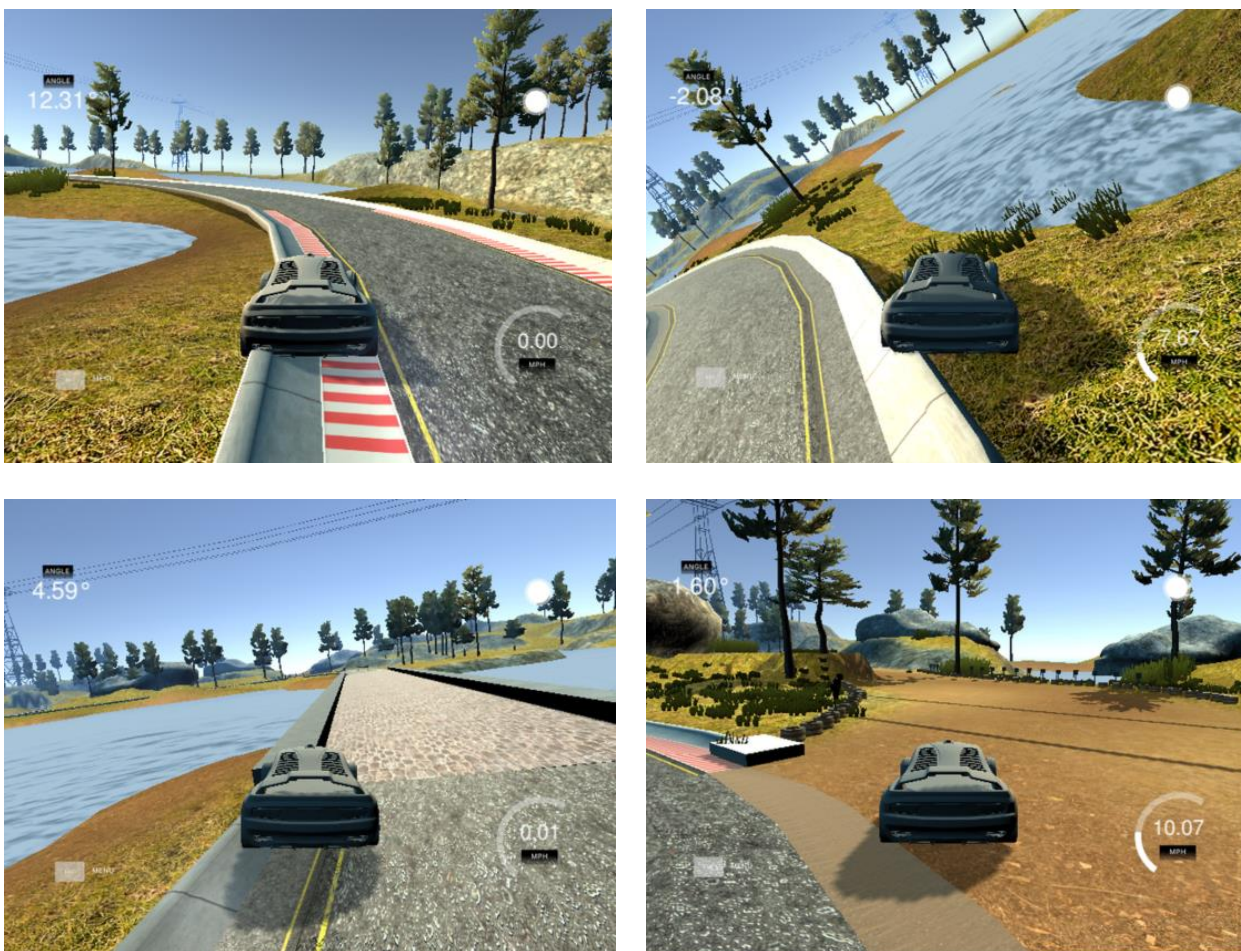
Slika 5.10 Merenje odstupanja vozila od centra kolovozne trake; (a) postoji odstupanje, vozilo se kreće bliže levoj ivici puta, (b) primer dijagrama relativnog odstupanja kretanja vozila od centralne linije na čitavoj putanji.



Kvantitativno merenje odstupanja kretanja vozila od sredine kolovozne trake se vrši računanjem dve greške, srednje kvadratne greške (MSE – *mean square error*) i integralne apsolutne greške (IAE – *integral absolute error*). Srednja kvadratna greška je predstavljena jednačinom 5.2, a apsolutna greška je predstavljena jednačinom 5.3.

$$IAE = \int |y_j - \hat{y}_j| dt. \quad (5.3.)$$

Neuspešnom vožnjom se smatra skretanje vozila sa zadate putanje, što u scenariju korišćenja simulatora znači da je vozilo završilo vožnju u jezeru, šumi, da je udario u bankinu ili ogradu mosta. Na slici 5.11 dati su primeri skretanja vozila sa kolovozne trake nakon kojih je vozilo ili zaustavljeno ili ne može nastaviti vožnju po definisanoj kolovoznoj traci. Bilo koje od ovih skretanja sa definisane putanje znači neuspešnu vožnju i da vozilo nije u stanju da nastavi dalje kretanje po definisanoj putanji.



Slika 5.11 Primeri neuspešne vožnje – skretanje vozila sa kolovozne trake.

## 6. Projektovanje i implementacija nove arhitekture modela neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju

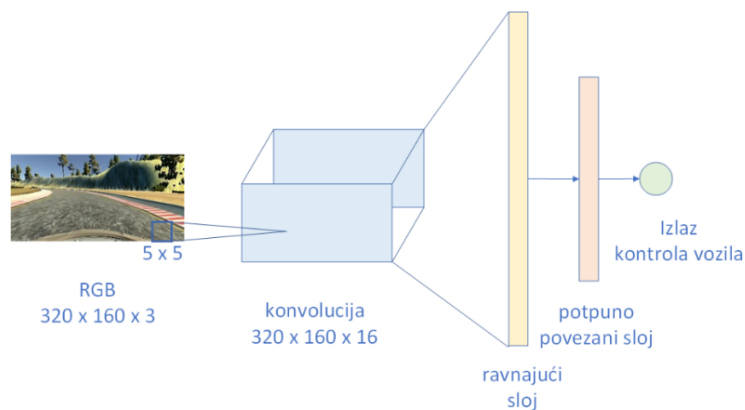
U ovom poglavlju opisana je metodologija projektovanja nove arhitekture modela duboke konvolucione neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju. Okruženje u kome se validiraju modeli mašinskog učenja je simulator autonomne vožnje opisan u prethodnom poglavlju.

### 6.1. Definisane metodologije kreiranja arhitekture duboke neuralne mreže

S obzirom da su duboke neuralne mreže kompleksni sistemi, čije karakteristike i performanse definiše mnoštvo nezavisnih parametara, izabran je pristup parcijalne selekcije parametara kroz hijerarhijsku analizu. Najpre se analiziraju globalne karakteristike arhitekture neuralne mreže koje dominantno čine broj i tipovi slojeva, što ujedno predstavlja i dubinu neuralne mreže. Nakon toga se na sledećem hijerarhijskom nivou analiziraju uticaji veličine samih slojeva. Na kraju se u najnižem hijerarhijskom nivou razmera i uticaj veličine samih konvolucionih kernela.

Osnovna ideja tokom procesa dizajna duboke neuralne mreže je ostvarivanje autonomne vožnje korišćenjem učenja od-kraja-do-kraja tako da se koristi računarski najefikasniji model, dok se istovremeno postiže željena tačnost u autonomnoj vožnji, odnosno uspešno održanje vozila u kolovoznoj traci. Računarski najefikasniji model je obično model sa najmanjim brojem parametara koje je potrebno trenirati, što je direktno u vezi s brojem matematičkih operacija koje je potrebno izvršiti nad čvorovima dizajnirane neuralne mreže. Stoga, vrsta i veličina slojeva neuralne mreže, veličina kernela i broj mapa odlika imaju direktan uticaj na računarsku performansu modela. Sa druge strane, uspešno održanje vozila u kolovoznoj traci se određuje direktno u simulatoru autonomne vožnje, merenjem odstupanja vozila od sredine definisane kolovozne trake tokom vožnje u realnom vremenu.

Kao što je detaljnije objašnjeno u trećem poglavlju, dizajniranje arhitekture dubokih neuralnih mreža i duboko učenje zasniva se više na eksperimentalnom pristupu nego što je to teorijska oblast. Stoga je i konačna arhitektura duboke neuralne mreže kojom rešavamo zadati problem, a koju smo nazvali J-Net, rezultat eksperimentisanja sa gradivnim blokovima neuralnih mreža – različitim brojem slojeva, dubine konvolucionih slojeva, odnosno broja mapa odlika, veličine kernela, umetanja slojeva udruživanja maksimizacijom i, na kraju, eksperimentisanja sa veličinom i brojem potpuno povezanih slojeva.



Slika 6.1 Arhitektura prve eksperimentalne duboke neuralne mreže.

Blok dijagram prve eksperimentalne arhitekture kojom je pokušano rešavanje problema automatskog održanja vozila u kolovoznoj traci dat je na slici 6.1. Arhitektura prve eksperimentalne duboke neuralne mreže je jednostavna duboka neuralna mreža koja se sastoji od samo jednog konvolucionog sloja sa 16 mapa odlika, jednim ravnajućim slojem, i jednim potpuno povezanim slojem.

Prvi korak u dizajnu novog rešenja bio je da se koristi jednostavna konvoluciona neuralna mreža. Primenjena je operacija dvodimenzione konvolucije nad sirovim ulaznim podacima – slici, odnosno nad pikselima:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (6.1)$$

gde ulazna slika  $I$  ima tri kanala, RGB, i dimenzije dimenzijama  $320 \times 160$  piksela. Korišćen je kernel veličine  $5 \times 5$ . Kernel se koristi da se izdvoje delovi slike u operaciji konvolucije. Izlaz iz operacije konvolucije je  $S(i, j)$ , dvadimenzioni tenzor mape odlika. Težinski faktori  $w$ , se dele kroz sve delove kernela za dati sloj u konvolucionoj neuralnoj mreži kako bi se detektovala određena reprezentacije bez obzira na to na kom delu slike se ona nalazi.

Jednačina za računanje visine i dužine konvolucionog izlaznog sloja je:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

$$H_{out} = \frac{H - F + 2P}{S} + 1 \quad (6.2)$$

gde su  $W$  i  $H$  dužina i visina ulaznog sloja,  $F$  je veličina kernela,  $S$  je korak,  $P$  je ispunjavanje, i  $K$  je broj filtera. U našem eksperimentu, ulazna slika je veličine  $320 \times 160$  piksela,  $F = 2$ , korak je 1, i nismo koristili ispunjavanje,  $P = 0$ . Ovo dovodi do veličine izlaznog sloja:  $W_{out} = 319$ , i  $H_{out} = 159$ . Dubina izlaznog sloja je jednaka broju filtera;  $D_{out} = 16$  u ovom slučaju. Izlazni volumen nakon operacije konvolucije je:

$$W_{out} * H_{out} * D_{out} \quad (6.3)$$

što iznosi 811.536. Nakon konvolucije, *ReLU* aktivaciona funkcija [99] je primenjena:

$$ReLU(x) = \max(0, x). \quad (6.4)$$

Izvršena konvolucija je praćena sa ravnajućim slojem, pri čemu se dvodimenzionalna mapa odlika konvertuje u jednodimenzioni vektor. Ravnajući sloj ne rezultira novim parametrima za treniranje, s obzirom da imamo konverziju čvorova iz prethodnog sloja u jednu dimenziju, bez ikakvih matematičkih operacija nad težinskim faktorima ovih čvorova. Kada postoje prostorne odlike slike kao rezultat konvolucije, primenjuje se potpuno povezani sloj gde se svi čvorovi iz ravnajućeg sloja kombinuju u jedan jedinstveni izlaz koji direktno predviđa vrednosti ugla okretanja upravljača. Kako postoji samo jedan izlaz ovakve neuralne mreže, u pitanju je regresivna mreža. Ukupan broj čvorova, odnosno trenabilnih parametara prve eksperimentalne neuralne mreže iznosi 326.865 nakon poslednjeg potpuno povezanog sloja. Projektovani model prve eksperimentalne arhitekture ćemo nazvati  $J\_exp0$ .

Model je treniran korišćenjem baze podataka koja je prethodno kreirana, korišćenjem srednje kvadratne greške, eng. *mean squared error* (MSE), funkcije gubitka i *Adam* optimizatora, što je detaljnije objašnjeno u poglavlju 5.4. Autonomno održanje vozila u kolovoznoj traci u realnom vremenu korišćenjem opisane duboke neuralne mreže nije uspešno ostvareno. Korišćenjem predstavljene konvolucione neuralne mreže, vozilo nije bilo u stanju da se održava u kolovoznoj traci, već je skrenulo s definisanog puta. Međutim, ocena kvalitativnog učinka pokazuje da je istrenirani model ipak naučio neke korisne karakteristike. Kao što se može videti iz video snimka autonomne vožnje u okruženju simulatora [100], model je naučio da prati liniju, tako da vozilo uspešno vozi pored same ivice jezera, prateći liniju između zemlje i vode. Sprovedeni eksperiment je pokazao da je odabran pravac kreiranja duboke neuralne mreže za održanje vozila u kolovoznoj traci odgovarajući. Jedino, modelu je neophodno obezbediti izdvajanje više odlika iz originalne slike kako bi se postigla uspešna autonomna vožnja, tj. održanje vozila u kolovoznoj traci.

Kao što se može videti iz prvih eksperimentalnih rezultata, iako je korišćena neuralna mreža veoma plitka, sastoji se iz samo tri sloja, broj čvorova, težinskih faktora i parametara koji se treniraju je visok. Objašnjenje je u tome što se kao ulaz u neuralnu mrežu koristi čitava slika snimljena kamerom koja se nalazi na vozilu. Nakon primene konvolucije i ostalih operacija, dolazi se do velikog broja čvorova u izlaznom sloju neuralne mreže. Takođe, u predstavljenom modelu, nisu se koristile operacije za redukciju broja parametara, kao npr. operacija udruživanjem. Rezultati eksperimenta sa predstavljenom jednostavnom neuralnom mrežom su važni za sledeću fazu razvoja konačnog rešenja odgovarajuće duboke neuralne mreže za autonomnu vožnju. Mogući pravci za unapređenje arhitekture neuralne mreže:

1. Isecanje slike. Delovi slike kao što su nebo ili donji deo slike koji zahvata haubu automobila nisu relevantni za autonomnu vožnju. Ono što je relevantno za učenje odgovarajućih reprezentacija su put, krivine i ivice puta kao što su crvena granična linija puta, ivičnjaci, bankina, prašina ili ograda most.
2. Izvršiti normalizaciju podataka kako bi se imao isti opseg vrednosti za svaki od ulaznih podataka modela. Ovo garantuje stabilnu konvergenciju težinskih faktora i odstupanja.
3. Koristiti operaciju udruživanja kako bi se smanjio broj čvorova neuralne mreže u narednom sloju, i posledično, broj parametara koji se treniraju. Sloj sa operacijom udruživanja se generalno koristi za smanjivanje veličine izlaza mreže i prevenciju previše prilagođenog modela.
4. Koristiti više konvolucionih slojeva, s obzirom da su konvolucione operacije odgovorne za izdvajanje odlika. Prvi eksperiment je pokazao da konvolucionni sloj može da izdvoji neke odlike neophodne za autonomnu vožnju (npr. jedna odlika je da vozilo prati liniju), ali jedan konvolucionni sloj nije dovoljno dobar za rešavanje postavljenog problema, već je potrebno kreirati više mapa odlika.

Kako arhitekturu dubokih konvolucionih neuralnih mreža čine parametri kao što su dubina po broju i tipu slojeva, dubina konvolucionih slojeva, odnosno broj mapa odlika, veličina kernela konvolucionog sloja, korišćenje udruživanja maksimizacijom, kao i broj i veličina potpuno povezanih slojeva, eksperimentisanje sa varijacijama ovih parametara dovodi nas do konačne arhitekture duboke neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju.

## **6.2. Analiza uticaja dubine neuralne mreže na performanse funkcionalnog modela**

Kako je uspešna autonomna vožnja na reprezentativnoj putanji cilj postavljenog problema, odlike koje je potrebno da isprojektovana neuralna mreža izdvoji nisu objekti, već jednostavnije odlike ili geometrijski oblici, odnosno karakteristike kolovozne trake i njenog okruženja. Kako bi se izdvojio

odgovarajući broj ovih karakteristika, u metodologiji koja je primenjena pristupilo se eksperimentisanju sa nekoliko konvolucionih slojeva, sa različitim brojem mapa odlika, odnosno broja filtara koji se koriste pri svakoj konvoluciji. Poslednji konvolucionni sloj je praćen ravnujućim slojem, dok je za poslednje slojeve mreže odabrano nekoliko potpuno povezanih slojeva.

Uvođenje više skrivenih slojeva u arhitekturu duboke neuralne mreže pozitivno utiče na efikasnost parametara mreže. Bolje performanse rešenja se dobijaju dodavanjem novih slojeva neuralnoj mreži a ne povećavanjem dimenzija postojećih slojeva. Pored ovoga, primena dubokih neuralnih mreža na slike i kompjutersku viziju uopšte je veoma efikasna s obzirom na to da se u slikama može pronaći hijerarhijska struktura koju model može da uoči. Početni slojevi duboke neuralne mreže uočavaju jednostavnije odlike kao što su linije ili ivice. Dalji slojevi izdvajaju komplikovanije odlike koje se dobijaju kombinacijom prethodnih jednostavnijih odlika, npr. geometrijski oblici, i poslednji slojevi duboke neuralne mreže su u mogućnosti da izdvoje čitave objekte. Iz tog razloga, u konačnom modelu koji se predlaže za rešavanje postavljenog problema, odabrana su tri konvoluciona sloja praćena ravnujućim slojem, i na kraju se nalaze dva potpuno povezana sloja.

Kako bi se bolje izdvojile odlike, za početak eksperimenta, odabrana su tri konvoluciona sloja za izdvajanje odlika. Ova tri konvoluciona sloja su projektovana sa 16, 32, i 64 mapa odlika, respektivno. Veličina ulaza u neuralnu mrežu, odnosno u prvi konvolucionni sloj je  $320 \times 65 \times 3$ , nakon normalizacije i odsecanja slike. Nakon ovoga, primenjuje se kernel veličine  $5 \times 5$  sa 16 mapa odlika. Na osnovu jednačina (6.2) i (6.3), ukupan broj parametara koji se treniraju nakon prvog konvolucionog sloja je 1216.

Kako je inicijalna ideja da se projektuje rešenje koje nije suviše kompleksno i zahtevno sa stanovišta računarske snage i procesorskih resursa, a konvolucija predstavlja veoma skupu operaciju koja dodaje veliki broj čvorova neuralnoj mreži i težinskih faktora koji su dodeljeni ovim čvorovima, neophodno je primeniti neku od operacija za smanjenje broja čvorova. Jedno rešenje bi moglo biti da se uključi veći korak pri svakoj konvoluciji, da se filteri pomeraju za po nekoliko piksela u svakoj iteraciji i time smanji veličina mape odlika. Međutim, ovakvo smanjenje uzrokovanja piksela iz prethodnog sloja neuralne mreže može da uzrokuje gubitak nekih važnih odlika jer bi se uklonile mnoge informacije iz slike nad kojim se radi konvolucija. Drugo rešenje za smanjenje broja čvorova je primena operacije udruživanja. Umesto da se pikseli preskaču između svake dve konvolucije, koristi se metoda udruživanja maksimizacijom, gde se iz jednog okruženja piksela bira najveća vrednost i ona ulazi u naredni sloj neuralne mreže, predstavljajući sve okolne piksele koji će ubuduće biti zanemareni kao takvi, ali koji su u samom postupku udruživanja bili uzeti u obzir.

U postupku udruživanja maksimizacijom, svaki piksel mape odlika se poredi sa malim okruženjem oko tog piksela i piksel sa maksimalnom vrednošću se uzima za dalje računanje:

$$y = \max (X_i), \quad (6.5)$$

gde je  $X_i$  vrednost jednog od ulaza, u našem slučaju piksela.

Prva prednost korišćenja operacije udruživanja maksimizacijom je to što ova operacija ne generiše nove parametre, odnosno nove težinske faktore dodeljene pikselima, već se prosleđuju parametri iz prethodnog sloja. Ovim se smanjuje mogućnost da dođe do problema previše prilagođenog modela, eng. *overfitting*. Drugo, udruživanje maksimizacijom obično vodi do toga da model postiže veću tačnost. Dodavanjem novog sloja kao što je udruživanje, povećava se broj hiperparametara koji se mogu prilagođavati, veličina segmenta nad kojim se izvršava udruživanje i korak udruživanja. Sloj za udruživanje izvršava operacije nezavisno za svaku dubinu mape odlika ulaznog sloja menjajući veličinu ovog sloja prostorno. U našem modelu izabran je sloj udruživanja



maksimizacijom, eng. *MaxPooling*, veličine  $2 \times 2$  čime se broj čvorova u svakoj dubini prethodnog sloja smanjuje duplo, što dovodi do ukupnog umanjenja od 75% aktivacija, u odnosu na prethodni sloj. Dimenzija dubine sloja, mapa odlika, ostaje neizmenjena. Primenom operacije udruživanja maksimizacijom se redukuje broj parametara za treniranje neuralne mreže, dok se mapa odlika ne degradira značajno.

Najpre se pristupilo analizi uticaja dubine neuralne mreže na realizaciju funkcionalnog modela. Deo različitih arhitektura sa kojima je eksperimentisano, zajedno sa opisom unutrašnjih slojeva ovih mreža, je prikazan u tabeli 6.1. Prvi model predstavlja mrežu koja je inspirisana LeNet-5 arhitekturom [102], nakon čega slede tri eksperimentalne arhitekture sa različitim brojem slojeva, od 3 do 7. Cilj ove analize je bio da se utvrdi minimalna kompleksnost mrežne arhitekture koja daje funkcionalni model, odnosno kojim je moguće vršiti uspešno održanje vozila u kolovoznoj traci. Stoga, osim uspešno izvršenog zadatka autonomne vožnje, posmatrana je mrežna arhitektura sa najmanjim brojem parametara, odnosno matematičkih operacija. Modeli neuralnih mreža su trenirani koristeći postupak koji je opisan u poglavlju 5.4, dok se uspešnost izvršavanja postavljenog zadatka definiše u poslednjem redu tabele.

**Tabela 6.1** Eksperimentalne arhitekture dubokih neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju testirane na simulatoru autonomne vožnje.

	LeNet-like	J_exp0	J_exp1	J_exp2	J_exp3
sloj 1	MaxPool2D (2, 10)	Cnonv2D (16, (5, 5))	Cnonv2D (16, (5, 5))	Cnonv2D (16, (5, 5))	Cnonv2D (16, (5, 5))
/	Cnonv2D (6, (5, 5))	/	MaxPool2D (2, 2)	MaxPool2D (2, 2)	MaxPool2D (2, 2)
sloj 2	MaxPool2D (2, 2)	Flatten	Conv2D (32, (5, 5))	Conv2D (32, (5, 5))	Conv2D (32, (5, 5))
/	Conv2D (16, (5, 5))	/	MaxPool2D (2, 2)	MaxPool2D (2, 2)	MaxPool2D (2, 2)
sloj 3	MaxPool2D (2, 2)	Dense(1)	Flatten	Conv2D (64, (3, 3))	Conv2D (64, (3, 3))
/	/	/	/	MaxPool2D (2, 2)	MaxPool2D (2, 2)
sloj 4	Flatten	/	Dense(10)	Flatten	Flatten
sloj 5	Dense(120)	/	Dense(1)	Dense(10)	Dense(100)
sloj 6	Dense(84)	/	Dense(1)	Dense(1)	Dense(10)
sloj 7	Dense(1)	/	/	/	Dense(1)
broj parametara	61,241	326,865	334,389	150,965	1,217,665
uspešnost	neuspešna	neuspešna	neuspešna	uspešna	uspešna

Slojevi arhitekture neuralnih mreža u tabeli 6.1 su prikazani korišćenjem naziva na engeskom jeziku, koji ujedno predstavljaju i nazive konkretnih funkcija u Keras biblioteci za mašinsko učenje koja je korišćena za implementaciju predstavljenih rešenja: *Conv2D* – dvodimenzioni konvolucionni sloj, *MaxPool2D* – operacija udruživanja maksimizacijom, *Flatten* – ravnajući sloj, *Dense* – potpuno

povezani sloj. Na primer, konvencija  $C_{nonv2D}(16, (5, 5))$  predstavlja dvodimenzionu operaciju konvolucije, sa 16 mapa odlika, i veličinom kernela  $5 \times 5$ . Konvencija  $MaxPool2D(2, 2)$  predstavlja operaciju udruživanja maksimizacijom, gde se od 4 piksela ( $2 \times 2$ ) bira maksimalni piksel kao ulaz u naredni sloj duboke neuralne mreže. Konvencija  $Dense(10)$  predstavlja potpuno povezani sloj, gde se svi čvorovi neuralne mreže koji su ulaz u taj potpuno povezani sloj povezuju sa 10 izlaznih čvorova iz tog sloja. Takođe, u prvoj koloni table 6.1. prikazani su redni brojevi čvorova. Kako je  $MaxPool2D$  operacija udruživanja maksimizacijom, a ne poseban sloj neuralne mreže, ova operacija se ne ubraja u broj slojeva eksperimentalne mrežne arhitekture.

Na osnovu rezultata testiranih eksperimentalnih arhitektura, odlučeno je da se nastavi sa arhitekturom modela J\_exp2, kako je ovo model sa najmanjim brojem trenarabilnih parametara, 150.965, a koji je ostvario uspešnu autonomnu vožnju. Dalje će se izvršiti analiza uticaja dubine konvolucionog sloja i veličine kernela na realizaciju funkcionalnog modela.

### 6.3. Analiza uticaja dubine konvolucionog sloja i veličine kernela na performanse funkcionalnog modela

Varijacijom broja i vrste slojeva duboke neuralne mreže i testiranjem na opisanom okruženju koje koristimo za validaciju autonomnog održanja vozila u kolovoznoj traci, zaključeno je da eksperimentalni model J\_exp2 daje najbolje rezultate, pa se dalje pristupilo varijacijama parametara unutar ove mrežne arhitekture. Ova arhitektura se sastoji od tri konvoluciona sloja, sa 16, 32 i 64 odlika mapa respektivno, gde je svaki konvolucioni sloj praćen operacijom urduživanja maksimizacijom, jednog ravnanjućeg sloja, i dva potpuno povezana sloja od 10 i 1 završnih čvorova.

U novoj analizi, model J\_exp2 se naziva J1, i dalje se pristupa varijacijama dubine konvolucionih slojeva. Vršene su varijacije konvolucionih slojeva sa 16, 32 i 64 odlika mapa, odnosno dubina po konvolucionom sloju. Razlike u dubini pojedinačnog konvolucionog sloja u odnosu na njegov redni broj u arhitekturi neuralne mreže značajno utiče na krajnji broj parametara, kao što se može videti u tabeli 6.2, gde je u četvrtom redu otpozadi prikazan broj parametara za svaki od testiranih modela.

**Tabela 6.2** Varijacije J-Net modela po broju mapa odlika u svakom konvolucionom sloju.

Slojevi	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12
conv2D	16, (5,5)	16, (5,5)	16, (5,5)	32, (5,5)	32, (5,5)	32, (5,5)	64, (5,5)	64, (5,5)	64, (5,5)	16, (5,5)	64, (5,5)	16, (5,5)
MaxPool	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
conv2D	32, (5,5)	16, (5,5)	64, (5,5)	32, (5,5)	16, (5,5)	64, (5,5)	16, (5,5)	64, (5,5)	32, (5,5)	16, (5,5)	16, (5,5)	64, (5,5)
MaxPool	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
conv2D	64, (3,3)	32, (3,3)	16, (3,3)	64, (3,3)	32, (3,3)	16, (3,3)	32, (3,3)	16, (3,3)	16, (3,3)	64, (3,3)	64, (3,3)	32, (3,3)
MaxPool	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
Flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten
Dense	10	10	10	10	10	10	10	10	10	10	10	10
Dense	1	1	1	1	1	1	1	1	1	1	1	1
Ukupno parametara	150965	71493	65733	164981	79109	92549	94341	146181	90341	135333	158181	104565
uspešnost	uspešan	neuspešan	uspešan	neuspešan	uspešan	neuspešan	neuspešan	neuspešan	neuspešan	neuspešan	uspešan	neuspešan
MSE [ $m^2$ ]	11,04	/	26,03	/	24,91	/	/	/	/	/	12,34	/
IAE [ $m$ ]	0,26	/	0,41	/	0,42	/	/	/	/	/	0,26	/

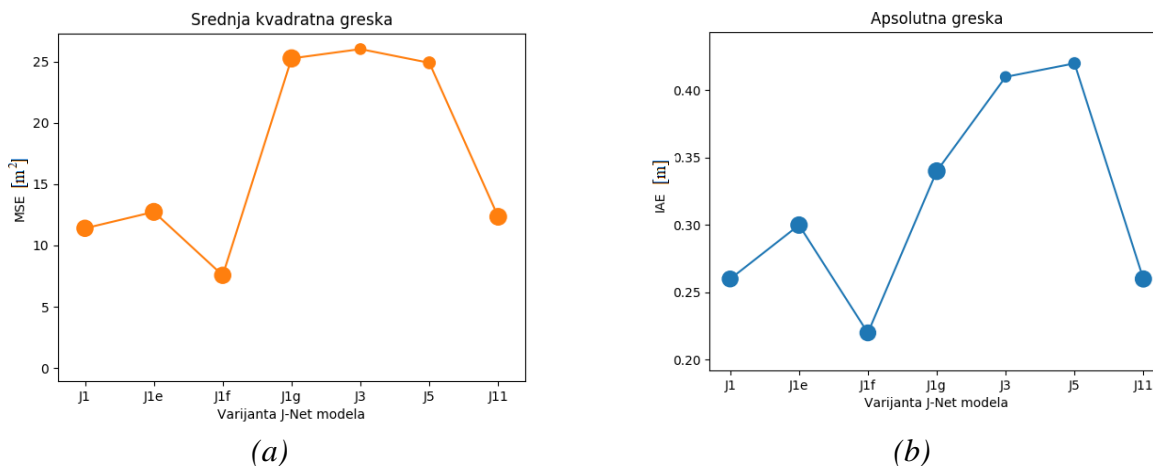
Metoda za merenje uspešnosti održanja vozila u kolovoznoj traci je detaljno opisana u poglavlju 5.5, a u tabeli 6.2 uspešnost modela je označena u trećem redu otpozadi, gde se svaki od modela definiše kao uspešan ukoliko je moguće ostvariti autonomno održanje vozila na kolovoznoj traci tokom čitave putanje testirano u simulatoru autonomne vožnje, a neuspešan ukoliko je vozilo skrenulo sa putanje sa nemogućnošću vraćanja na istu ili se zaustavilo. U poslednja dva reda tabele 6.2 su dati rezultati za srednju kvadratna greška (MSE) i integralnu apsolutnu grešku (IAE). Postupak za merenje ovih vrednosti opisan je u poglavlju 5.5.

Na osnovu rezultata iz tabele 6.2 zaključuje se da korišćenjem modela J1 dolazimo do uspešne autonomne vožnje sa najmanjom srednje kvadratnom i apsolutnom greškom. Dalje eksperimentisanje, odnosno variranje parametara arhitekture duboke neuralne mreže je vršeno upravo nad ovim modelom, koji za dubine svoja tri konvoluciona sloja ima 16, 32 i 64 mapa odlika, respektivno. U tabeli 6.3. prikazani su modeli kod kojih je varirana veličina kernela konvolucionih slojeva. Model J1 je testiran u 7 varijanti, od J1-a do J1-g. Rezultati uspešnosti autonomne vožnje, kao i srednje kvadratna i apsolutna greška koje su računate za uspešne modele, su prikazani u poslednja tri reda tabele 6.3.

**Tabela 6.3** Varijacije J-Net modela po veličini kernela unutar konvolucionih slojeva.

slojevi	J1-a	J1-b	J1-c	J1-d	J1-e	J1-f	J1-g
conv2D	<b>16, (5,5)</b>	<b>16, (5,5)</b>	<b>16, (3,3)</b>	<b>16, (5,5)</b>	<b>16, (5,5)</b>	<b>16, (3,3)</b>	<b>16, (5,5)</b>
MaxPool	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
conv2D	<b>32, (5,5)</b>	<b>32, (3,3)</b>	<b>32, (3,3)</b>	<b>32, (3,3)</b>	<b>32, (5,5)</b>	<b>32, (5,5)</b>	<b>32, (5,5)</b>
MaxPool	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
conv2D	<b>64, (3,3)</b>	<b>64, (3,3)</b>	<b>64, (3,3)</b>	<b>64, (1,1)</b>	<b>64, (1,1)</b>	<b>64, (3,3)</b>	<b>64, (2,2)</b>
MaxPool	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)
Flatten	flatten	flatten	flatten	flatten	flatten	flatten	flatten
Dense	10	10	10	10	10	10	10
Dense	1	1	1	1	1	1	1
Ukupno parametara	150965	170293	169525	182709	162101	150192	168245
uspesnost	uspešan	neuspešan	neuspešan	neuspešan	uspešan	uspešan	uspešan
MSE [ $m^2$ ]	11,04	/	/	/	12,74	7,56	25,27
IAE [ $m$ ]	0,26	/	/	/	0,30	0,22	0,34

Srednja kvadratna greška (MSE) i apsolutna greška (IAE) za pun krug vožnje po reprezentativnoj putanji su izračunate korišćenjem jednačine 5.2 za MSE i jednačine 5.3 za IAE, a na osnovu postupka opisanog u poglavlju 5.5. Na slici 6.2, prikazane su greške odstupanja kretanja automobila od sredine kolovozne trake, i to (a) srednja kvadratna greška - MSE i (b) integralna apsolutna greška - IAE. U oba slučaja veličine modela, ukupan broj parametara mreže, su proporcionalno prikazane veličinom tačaka na grafiku, pa se tako može videti da je veličina tačke na grafiku za model J11 značajno veća od veličine tačke koja predstavlja model J3, s obzirom da se srazmerno razlikuje i ukupni broj trenabilnih parametara.



Slika 6.2 Grafički prikaz odstupanja automobila od sredine kolovozne merenjem srednje kvadratne i integralne apsolutne greške; (a) MSE i (b) IAE.

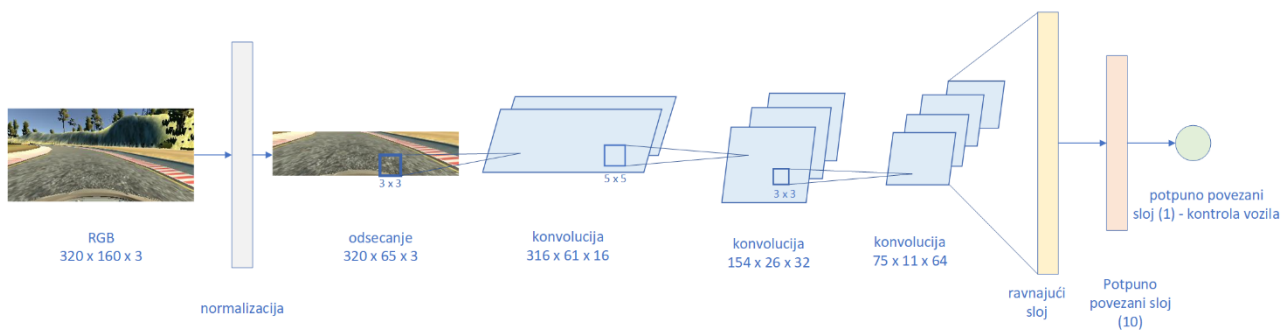
Za modele koji nisu ostvarili uspešnu autonomnu vožnju, poput eksperimentalnih arhitektura J2, J4 i druge, nisu računane srednja kvadratna greška i apsolutna greška. Razlog za ovo je taj što poređenje greške odstupanja vozila od sredine kolovozne trake ne bi bilo adekvatno jer se greške ne bi računale na čitavoj putanji. Detaljnije, kada vozilo uspešno prođe čitav krug u simulatoru, putanja sadrži više kritičnih krivina u kojima se vozilo može naći uz ivicu puta i tada su odstupanja od sredine puta najveća. Ukoliko bismo računali MSE i IAE za neuspešne modele, ovo merenje bi se moglo izvršiti samo na delu puta koji je bio uspešan, međutim, to onda nije adekvatno poređenje jer taj uspešno pređeni deo puta za neuspešne modele je značajno kraći od kompletne putanje i ne sadrži sve kritične tačke u kojima dolazi do najvećih odstupanja.

Polazne hipoteze za odabir konačne neuralne mreže koja uspešno izvršava zadatak autonomnog održanje vozila na kolovoznoj traci, a čije odstupanje vožnje od sredine kolovozne trake ne prelazi  $15 m^2$  za MSE i  $0,35 m$  za IAE, i koja pri tome ima najmanju kompleksnost, odnosno najmanji broj trenarabilnih parametara i matematičkih operacija koje se izvršavaju u jednoj iteraciji. Ovim se zaključuje da eksperimentalna arhitektura duboke neuralne mreže J1-f zadovoljava sve kriterijume, uspešno izvršava zadatak autonomnog održanja vozila na kolovoznoj traci, ima najmanje odstupanje od svih testiranih modela, MSE iznosi  $7,56 m^2$ , a IAE  $0,22 m$  što predstavlja minimum na oba dijagrama na slici 6.2. Ukupan broj trenarabilnih parametara ovog modela iznosi 150.192. J1-f model proglašavamo konačnom varijantom J-Net modela.

## 6.4. Konačna arhitektura duboke neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju

Na osnovu eksperimenata i analize izložene gore u ovom poglavlju, proglašena je konačna arhitektura duboke neuralne mreže za učenje od-kraja-do-kraja za autonomnu vožnju, J-Net.

Kod konačnog J-Net modela, raspored slojeva duboke neuralne mreže i operacija je sledeći: prvi konvolucioni sloj ima 16 mapa odlika, i veličinu kernela  $3 \times 3$ , drugi konvolucioni sloj ima 32 mapa odlika, veličinu kernela  $5 \times 5$ . Treći konvolucioni sloj se sastoji od 64 mapa odlika i veličine kernela  $3 \times 3$  i ovaj konvolucioni sloj ima 64 mape odlika. Svaki konvolucioni sloj je direktno paćen ReLU aktivacionom funkcijom, koja je opisana u poglavlju 3.2, praćeno sa slojem udruživanja maksimizacijom veličine  $2 \times 2$ .



Slika 6.3 Arhitektura predložene duboke mreže za učenje od-kraja-do-kraja, J-Net.

Kako je u pitanju projektovanje mreže za reprezentativno učenje, ovi čvorovi neuralne mreže se moraju spojiti u jedan krajnji čvor. Nakon poslednje konvolucije, primenjuje se ravnajući sloj čime se svi parametri neuralne mreže dovode u jednodimenzioni vektor. Konačno, poslednja dva sloja projektovane duboke neuralne mreže su dva potpuno povezana sloja. Prvi potpuno povezani sloj sadrži deset izlaznih čvorova, a drugi potpuno povezani čvor se sastoji iz samo jednog izlaznog čvora, koji predstavlja ugao okretanja upravljača. Ukupan broj parametara koji su direktno vezani za težinske faktore čvorova ovako kreirane duboke neuralne mreže, za ulaznu sliku veličine  $320 \times 160 \times 3$  iznosi 150.192. Ovu mrežnu arhitekturu nazvaćemo J-Net. Arhitektura konačne J-Net modela arhitekture duboke neuralne mreže je prikazana na slici 6.3.

## 7. Verifikacija autonomne vožnje u simuliranim uslovima

Autonomno održanje vozila u kolovoznoj traci korišćenjem informacija sa vizuelnih senzora se može rešavati metodama klasične kompjuterske vizije ili primenom metoda dubokog učenja. Ukoliko se koriste algoritmi klasične kompjuterske vizije, održanje vozila u kolovoznoj traci predstavlja jednu komponentu autonomne vožnje, dok se druge komponente moraju nezavisno i u paraleli rešavati. Sa druge strane, korišćenjem dubokog učenja, preciznije korišćenjem duboke neuralne mreže za učenje od-kraja-do-kraja moguće je postići takozvano kloniranje ponašanja vozača. U ovom slučaju jedan model mašinskog učenja preuzima kontrolu nad vozilom tako što u realnom vremenu na osnovu slike sa kamere postavljene na vozilu, a koja predstavlja ulaz u istrenirani model neuralne mreže, donosi odluku o kontroli nad vozilom, odnosno o uglu okretanja upravljača. Kako bi zadatak bio pojednostavljen, brzina vozila je postavljena na konstantnu vrednost. Ovakvom metodom se zaista postiže potpuna autonomija u vožnji. U ovom radu, pod terminom autonomne vožnje podrazumeva se zadatak autonomnog održanja vozila u kolovoznoj traci.

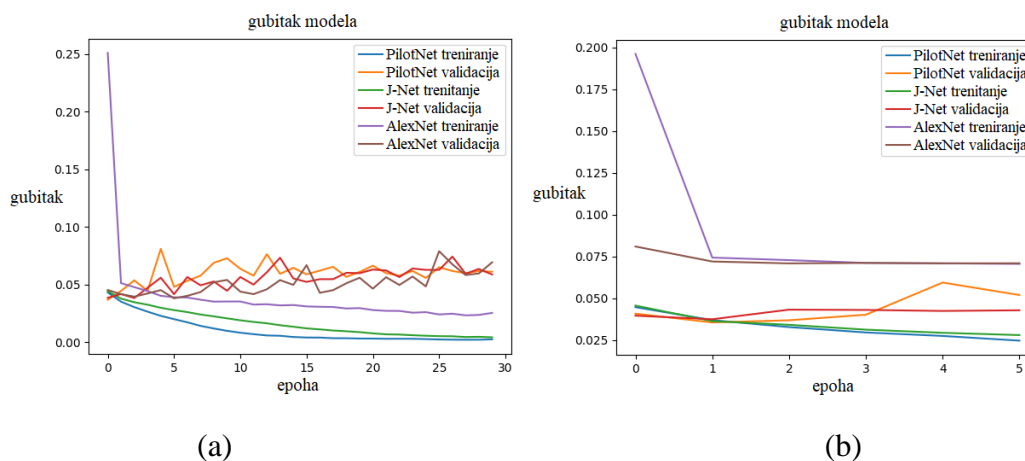
U ovom poglavlju je dat opis kompletnog postupka u ostvarivanju autonomne vožnje, prikupljanje baze podataka korišćenjem simulatora autonomne vožnje, treniranje neuralne mreže za učenje od-kraja-do-kraja korišćenjem ove baze i poređenje performansi autonomne vožnje korišćenjem J-Net modela i drugih poznatih modela.

### 7.1. Treniranje duboke neuralne

Implementacija tri arhitekture dubokih neuralnih mreža: J-Net, AlexNet i PilotNet je izvršena u programskom jeziku Python korišćenjem Keras biblioteke za mašinsko učenje [101] koja je napisana kao nadskup biblioteke TensorFlow. Sve tri mrežne arhitekture su implementirane od samog početka, kreirajući sloj po sloj. Platforma koja je korišćena za treniranje i fazu zaključivanja je personalni računar sa procesorom od 12 jezgra koji radi na 3.2 GHz i NVIDIA grafičkom kartom GeForce GTX 1070-Ti sa 8GB GDDR5 memorije i 8Gbps brzine memorije.

Postupak treniranja neuralnih mreža opisan je u poglavlju 5.4, i u istom poglavlju na slici 5.8. je prikazan dijagram toka treniranja duboke neuralne mreže za učenje od-kraja-do-kraja. Sva tri dizajnirana modela su nezavisno trenirana korišćenjem baze podataka koja je prethodno kreirana. Sve tri mreže su trenirane korišćenjem MSE funkcije gubitka i Adam optimizatora. Rezultati treniranja, odnosno dijagram gubitaka u različitim epohama tokom treniranja i validacije za sva tri modela prikazan je na slici 7.1.

Kako je korišćen relativno mali skup podataka, postoji opasnost od pojave problema previše prilagođenog modela. Iz ovog razloga primenjena je regularizaciona metoda ranog zaustavljanja, eng. *early stopping*. Razvijeni modeli J-Net, PilotNet i AlexNet su trenirani korišćenjem različitog broja epoha treniranja. Na početku, svi modeli su trenirani u 30 epoha, slika 7.1(a). Međutim, samo za AlexNet odabran broj epoha je dao zadovoljavajuće rezultate; gubitak tokom validacije se smanjuje istovremeno kao i gubitak tokom treniranja. Sa druge strane, treniranje J-Net i PilotNet modela u 30 epoha je dovelo do previše prilagođenog modela – gubitak tokom validacije se povećava, dok se gubitak tokom treniranja smanjuje. Korišćenjem J-Net i PilotNet mreža treniranih u ovom broju epoha u autonomnom režimu rada doveo je do toga da se vozilo uspešno kreće jednim delom puta, da bi nakon ulaska u krivinu skrenulo van puta i time nije uspešno izvršena autonomna vožnja na predefinisanoj traci.



Slika 7.1 Gubitak modela tokom treniranja i validacije za J-Net, PilotNet i AlexNet: (a) treniranje u 30 epoha; (b) treniranje u 6 epoha.

Kako bi se sprečio problem previše prilagođenog modela, najpre se pristupilo tehnici ispadanja, što je takođe regularizaciona tehnika, gde se pojedini čvorovi iz neuralne mreže prosto obrišu, odnosno izbace. Ova tehnika izbacivanja je primenjena kao sloj neuralne mreže nakon prvog ravnujućeg sloja u mrežnoj arhitekturi. Međutim, iako je tehnika izbacivanja delimično poboljšanja uspešnost održanja vozila u kolovoznoj traci tokom autonomne vožnje, vozilo nije bilo u mogućnosti da pređe čitavu putanju predefinisane trajektorije.

U konačnom rešenju, primenjena je regularizaciona metoda ranog zaustavljanja. Kao što je bilo očekivano, ranim zaustavljanjem je postignut manji gubitak u validaciji za oba modela, J-Net i PilotNet, dok je gubitak tokom validacije AlexNet modela ostao na istom nivou kao i pri upotrebi većeg broja epoha. S obzirom na to da je set podataka korišćenih za treniranje isti kod sva tri modela, razlika u odnosu između gubitka tokom treniranja i validacije u različitom broju epoha je direktno povezan sa razlikama u mrežnim arhitekturama. Oba modela, J-Net i PilotNet, imaju manji broj parametara koji se treniraju u odnosu na AlexNet model. Iz tog razloga, za istu veličinu baze podataka kojom se treniraju ovi modeli, više je verovatno da će se problem previše prilagođenog modela pojaviti kod prva dva modela. Primenom tehnike ranog zaustavljanja, J-Net i PilotNet modeli su proizveli očekivanu reakciju, smanjio se problem previše prilagođenog modela, odnosno smanjen je gubitak tokom validacije kada je smanjivan i gubitak tokom treniranja, slika 7.1(b).

U slučaju treniranja PilotNet modela, u obe varijante, kada se trenira u 30 epoha i 6 epoha, postoji maksimum u petoj epohi za gubitak validacije, kao što se može videti sa slike 7.1. Primetna razlika u vrednosti gubitka pri treniranju i validaciji za tu epohu je jasan znak da je potrebno primeniti tehniku ranog zaustavljanja i odabrati četiri epohe za treniranje PilotNet modela. Testiranje istreniranog modela u režimu autonomne vožnje potvrđuje ovaj zaključak. Istrenirani PilotNet model ima najbolju performansu autonomne vožnje kada je model treniran u četiri epohe. Sa druge strane, u slučaju J-Net modela odabir 6 epoha za treniranje je posledica empirijskog izbora.

Za treniranje J-Net modela, eksperimenti su pokazali da iako je gubitak tokom validacije nizak za početne epohe treniranja, broj epoha može varirati od četiri do 10, pri čemu se dolazi do sličnih rezultata. Odabrano je šest epoha za treniranje J-Net modela, čime je dobijen model koji uspešno ispunjava zadatak autonomne vožnje na čitavoj predefinisanoj putanji.

Istrenirani modeli su snimljeni i korišćeni kasnije u režimu zaključivanja, odnosno u režimu autonomne vožnje. Kao što se moglo očekivati, na osnovu ukupnog broja parametara, odnosno čvorova neuralne mreže sa dodeljenim težinskim parametrima, istrenirani model koji zauzima najmanje memorijskog prostora je J-Net sa samo 1,8 MB. Model koji zahteva najviše memorijskog

prostora je AlexNet sa 509,5 MB, što je u skladu sa brojem trenabilnih parametara netrenirane mreže, ukupno preko 44 miliona parametara. Istrenirani model PilotNet zauzima 4,2 MB memorijskog prostora.

## **7.2. Usporedna analiza J-Net i drugih modela neuralnih mreža za autonomnu vožnju**

Model projektovane duboke neuralne mreže J-Net je upoređen sa PilotNet i AlexNet modelima, koji su modifikovani i implementirani kako bi se izvršila objektivna evaluacija performansi novog dizajna, odnosno J-Net arhitekture. Modeli sve tri mrežne arhitekture su implementirani, trenirani sa istom bazom podataka i istrenirani modeli su korišćeni u režimu zaključivanja za autonomnu vožnju u simulatoru. Upoređeni su rezultati: performanse autonomne vožnje, odnosno uspešno održanje vozila u kolovoznoj traci na predefinisanoj traci u simulatoru, kompleksnost neuralne mreže, broj trenabilnih parametara i veličina treniranih modela. Tokom autonomne vožnje upoređeni su kvantitativni rezultati vožnje u realnom vremenu sa akcentom na kašnjenje.

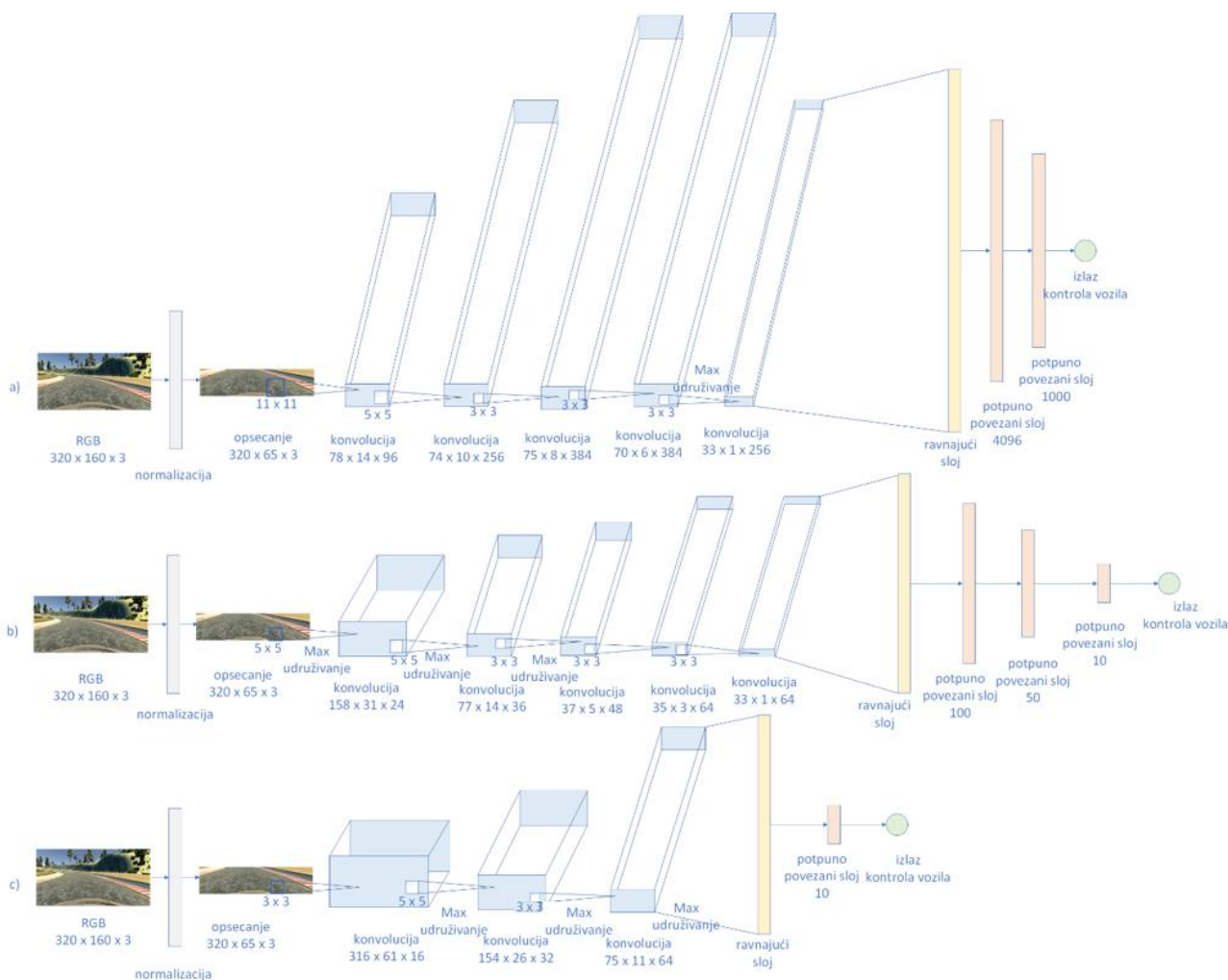
### **7.2.1. Računarska kompleksnost**

Kako bi se napravila objektivna evaluacija performansi duboke neuralne mreže za učenje od-kraja-do-kraja J-Net, najpre je izvršeno poređenje sa tri poznate mrežne arhitekture koje su takođe implementirane za potrebe učenja od-kraja-do-kraja i primenjene na isti problem autonomne vožnje: LeNet-5 [102], AlexNet [55], and PilotNet [91]. Kako bi ove poznate mrežne arhitekture mogle biti primenjene na konkretnom problemu koji je rešavan, bilo je neophodno da se izvrše izvesne modifikacije u arhitekturama ovih mreža. Osnovna ideja je da se napravi objektivna evaluacija uspešnosti J-Net modela u odnosu na učinak koji je moguće postići korišćenjem do sada poznatih arhitektura. Iz ovog razloga, sva tri modela su implementirana i trenirana korišćenjem iste baze podataka, i autonomna vožnja je testirana u istim uslovima.

U slučaju modela LeNet-5 nije bilo moguće izvršiti uspešnu autonomnu vožnju, odnosno vozilo nije moglo da se održi u kolovoznoj traci u režimu autonomne vožnje. U slučaju korišćenja modela AlexNet i PilotNet postignuta je uspešna autonomna vožnja, što će omogućiti adekvatno poređenje performansi ovih modela sa projektovanim J-Net modelom. Na slici 7.2 je prikazano poređenje arhitektura modela J-Net, PilotNet i AlexNet.

Izvršavanje modela dubokih neuralnih mreža zavisi direktno od seta statičkih konstanti, težinskih faktora, koji se takođe nazivaju parametrima mreže. Sama arhitektura mreže, veze između čvorova, direktno određuju računarsku kompleksnost mreže. Jedna od osnovnih razlika između klasičnih neuralnih mreža i konvolucionih neuralnih mreža je to što kod potonjih nisu sve konekcije među neuronima povezane. Stoga, konkretna organizacija slojeva dubokih neuralnih mreža i precizna karakterizacija filterskih elemenata koji se primenjuju nad ovim slojevima određuje kompleksnost mreže.





Slika 7.2 Poređenje arhitektura dubokih neuralnih mreža koje primenjuju učenje od-kraja-do-kraja za autonomnu vožnju: (a) AlexNet; (b) PilotNet; (c) J-Net.

Kao kvantitativna mera računarske kompleksnosti svake od treniranih mreža, poredi se broj težinskih faktora, odnosno trenarabilnih parametara. Dubina mreže, broj slojeva, vrsta ovih slojeva, konvolucija, udruživanja, i ravnajućih slojeva, jedinstveno određuju broj trenarabilnih parametara. Ovo se odnosi na dizajn mrežnog modela koji još uvek nije treniran. U tabeli 7.1, prikazani su slojevi neuralnih mreža za svaki od tri modela, kao i ukupan broj trenarabilnih parametara.

**Tabela 7.1** Kvantitativni opis mrežnih arhitektura: J-Net, PilotNet i AlexNet.

Slojevi i parametri		AlexNet	PilotNet	J-Net
Konvolucija		5	5	3
Ravnajući sloj		1	1	1
Potpuno povezani sloj		3	4	2
Ukupan broj trenarabilnih parametara *		42.452.305	348.219	150.192
Operacije	Množenje	42,45 m	347,82 k	150,84 k
	Sabiranje	42,45 m	347,82 k	150,84 k

\* Ukupan broj trenarabilnih parametara je izračunat na osnovu veličine ulazne slike. Nakon normalizacije i odsecanja, veličina ulaznog podatka u mrežu je  $65 \times 320 \times 3$ .

Kao što se može videti iz table 7.1, AlexNet ima 42.452.305 trenarabilnih parametara, PilotNet ima 328.219 trenarabilnih parametara, a J-Net svega 150.192 trenarabilnih parametara. Dakle, J-Net ima oko 280 puta manje parametara nego AlexNet. Pored ovoga, tokom samog procesa treninga, izračunat je broj operacija sa pokretnim zarezom za svaki model koji je, kao što se i očekivalo, proporcionalan broju trenarabilnih parametara: 42,45 miliona operacija množenja i isti broj operacija sabiranja za AlexNet, 347,82 hiljada operacija množenja i isto toliko operacija sabiranja za PilotNet i oko 150,84 hiljada operacija množenja i sabiranja za J-Net, tabela 7.1.

Dodatno, upoređene su veličine treniranih modela: AlexNet model ima veličinu od 509,5 MB, PilotNet 4,2 MB, i J-Net samo 1,8 MB; tabela 7.2. Svi modeli su trenirani korišćenjem iste baze podataka, funkcije gubitka i optimizatora. Broj epoha korišćen za treniranje je različit za svaki model usled različitih potreba za rešavanje problema previše prilagođenog modela. Dok sama arhitektura mreže, tipovi i veličine slojeva i veze između čvorova, direktno utiču na cenu računanja, veličina istreniranog modela ima uticaj na fazu zaključivanja, kada se predikcije vrše u realnom vremenu direktno na hardverskoj platformi, s obzirom na zahteve u vezi memojskih resursa namenskih računarskih sistema.

**Tabela 7.2** Veličine istreniranih modela: J-Net, PilotNet, i AlexNet.

	<b>AlexNet</b>	<b>PilotNet</b>	<b>J-Net</b>
Broj epoha treniranja modela	30	4	6
Veličina istreniranog modela	509.5 MB	4.2 MB	1.8 MB

J-Net mreža ima oko 150 hiljada trenarabilnih parametara, što je upola manje od implementirane PilotNet mreže koja ima oko 350 hiljada trenarabilnih parametara, i broj parametara J-Net modela je 280 puta manji od broja parametara naše implementacije AlexNet mreže koja ima preko 41 milion parametara. Broj matematičkih operacija sa pokretnim zarezom, operacija množenja i sabiranja, je proporcionalan broju trenarabilnih parametara. Stoga se zaključuje da je J-Net najmanje računarski zahtevno rešenje. Veličina J-Net modela je četiri puta manja od PilotNet modela i oko 250 puta manja od AlexNet modela. Manja veličina istrenirane mreže i manji broj parametara mreže, težinskih faktora dodeljenih čvorovima mreže, direktno dovodi do poboljšanja performansi rada sistema u realnom vremenu u smislu smanjenja kašnjenja i samanjivanja zahteva za hardverskim resursima na kojima će se izvršavati model u fazi zaključivanja, sa stanovišta računarske snage, cene i veličine.

Bazirano na dobijenim rezultatima, može se zaključiti da J-Net mreža ima manje duboku arhitekturu od ostalih rešenja sa kojima je poređena, manji broj trenarabilnih parametara, i posledično, manju veličinu istreniranog modela. Sve ovo preporučuje dizajniranu mrežu za spuštanje na automobilsku platformu sa namenskim računarskim sistemom, odnosno na robo-vozilo.

### **7.2.2. Evaluacija performansi tokom autonomne vožnje u simuliranim uslovima**

Tokom režima autonomne vožnje, slika sa centralne kamere montirane na vozilu se kontinuirano prikuplja i šalje kao ulazni podatak istreniranom modelu neuralne mreže koja se koristi u fazi zaključivanja, izlaz ovog modela je zadata vrednost ugla okretanja upravljača. Razlika u odnosu na fazu treniranja je to što se u režimu zaključivanja koristi slika samo sa centralne kamere. Uspešna autonomna vožnja, odnosno održanje vozila u kolovoznoj traci u simulatoru, za sva tri modela je snimljena i video snimci ovih vožnji su dati u [103], [104] i [105] za AlexNet, PilotNet i J-Net, respektivno. Kao što se može uočiti iz pomenutih video snimaka, J-Net mreža je uspešno ispunila zahtev za autonomnom vožnjom na predefinisanoj putanji, pri čemu se vozilo održava u kolovoznoj traci tokom čitave vožnje. Kao mera boljeg performansa autonomne vožnje na datom putu uzima se

ponašanje vozila koje se kreće sredinom puta. Ovaj zahtev je J-Net model uspešno ispunio. Kvalitativna evaluacija autonomne vožnje korišćenjem implementiranih mreža je data u tabeli 7.3.

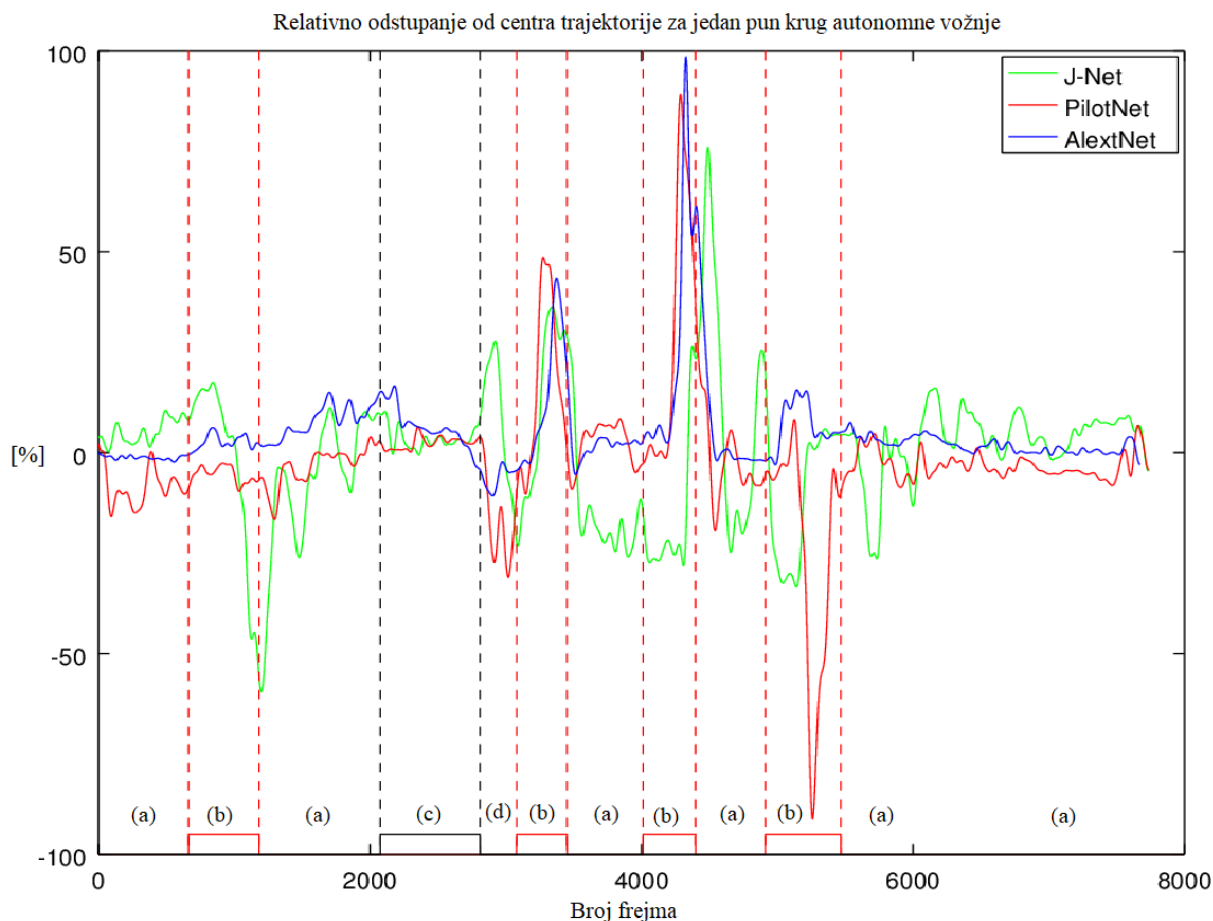
**Tabela 7.3** Evaluacija autonomne vožnje korišćenjem modela AlexNet, PilotNet i J-Net.

<b>Autonomna vožnja</b>	<b>AlexNet</b>	<b>PilotNet</b>	<b>J-Net</b>
Autonomno održanje vozila u kolovoznoj traci	uspešno	uspešno	uspešno
Srednja kvadratna grška (MSE) [m <sup>2</sup> ]	2,85	3,09	7,56
Apsolutna greska (IAE) [m]	0,11	0,10	0,22

U tabeli 7.3 u prvom redu opisana je uspešnost autonomne vožnje korišćenjem svakog od modela, AlexNet, PilotNet ili J-Net. Autonomno održanje vozila u kolovoznoj traci korišćenjem sva tri modela je uspešno. Na osnovu opservacija ponašanja vožnje u krivinama, može se zaključiti da korišćenjem AlexNet mreže tokom autonomne vožnje, vozilo se nalazi na sredini puta većinu vremena, dok se tokom autonomne vožnje korišćenjem PilotNet i J-Net modela vozilo nalazi u sredini puta uglavnom, ali u nekim krivinama vozilo dolazi do same ivice puta. Međutim, sve tri implementacije neuralnih mreža uspešno savladavaju autonomnu vožnju u simulatoru tokom čitavog trajanja vožnje, ne silazeći sa kolovozne trake.

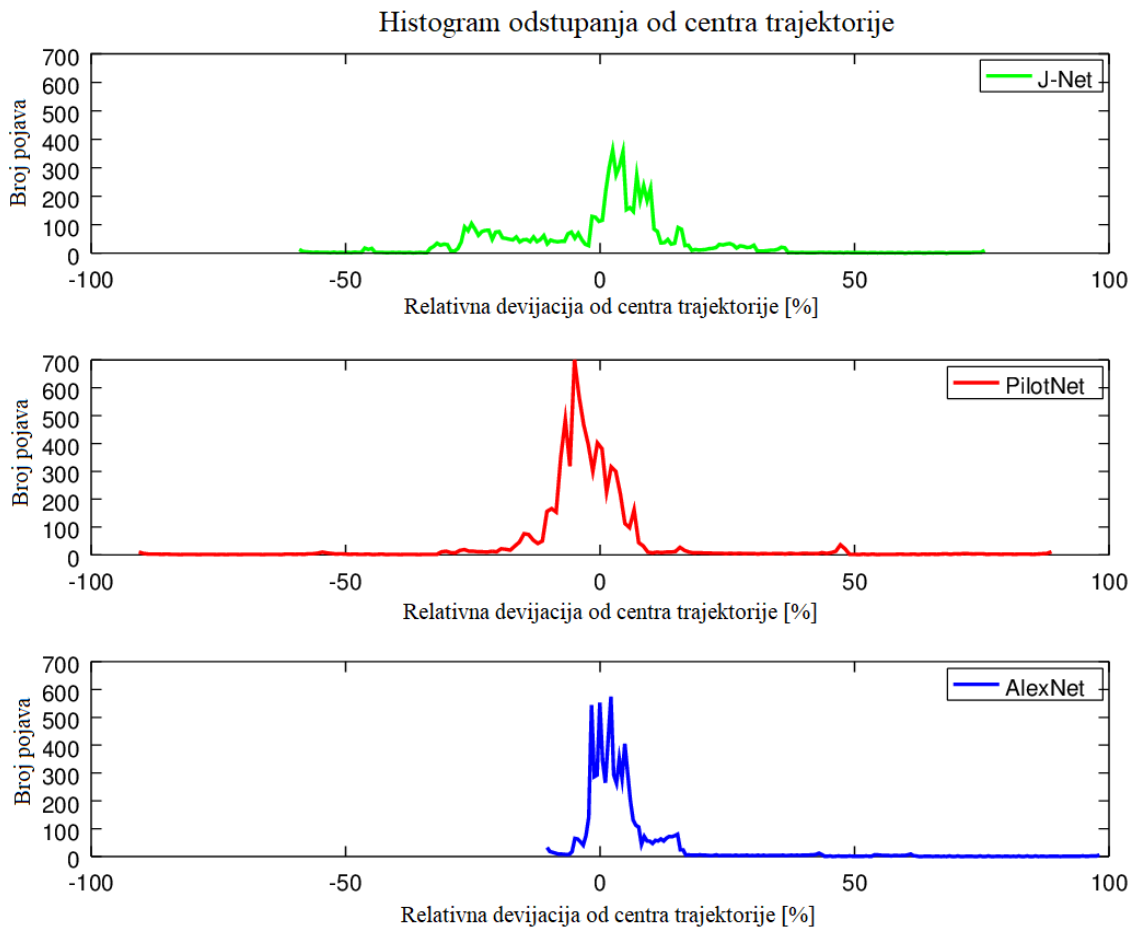
Kvantitativna ocena kvaliteta autonomnog održanja vozila na reprezentativnoj putanji je ostvarena merenjem odstupanja položaja vozila od sredine putanje kojom je predviđeno kretanje vozila. Tokom vožnje merene su pozicije vozila u svakom frejmu. Izračunata su odstupanja od centralne linije puta za autonomne vožnje korišćenjem AlexNet, PilotNet i J-Net modela. Izračunate su srednja kvadratna greška (MSE) i apsolutna greška (IAE) za pun krug vožnje po reprezentativnoj putanji, kao što je detaljnije objašnjeno u poglavlju 5.5. Rezultati za MSE i IAE za autonomnu vožnju korišćenjem sve tri mreže su date u tabeli 7.3 u drugoj i trećoj koloni, respektivno. Odstupanje je mereno u pikselima, pri čemu je uvedena konverzija 100 piksela = 1 m, kako bi se vrednosti predstavile i u metrima. Srednja kvadratna greška iznosi 2,85 m<sup>2</sup>, 3,09 m<sup>2</sup> i 7,56 m<sup>2</sup> za AlexNet, PilotNet i J-Net, respektivno. Apsolutna greška iznosi 0,11 m, 0,10 m i 0,22 m za AlexNet, PilotNet i J-Net, respektivno. Na osnovu vrednosti za MSE i IAE, zaključuje se da je postignuto najmanje odstupanje od centra predefinisane putanje korišćenjem AlexNet modela, a najveće odstupanje korišćenjem J-Net modela.

Dijagram relativnog odstupanja od centra trajektorije za jedan pun krug autonomne vožnje je predstavljen na slici 7.3. Osnovne karakteristike trajektorije se mogu podeliti u četiri kategorije: uglavnom ravan put ograničen sa bankinama, krivine ograničene crveno-belim prugama, most i delovi puta bez ikakve ograde gde je ivica puta definisana samo prašinom pored puta. Kao što se može videti sa slike 7.3, autonomna vožnja korišćenjem sva tri modela ima sličan obrazac. Modeli pokazuju da je autonomna vožnja gotovo bez oscilacija na ravnim delovima putanje. Međutim, u krivinama, odstupanje od centra trajektorije je bilo najveće. Na primer, velika odstupanja su u krivini nakon mosta – deo dijagrama sa slike 7.3 označen kao (c), postoje tri oštre krivine – slika 7.3 (d) i (b), dok je treća krivina najizazovnija. Dodatno, slika 6.9 pokazuje kako sve tri mreže imaju odstupanje na delu sa trećom krivinom, s tim da je kod AlexNet odstupanje najveće, a J-Net ima najbolji performans od sve tri mreže na ovom delu puta. Sa druge strane, J-Net ima više oscilacija tokom čitavog kruga, dok AlexNet ima najmanje oscilacija ako se posmatra čitav krug vožnje jer se vozilo nalazi na sredini putanje tokom najvećeg dela vožnje.



*Slika 7.3 Relativno odstupanje od centra trajektorije za jedan pun krug autonomne vožnje. Četiri glavne karakteristike puta: (a) uglavnom prav put oivičen bankinom, (b) krivine oivičene crveno-belim trakama, (c) most, (d) krivina oivičena prašinom (desno).*

Statistička analiza autonomne vožnje je predstavljena i putem histograma. Ova analiza je značajna za dugoročne testove. Kako bi se ispitale oscilacije, histogram relativnih odstupanja od centra trajektorije za jedan pun krug autonomne vožnje je predstavljen na slici 7.4. Histogram vožnje korišćenjem J-Net mreže je predstavljen na slici 7.4(a), gde se vidi da J-Net ima najmanje odstupanje u krivinama, dok su oscilacije od centra trajektorije najveće. Za PilotNet, slika 7.4(b), oscilacije su srednje jačine u odnosu na preostale dve mreže, ali model ima značajne oscilacije u krivinama, na obe, levu i desnu stranu, gde u oba pravca sporadična dešavanja imaju relativno odstupanje od centra trajektorije od skoro 100%. Histogram relativnog odstupanja od centra trajektorije za AlexNet je prikazan na slici 7.4(c). Vožnja korišćenjem AlexNet mreže je najstabilnija, sa najmanjim oscilacijama od centra trajektorije. Sa druge strane, postoje dešavanja sporadičnog odstupanja od centra trajektorije u jednom pravcu krivine. Međutim, ovo odstupanje je u okviru dozvoljenih granica, vozilo ne skreće sa puta, što je kriterijum kojim se definiše uspešna autonomna vožnja.



Slika 7.4 Histogram odstupanja od centra trajektorije za jedan pun krug autonomne vožnje.

Konačno se može zaključiti da su svi modeli daju zadovoljavajuće rezultate u autonomnoj vožnji, završivši čitav krug autonomne vožnje bez značajnog odstupanja od centra trajektorije. Razlike između autonomnih vožnji korišćenjem različitih modela su primetne, ali nisu velike.

**Tabela 7.4** Kvantitativna evaluacija autonomne vožnje korišćenje modela AlexNet, PilotNet i J-Net na platformi sa visokim performansama za simulator autonomne vožnje.

Autonomna vožnja	AlexNet	PilotNet	J-Net
Broj uspešnih krugova *	10	10	10
Kašnjenje	26.0 ms	24.1 ms	23.8 ms
Broj predikcija u sekundi	37	42	44

\* Autonomna vožnja je testirana na 10 uzastopnih krugova.

Bazirano na analizi računarske kompleksnosti, očekivano je da će J-Net imati najmanje kašnjenje i najveći broj ostvarenih predikcija u jednoj sekundi, u odnosu na preostala dva modela. Kvantitativna evaluacija verifikuje ovu tvrdnju, kao što se može videti u tabeli 7.4. Ova evaluacija je rađena na računarskoj platformi čije su performanse opisane na početku sekcije 7.1, što je platforma visokih performansi na kojoj je pokrenuto okruženje simulatora. Kako je reprezentativna putanja po kojoj se obavlja autonomna vožnja zatvorena petlja, meri se broj uspešnih uzastopnih krugova, u deset krugova ukupno. Korišćenjem sva tri modela obavljena je autonomna vožnja sa uspešnim održanjem vozila u kolovoznoj traci tokom čitave vožnje. Za merenje kašnjenja, izračunavano je vreme između



dve uzastopne predikcije. Ovaj broj varira tokom čitavog kruga autonomne vožnje, tako da je za kašnjenje uzimana srednja vrednost tokom jednog kruga vožnje. Broj predikcija u sekundi se izračunava brojanjem predikovanih uglova okretanja volana u periodu od jedne sekunde.

Na osnovu obavljenih merenja čiji su rezultati predstavljeni u tabeli 7.4, može se zaključiti da je korišćenje J-Net modela u fazi zaključivanja u realnom vremenu 30% brže od korišćenja AlexNet modela na platformi sa visokim performansama na kojoj se nalazi simulator autonomne vožnje, što se vidi iz kolone broj predikcija u sekundi u tabeli 7.4. Međutim, ukoliko bi se koristio skalarni procesor za zaključivanje u realnom vremenu, očekivala bi se veća razlika u brzinama izvršavanja. U ovom eksperimentu gde je korišćen simulator autonomne vožnje, platforma za zaključivanje u realnom vremenu je kompjuter visokih performansi sa GPU jedinicom koja omogućava paralelizaciju izvršavanja operacija. Iz ovog razloga, s obzirom da je arhitektura korišćenih neuralnih mreža ima više razlike u površini slojeva nego u dubini, većina operacija se obavlja u paraleli, pa je i razlika u brzini izvršavanja algoritma u fazi zaključivanja posledica sekvencionalnosti, što je proporcionalno dubini mreža koje se posmatraju.

Ukoliko bi operacije bile vršene samo skalarnim procesorima, moglo bi se očekivati da bi brzine zaključivanja bile značajno različite, uporedivo sa kapacitetom projektovanih mreža, brojem parametara i operacija nad njima. Realna implementacija J-Net mreže je namenjena namenskim računarskim platformama za autonomnu vožnju, robo-vozilima.

### 7.3. Evaluacija modela J-Net na novom delu kolovozne trake

Još jedna verifikacija autonomne vožnje je izvršena u simulatoru na novom delu kolovozne trake reprezentativne putanje. U ovom slučaju, razlika reprezentativne putanje u odnosu na putanju koja je korišćena za prikupljanje podataka za treniranje neuralnih mreža je u tome što je prvih 20% konačne putanje potpuno novo. Odnosno iz skupa podataka za treniranje modela isključeni su podaci dobijeni sa prvih 20% putanje definisane u simulatoru, kako bi se obezbedila verifikacija uspešnosti konačnih modela i na novim delovima puta, nikada viđenim u procesu treniranja. Ovako istrenirana mreža je validirana u simulatoru autonomne vožnje. Autonomno održanje vozila u kolovoznoj traci je uspešno. Međutim, performanse autonomne vožnje degradiraju u smislu odstupanja vozila od centralne linije puta, što je i očekivan rezultat. Srednja kvadratna greška odstupanja od centralne linije puta iznosi 45,63 m<sup>2</sup>, a apsolutna greška 0,43 m, tabela 7.5.

**Tabela 7.5** Evaluacija autonomne vožnje korišćenjem J-Net modela na novom delu kolovozne trake

<b>Autonomna vožnja</b>	<b>J-Net</b>	<b>J-Net-novo treniranje</b>
Autonomno održanje vozila u kolovoznoj traci	uspešno	uspešno
Srednja kvadratna greška (MSE) [m <sup>2</sup> ]	7,56	45,63
Apsolutna greška (IAE) [m]	0,22	0,43

## 8. Verifikacija autonomne vožnje u realnim uslovima

Za potrebe verifikacije autonomne vožnje u realnim uslovima, projektovan je i realizovan sistem za autonomnu vožnju. Korišćenjem sistema za autonomnu vožnju moguće je testirati autonomnu vožnju samovozeće robotske platforme, u daljem tekstu robo-vozila, po zadatoj putanji, odnosno kolovoznoj traci u realnim uslovima. Verifikacija autonomne vožnje u realnim uslovima ali u pojednostavljenim scenarijima, često laboratorijskim uslovima, je uobičajen način da se izvrši validacija projektovanih algoritama i rešenja za autonomnu vožnju [106]-[109].

### 8.1. Opis sistema za autonomnu vožnju

Cilj projektovanja i realizacije sistema za autonomnu vožnju je da se, korišćenjem ovog sistema, u realnim uslovima izvrši validacija modela nove arhitekture duboke neuralne mreže, J-Net, za autonomno održanje vozila u kolovoznoj traci.

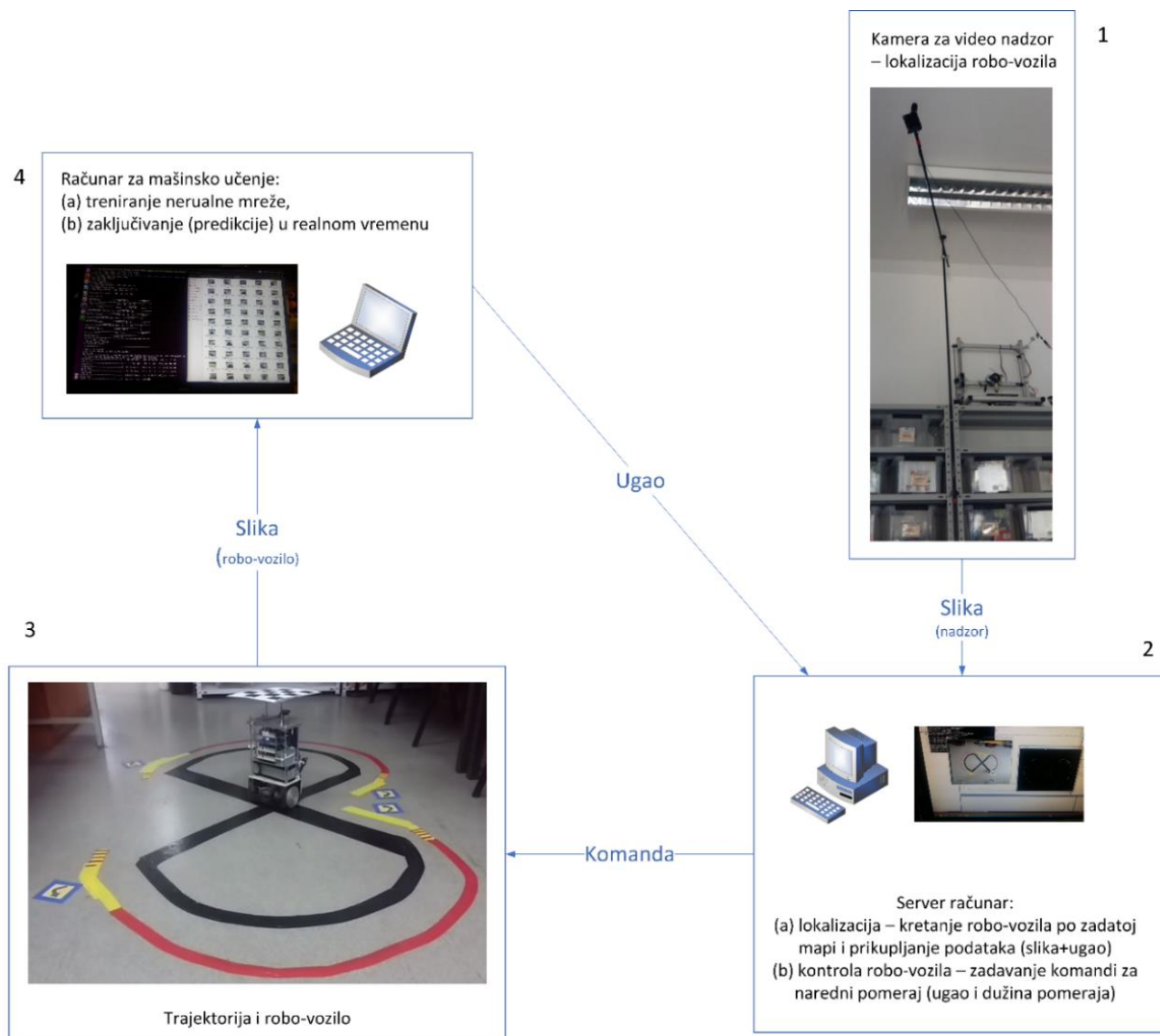
Projektovani sistem za autonomnu vožnju je kompleksan sistem koji se može koristiti u dva režima rada: automatski kontrolisano kretanje vozila i autonomno kretanje vozila, slika 8.1. U prvom režimu rada, robo-vozilo je automatski kontrolisano, tako što se kreće po zadatoj mapi koordinata. Automatski kontrolisani režim rada je projektovan u svrhu automatizovanog prikupljanja i labeliranja uzoraka koji čine bazu podataka za treniranje modela neuralne mreže za učenje od-kraja-do-kraja. U režimu rada za automatski kontrolisano kretanje vozila, vrši se akvizicija slika sa kamere montirane na robo-vozilu uparene sa informacijom o uglu kretanja vozila u tom trenutku. Ovako dobijeni podatak predstavlja jedan uzorak novokreirane baze podataka.

U drugom režimu rada, sistem se koristi za autonomnu vožnju u realnom vremenu, gde se slika sa robota prosleđuje algoritmu mašinskog učenja koji samo na osnovu slike kao ulaznog podatka donosi zaključak o narednom pomeraju vozila, odnosno o uglu pod kojim treba rotirati točkove u odnosu na uzdužnu osu vozila. U autonomnom režimu rada, brzina vozila je postavljena na konstantnu vrednost, tako da se upravljanje vozilom vrši isključivo zadavanjem narednog ugla pod kojim će se kretati vozilo. Dodatno, kretanje vozila je kontinuirano, tako da zaustavljanje vozila u slučaju da naiđe na prepreku izlazi iz opsega definisanog problema.



Slika 8.1 Dva režima rada sistema za autonomnu vožnju.

Centralne komponente sistema za autonomnu vožnju su: robo-vozilo koje se kreće tako što mu se zadaje komanda za pomeraj, a to su dužina pomeraja i rotacioni ugao u odnosu na tekuću poziciju, server računar koji vrši kontrolu nad vozilom, pri čemu ova kontrola može biti automatsko upravljanje vozilom ili autonomna vožnja. Automatsko upravljanje se postiže zadavanjem mape po kojoj se kreće robo-vozilo. Autonomna vožnja se ostvaruje u sistemu autonomne vožnje koji čine: server računar koji prosleđuje zaključke algoritma za mašinsko učenje o narednom uglu kretanja robo-vozila, računar na kome se izvršava algoritam mašinskog učenja koji vrši zadavanje narednog ugla u realnom vremenu, i sistem za video nadzor koji čini kamera povezana sa server računarem čija je uloga da vrši lokalizaciju robo-vozila. Na slici 8.2. prikazane su osnovne komponente sistema za autonomnu vožnju koji je razvijen na Elektrotehničkom fakultetu Univerziteta u Beogradu na Katedri za elektroniku: (1) kamera za lokalizaciju, (2) server, (3) robo-vozilo sa montiranom kamerom, kolovozna traka i (4) računar za mašinsko učenje.



*Slika 8.2 Osnovne komponente eksperimentalne postavke sistema za autonomnu vožnju: (1) kamera za lokalizaciju, (2) server računar, (3) robo-vozilo na definisanoj kolovoznoj traci i (4) računar na kome se vrši zaključivanje u realnom vremenu.*



Osnovne komponente eksperimentalne postavke sistema za autonomnu vožnju su:

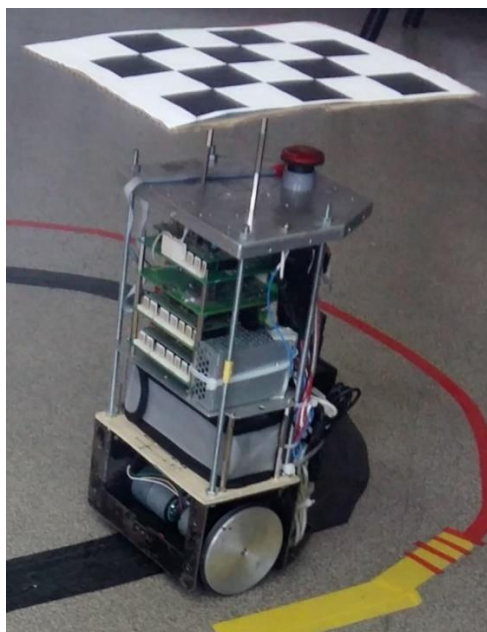
- (1) Kamera. Kamera za video nadzor se nalazi u visini plafona, pri čemu njen ugao vidljivosti obuhvata površinu poda laboratorije na kome se nalazi putanja, odnosno kolovozna traka, po kojoj je se kreće robo-vozilo.
- (2) Server računar. Na server računaru je pokrenut program koji upravlja robotom na dva nezavisna načina: prvo, koristeći predefinisanu mapu koordinata po kojima robot treba da se kreće i informaciju o poziciji robota dobijenu od sistema za lokalizaciju, i drugo koristeći izlaz algoritma za mašinsko učenje kojim se definiše sledeći pomeraj robota, odnosno ugao pod kojim je potrebno zaokrenuti robota. Server računar je povezan sa kamerom za video nadzor putem USB serijske komunikacije, sa računarom na robo-vozilu putem bežične Bluetooth komunikacije i sa računarom na kome se izvršava algoritam mašinskog učenja WiFi komunikacijom.
- (3) Trajektorija. Trajektorija (kolovozna traka) po kojoj se kreće robo-vozilo je u obliku broja osam, tako da se sastoji od više različitih delova: deo trajektorije sa ravnom putanjom, dva polukruga pri čemu se vrši blago skretanje u levo ili desno, i četiri veoma oštre krivine, po dve u levo i u desno.
- (4) Računar za mašinsko učenje. Računar za mašinsko učenje ima dvojak u ulogu. Na računaru se vrši treniranje duboke neuralne mreže korišćenjem baze slika koja je prikupljena korišćenjem sistema za autonomnu vožnju u prvom režimu, gde se robot kreće po predefinisanoj mapi koordinata. Druga uloga ovog računara je da se na njemu vrši zaključivanje algoritma mašinskog učenja, gde se u realnom vremenu slika koju robot snimi prosleđuje na ulaz istrenirane neuralne mreže i kao izlaz se dobija informacija o narednom pomeraju, odnosno uglu kretanja robo-vozila. Informacija o vrednosti prediktovanog ugla se prosleđuje do algoritma na server računaru koji dalje zadaje komandu robo-vozilu.

## 8.2. Samovozeća robotska platforma

Za eksperimentalno testiranje je korišćena samovozeća robotska platforma sa tri točka J-BOT, koji se u ovom radu naziva robo-vozilo, slika 8.3. Jedan točak je slobodan, dok se preostala dva točka pokreću jednosmernim motorima sa ugrađenim inkrementalnim enkoderima. Konstrukcija robota obezbeđuje da pored translatornog kretanja, robot ima mogućnost rotacije oko svoje centralne ose za proizvoljni ugao. Podržano je deo po deo linearno i kontinualno kretanje. Robotska platforma je opremljena jednom kamerom i ultrazvučnim i infracrvenim daljinomerima koji se mogu koristiti kao pomoć u lokalizaciji i mapiranju.

Platforma poseduje dva nivoa upravljanja. Primarni kontroler platforme je mikrokontrolerski uređaj koji upravlja pogonom motora i očitava sve senzore u realnom vremenu. Primarni kontroler poseduje žični RS-485 interfejs kojim se ostvaruje komunikacija u okviru samog robota i bežični Bluetooth link kojim je moguće ostvariti daljinsku kontrolu i upravljanje.

Pored primarnog kontrolera na robotu se nalazi i sekundarni kontroler koji je zasnovan na PC računaru u kompaktnom kućištu prilagođenom dimenzijama robota. Na PC računaru je moguće izvršavanje bilo koje standardne aplikacije pod Windows operativnim sistemom. Računar poseduje sopstveni WiFi link pogodan za ostvarivanje veze i prenos većih količina podataka između robota i serverskog računara. Prvenstvena namena sekundarnog kontrolera je akvizicija, skladištenje i prenos podataka sa kamere.



*Slika 8.3 Samovozeća robotska platforma, robo-vozilo.*

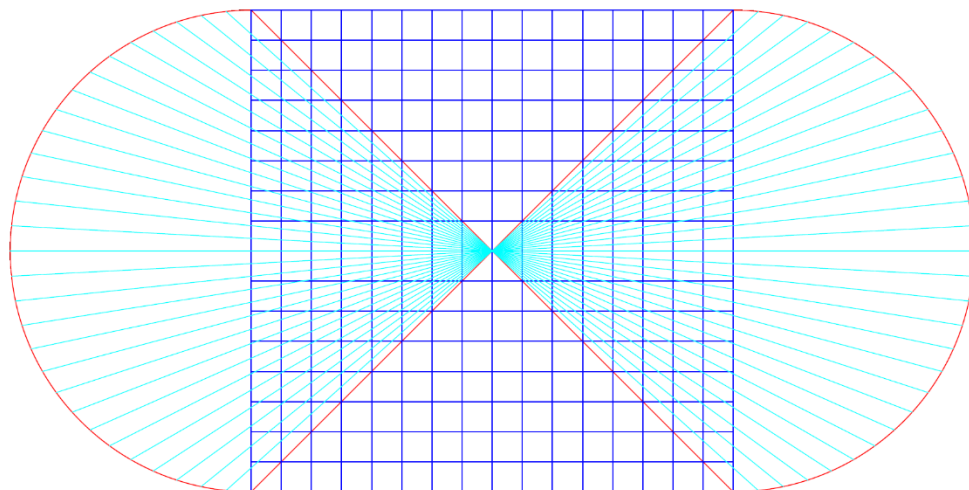
### **8.3. Zadana putanja za verifikaciju autonomne vožnje**

Trajektorija izabrana za eksperimentalnu proveru autonomne vožnje korišćenjem novog modela duboke neuralne mreže je u obliku broja osam, slika 8.4. Ovakva putanja, iako na prvi pogled jednostavna, sadrži sve značajne karakteristike neophodne za verifikaciju uspešne autonomne vožnje. Karakteristične komponente zadate putanje su pravolinijsko kretanje, polukružno kretanje u levo i desno, pri čemu vozilo treba da ostvari blag zaokret i četiri oštre krivine, po dve u levo i u desno. Oštre krivine predstavljaju najzahtevniji deo putanje zadate trajektorije.



*Slika 8.4 Zadana putanja za verifikaciju autonomne vožnje u realnim uslovima.*

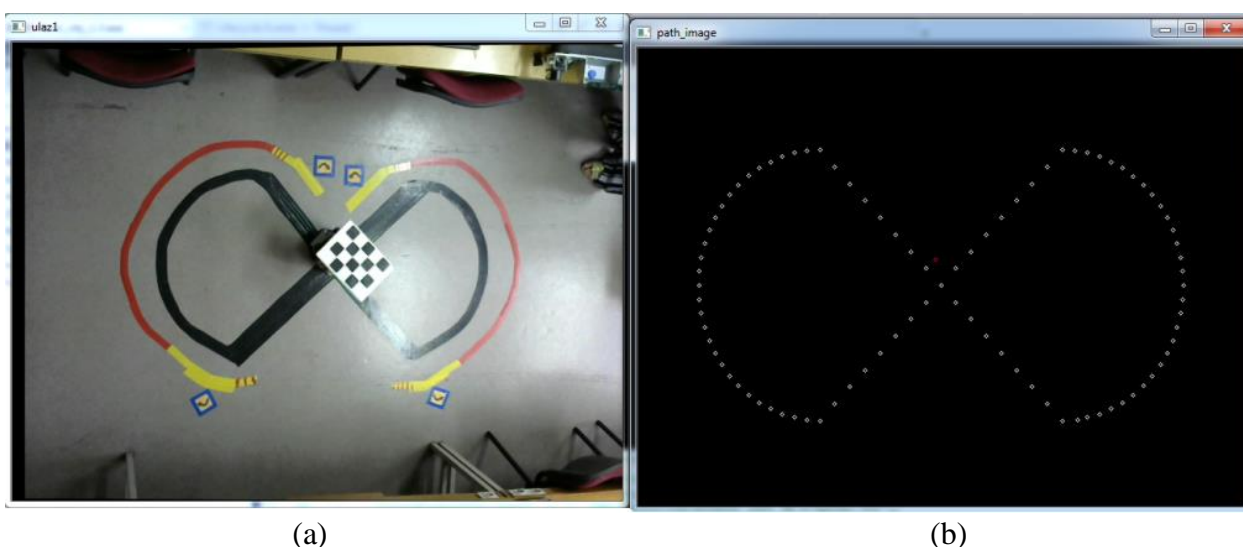
Putanja se zadaje sa nizom tačaka po kojima se kreće robot u automatskom režimu. Kako bi se kreirala trajektorija u obliku broja osam, niz koordinata čine 92 tačke, slika 8.5. Deo putanje definisan za pravolinijsko kretanje je određen mrežom kvadrata gde se tačke preseka ove mreže i projektovane trajektorije (označena crvenom linijom) uzimaju kao koordinate. Ove koordinate su na ekvidistalnom rastojanju. Polukružno kretanje je definisano tako što su sve koordinate na polukružnici međusobno udaljene pod uglom od 3 stepena. Ovako definisan niz koordinata čini mapu po kojoj se ostvaruje željeno automatsko kretanje robota.



Slika 8.5 Izbor koordinata putanje za eksperimentalnu verifikaciju autonomne vožnje.

## 8.4. Sistem za automatsku akviziciju i laberiranje slika – kreiranje baze podataka

Kreiranje baze slika kojom se treniraju modeli dubokih neuralnih mreža za učenje od-kraja-do-kraja koji se koriste za autonomnu vožnju, podrazumeva akviziciju slika sa kamere montirane na robo-vozilu uparene sa uglom pod kojim vozilo treba dalje da se kreće imajući u vidu stanje na kolovozu koje kamera vidi u tom trenutku. U autonomnom režimu rada, slika je ulazni podatak u neuralnu mrežu, a ugao pod kojim je potrebno da se robot kreće do naredne tačke je željena vrednost izlaza iz neuralne mreže.



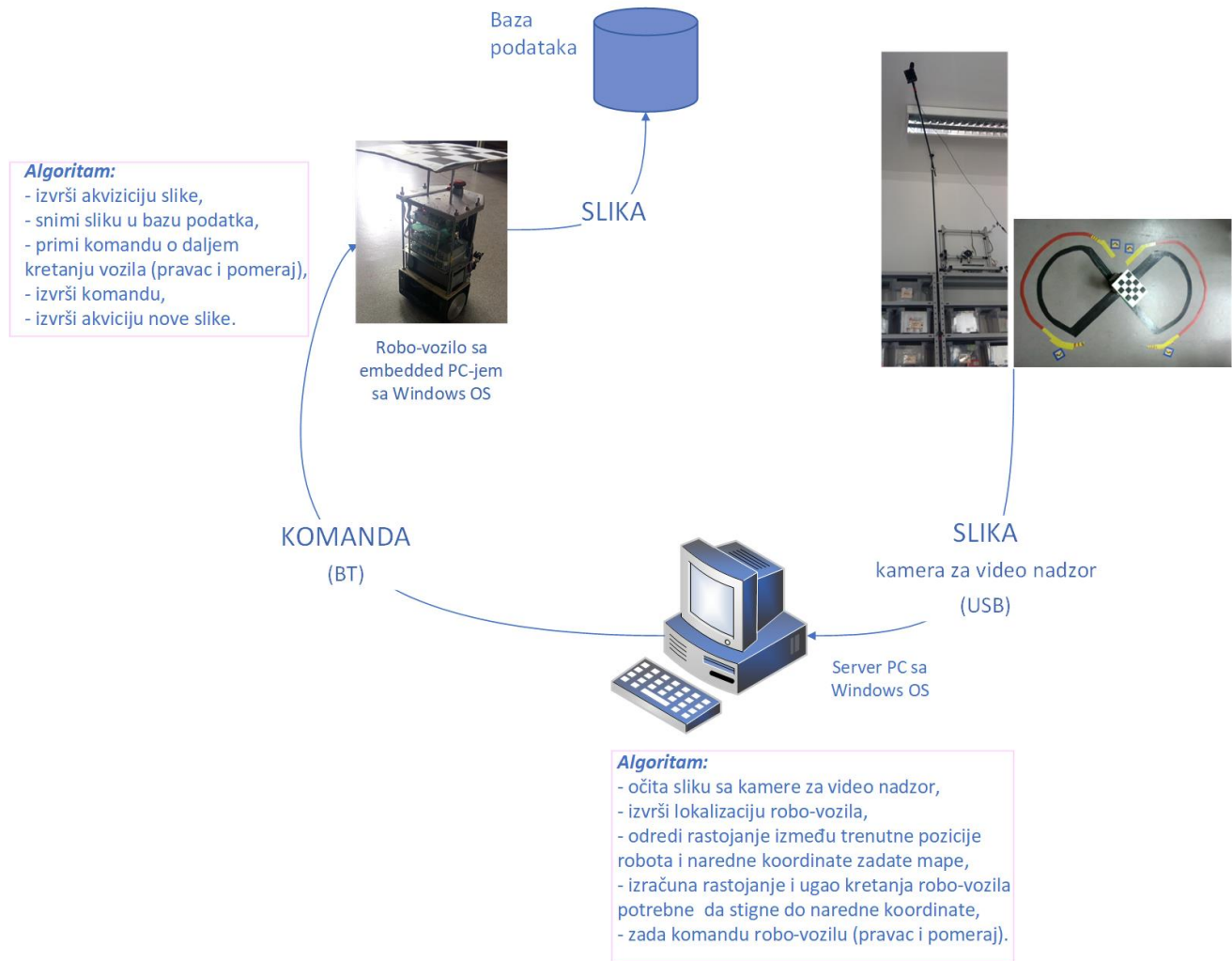
(a)

(b)

Slika 8.6 (a) Pogled na definisanu kolovoznu traku sa kamere za video nadzor; (b) Mapa koordinata kojom se definiše automatsko kretanje vozila.

Kako bi se ostvarilo automatsko označavanje prikupljenih slika sa odgovarajućim uglom, kreiran je sistem za automatsko kretanje robota po zadatoj mapi. Lokalizacija robota na definisanoj putanji se vrši korišćenjem kamere za video nadzor koja detektuje šahovsko polje koje se nalazi na vrhu robo-vozila. Detekcijom položaja šahovskog polja se definiše apsolutna koordinata robota u datom prostoru. Rešenje za lokalizaciju u našem sistemu za autonomnu vožnju je inspirisano sistemom za lokalizaciju mobilnih robota korišćenjem mašinske vizije [110]. Na slici 8.6(a) je prikazan pogled s kamere za video nadzor na definisanu putanju za autonomnu vožnju, dok je na slici 8.6(b) prikazan skup koordinata koje definišu putanju.

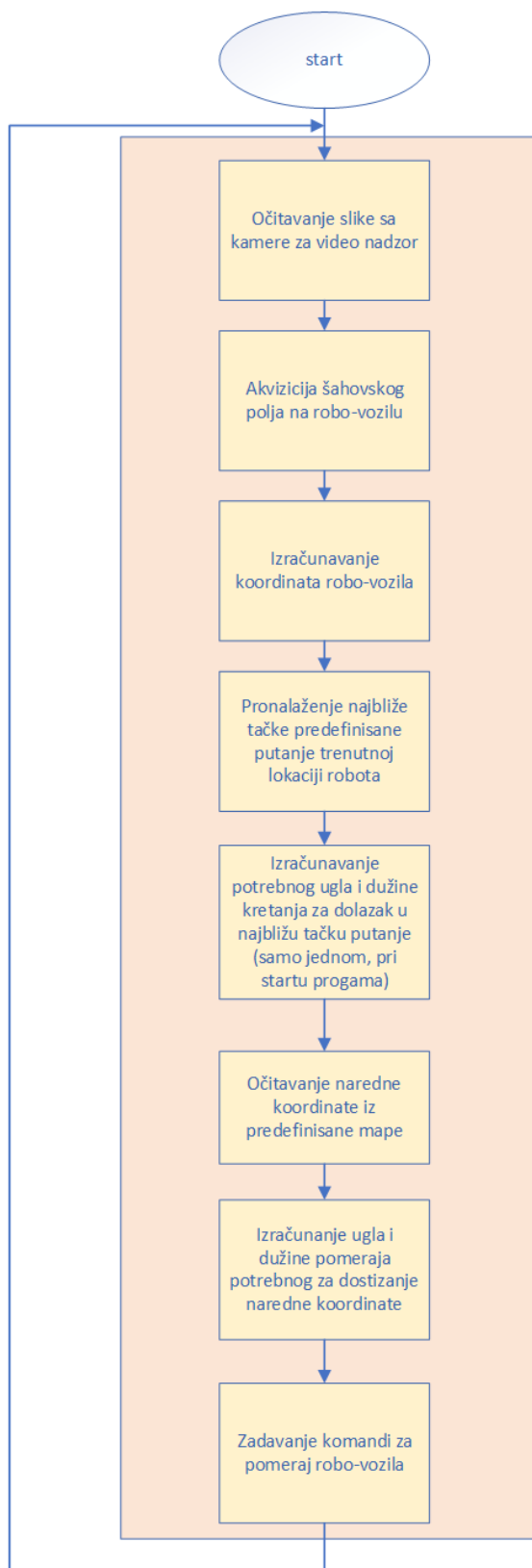
Blok dijagram sistema koji je projektovan za kreiranje baze podataka, odnosno akviziciju slika sa kamere montirane na robotu i snimanje ovih slika u jednu bazu, pri čemu se svaka slika automatski označava sa uglom kretanja vozila koji je zadat u tom trenutku, je prikazana na slici 8.7.



Slika 8.7 Blok dijagram sistema za kreiranje baze podataka (režim automatskog kretanja).

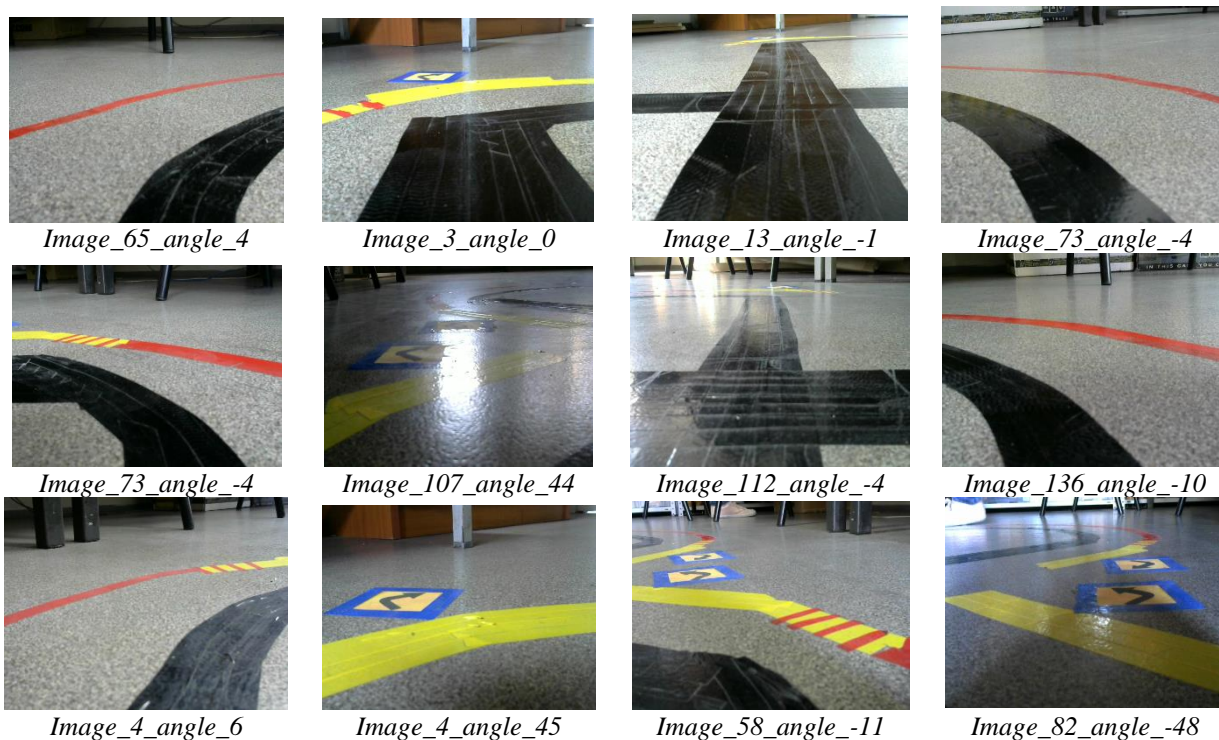
Algoritam koji se izvršava na server računaru, a koji upravlja automatskim kretanjem vozila kada se vrši akvizicija slika, odnosno kreiranje baze podataka, predstavljen je na slici 8.8. Najpre se očitava slika sa kamere za video nadzor, koja je sa server računarem povezana putem žičane serijske veze. Na akviziranoj slici se detektuje šahovsko polje koje je nalazi na robo-vozilu, na osnovu čije pozicije na akviziranoj slici se računa trenutna koordinata vozila. Za dolazak u prvu tačku zadate mape algoritam pronalazi najbližu tačku na definisanoj putanji za kretanje robo-vozila trenutnoj lokaciji vozila. Izračunava se potreban ugao okretanja vozila i dužina puta za dolazak u narednu tačku definisane mape. Za svaku sledeću tačku definisane mape se određuje samo rastojanje i potreban ugao

kretanja od trenutne pozicije do naredne tačke iz definisane mape. Kada se ugao (pravac kretanja) i dužina kretanja izračunaju, komanda se putem Bluetooth komunikacije šalje do mikroprocesorske jedinice robo-vozila koja izvršava komandu za pomeraj vozila.



Slika 8.8 Algoritam za lokalizaciju i automatsko kretanje po zadatoj mapi.





Slika 8.9 Primeri akviziranih slika za autonomnu vožnju u realnim uslovima.

Slike su akvizirane u različitim uslovima osvetljenja i dobima dana, tokom vidljivog dela dana sa prirodnom svetlošću i uveče pod električnim osvetljenjem. Od interesa je prikupiti uzorke u različitim svetlosnim uslovima kako bi trenirana mreža mogla adekvatno da odgovori na slične uslove koji postoje prilikom autonomne vožnje u realnom vremenu. Na primer, kod uzorka *Image\_107\_angle\_44*, druga slika u srednjem redu na slici 8.9, svetlost velikog intenziteta obasjava jedan deo trajektorije, što se smatra otežanim uslovom u toku faze zaključivanja algoritma mašinskog učenja.

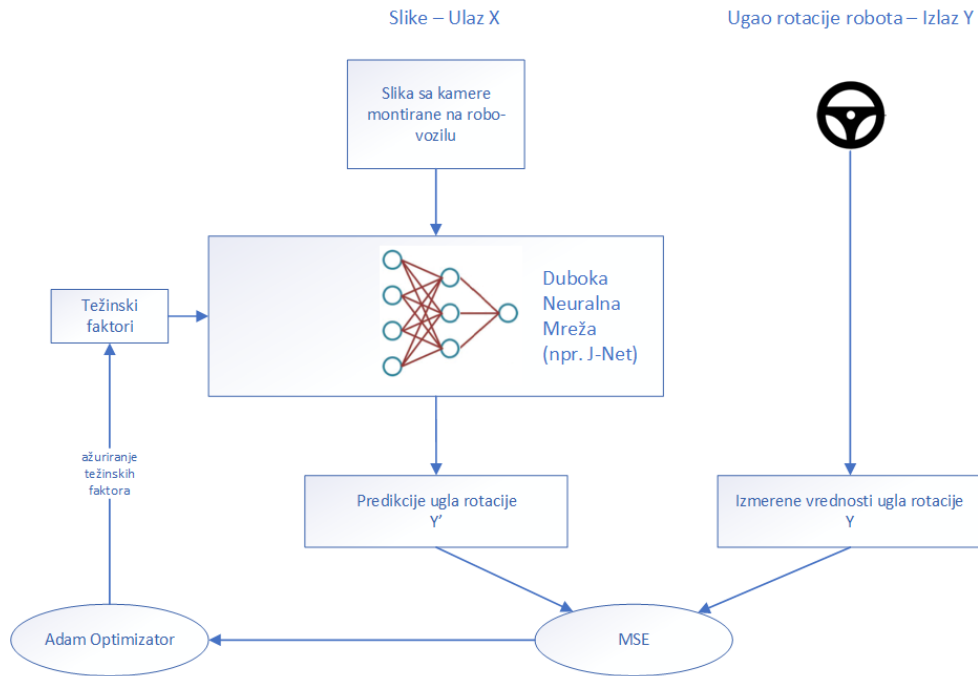
**Tabela 8.1** Baza podataka za treniranje duboke neuralne mreže za autonomnu vožnju u realnim uslovima.

	Treniranje	Validacija	Ukupno
Broj uzoraka	725	172	897
Procenat od ukupne baze podataka	81%	19%	100%

Izvršena je akvizicija ukupno 897 podataka. Primeri prikupljenih podataka su prikazani na slici 8.9. Podaci su podeljeni u dve kategorije, 80% podataka se koristi za treniranje duboke neuralne mreže, a 20% za validaciju, tabela 8.1.

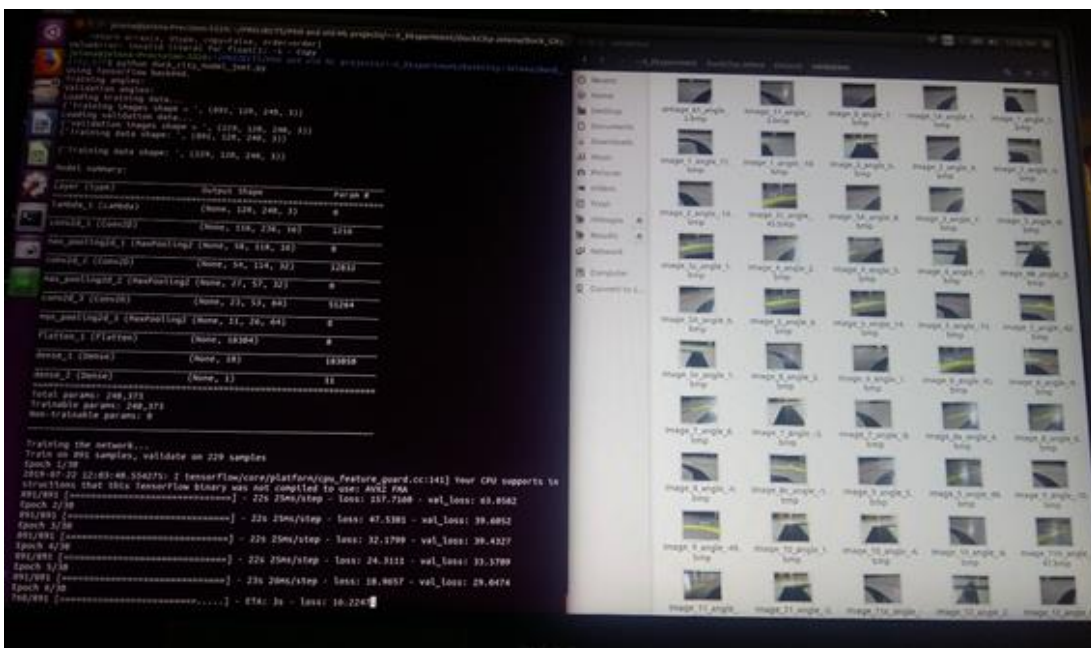
## 8.5. Treniranje duboke neuralne mreže

Treniranje dubokih neuralnih mreža od-kraja-do-kraja za autonomnu vožnju u realnim uslovima se vrši na isti način kao što je to bilo učinjeno za autonomnu vožnju u simulatoru. Jedina razlika je što se u realnim uslovima koriste slike sa samo jedne kamere koja je montirana na dno robo-vozila, slika 8.3. Blok dijagram treniranja prikazan je na slici 8.10.



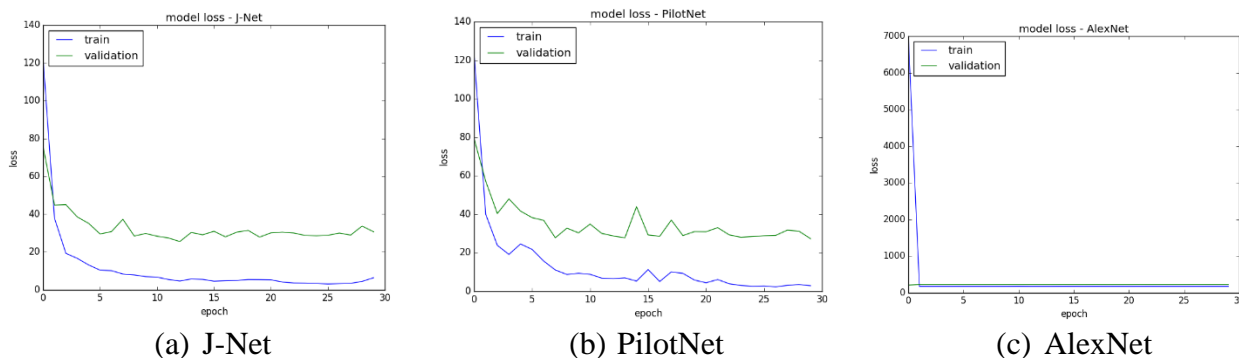
Slika 8.10 Blok dijagram treniranja neuralne mreže za autonomnu vožnju u realnim uslovima.

Treniranje je obavljeno na računaru sa Linux operativnim sistemom, koristeći TensorFlow biblioteku instaliranu sa CUDA akceleracijom. Postupak treniranja prikazan je na slici 8.11.



Slika 8.11 Treniranje duboke neuralne mreže, snimak ekrana.

Originalna veličina slike sa kamere montirane na robo-vozilu je 640 x 480 piksela. J-Net i PilotNet je moguće istrenirati na opisanom računaru, međutim AlexNet u slučaju korišćenja ulaznog podatka veličine 640 x 480 piksela ima izuzetno puno parametara, više stotina miliona, i nije bilo moguće istrenirati AlexNet koristeći dati računar. Iz ovog razloga, slici je promenjena veličina na 120 x 240 piksela. Kako bi poređenje rezultata bilo adekvatno, sve tri neuralne mreže, J-Net, PilotNet i AlexNet su trenirane sa ulazom veličine 120 x 240 piksela. Treniranje je vršeno u 30 epoha. Dijagrami gubitaka tokom treniranja sve tri mreže, J-Net, PilotNet i AlexNet su prikazani na slici 8.12.



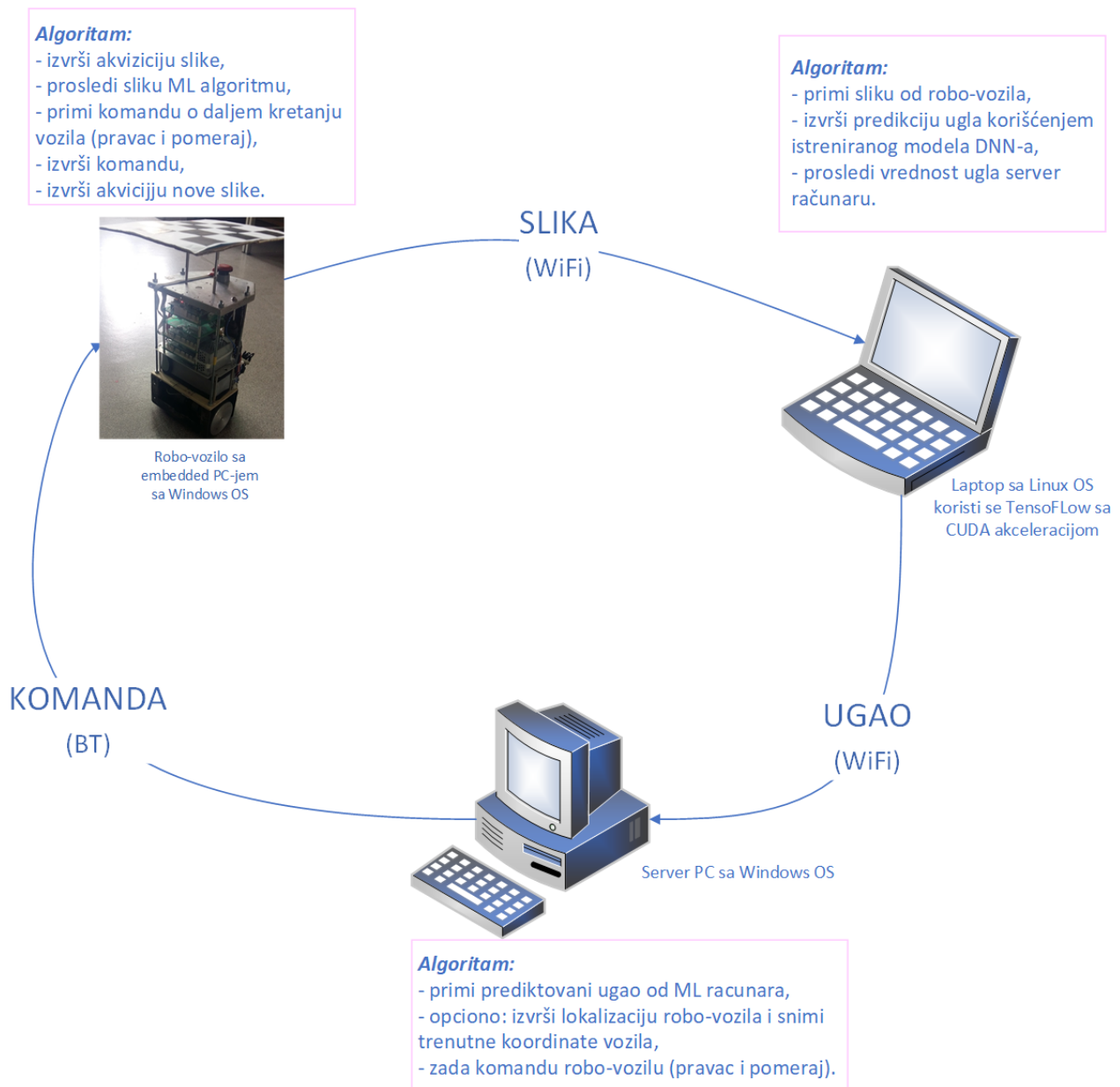
Slika 8.12 Dijagrami gubitka pri treniranju neuralnih mreža u 30 epoha.

## 8.6. Autonomna vožnja u realnom vremenu

Eksperimentalna postavka sistema za autonomnu vožnju u realnom vremenu se sastoji od tri glavne softverske komponente, pri čemu se svaka od ovih komponenti u izvedenom eksperimentu izvršava na različitom računaru. Robo-vozilo najpre vrši akviziciju slike sa kamere koja je montirana na dnu vozila tako da gleda u put pred vozilom. Primeri slike koja se akvizira sa kamere montirane na vozilu su dati na slici 8.9. Slika uslikana kamerom robo-vozila se u realnom vremenu prosleđuje algoritmu mašinskog učenja koji na osnovu samo slike kao ulaznog podatka donosi predikciju o uglu kretanja vozila, kako bi vozilo ostalo na definisanoj kolovoznoj traci. Brzina vozila je u postavci problema definisana kao konstanta. Brzina se definiše kroz pomeraj vozila, tako da je pomeraj vozila u svakom trenutku jednak i iznosi 30 mm. Slika dobijena sa kamere montirane na robo-vozilu se prenosi do računara na kome se vrši predikcija WiFi komunikacijom.

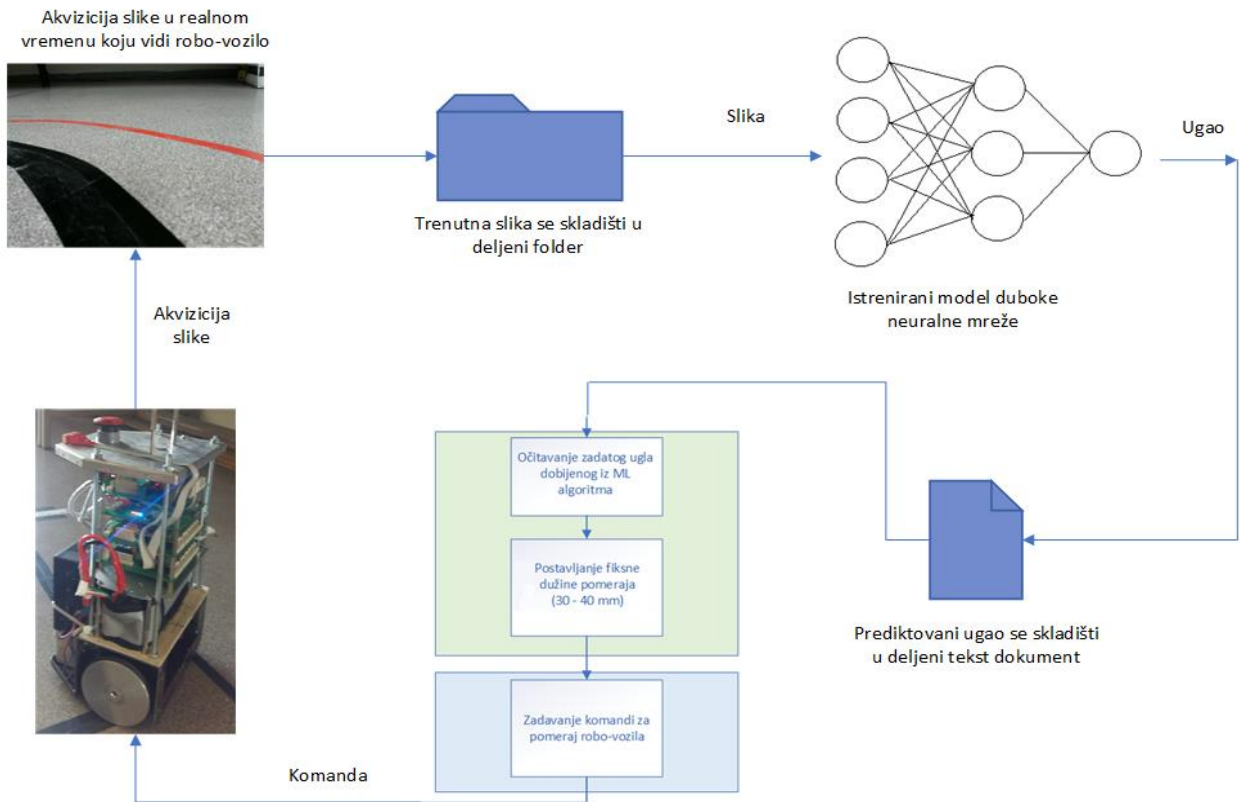
Blok dijagram sistema za autonomnu vožnju sa definisanim softverskom komponentama je prikazan na slici 8.13. Algoritam mašinskog učenja za postizanje autonomne vožnje u realnom vremenu prosleđuje sliku sa robo-vozila na ulaz u istrenirani model duboke neuralne mreže i izlaz iz ovog modela, odnosno prediktovani ugao smešta u tekstualni fajl. Centralni računar, server, vrši prijem informacije o prediktovanom uglu očitavanjem vrednosti ugla iz tekstualnog fajla, i zadaje komandu robo-vozilu o pomeraju, pri čemu je pomeraj definisan dužinom pomeraja i uglom rotacije u odnosu na trenutnu poziciju vozila. Komanda se sa centralnog računara do robo-vozila prenosi Bluetooth komunikacijom. Vozilo po dobijanju komande izvršava komandu, šalje novu sliku algoritmu mašinskog učenja i čeka novu informaciju o sledećem pomeraju. Na samom robo-vozilu nalazi se mikrokontrolerski uređaj koji upravlja pogonom jednosmernog motora koji upravljaju točkovima vozila.





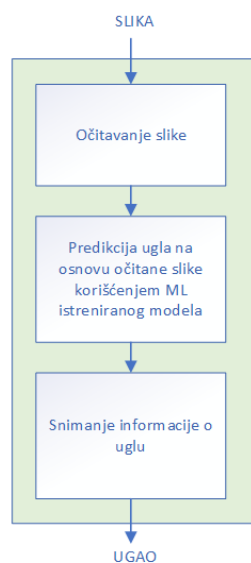
Slika 8.13 Blok dijagram sistema za autonomnu vožnju robo-vozila u realnom vremenu.

Detaljniji dijagram toka podataka i naredbi tokom autonomne vožnje u realnom vremenu je prikazan na slici 8.14. Slika se akvizira sa kamere i smešta u deljenu datoteku na Windows operativnom sistemu kompaktne PC platforme koja je deo robo-vozila. Pristup ovom deljenom folderu sa trenutnom slikom sa kamere ima Linux operativni sistem računara za mašinsko učenje. Deljenje foldera između Windows i Linux operativnih sistema predstavlja manji inženjerski izazov, koji je u ovom slučaju rešen korišćenjem Samba softverskog paketa [111]. Slika se potom dovodi na ulaz istreniranog modela mašinskog učenja čiji je izlaz ugao okretanja vozila. Informacija o uglu se upisuje u jedan tekstualni dokument, koji se, poput slike, nalazi u deljenoj datoteci koja je u slučaju informacije o uglu smeštena na Windows operativnom sistemu server računara. Na server računaru, očitava se vrednost ugla i na osnovu toga se zadaje nova komanda robo-vozilu. Tok protoka podataka: slika – model mašinskog učenja – ugao okretanja vozila se obavlja u realnom vremenu i u svakom trenutku samo jedna slika i jedna informacija o uglu okretanja vozila se nalazi u zajedničkim deljenim datotekama, čime je osiguran rad u realnom vremenu sa validnim podacima.

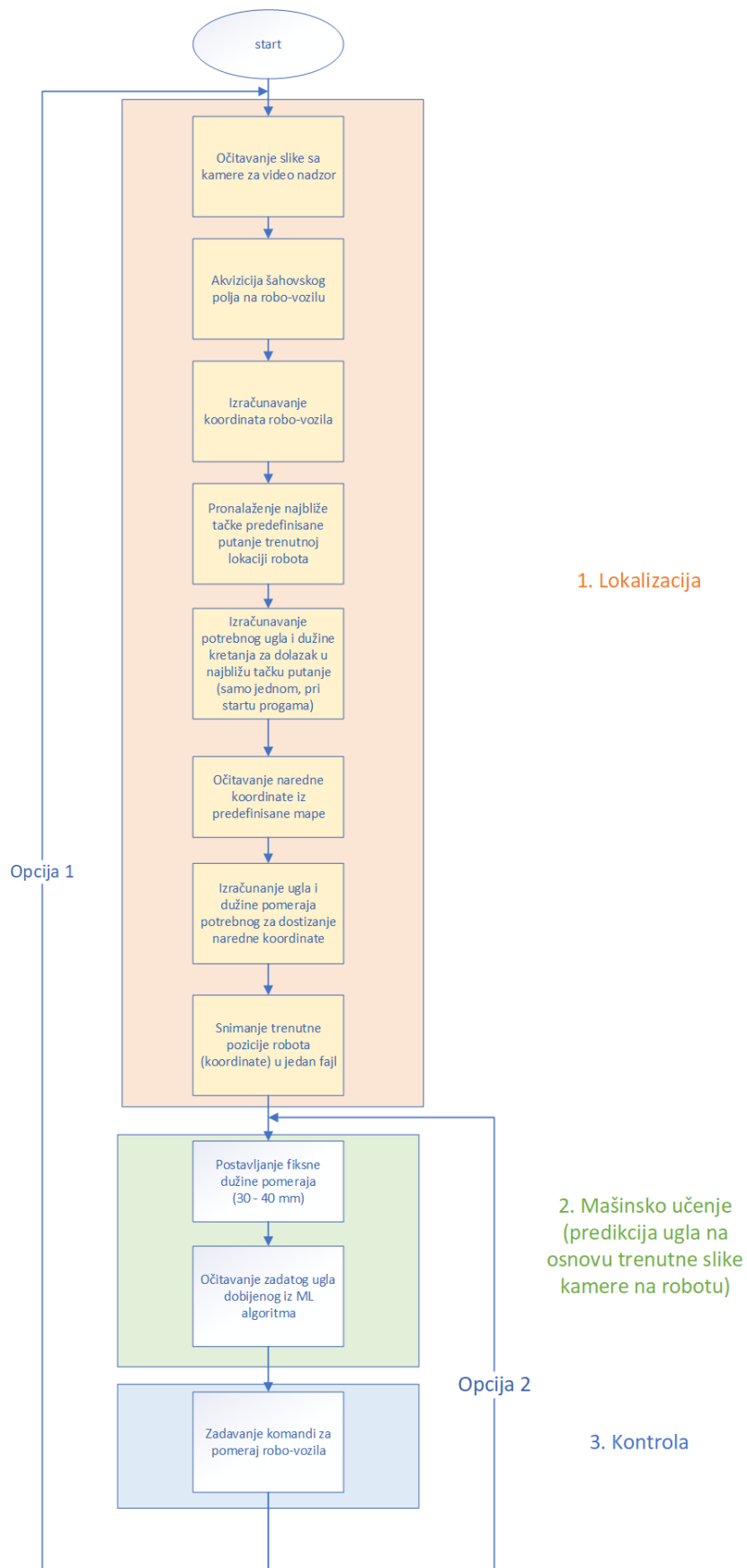


Slika 8.14 Dijagram toka podataka i naredbi tokom autonomne vožnje u realnom vremenu.

Algoritam zaključivanja na računaru za mašinsko učenje je prikazan na slici 8.15. Slika očitana sa kamere robo-vozila se dovodi na ulaz istreniranog modela za mašinsko učenje, koji donosi zaključak o narednom uglu okretanja vozila. Algoritam se izvršava kontinualno u beskonačnoj petlji, kontinuirano uzima sliku dobijenu sa robo-vozila i vrši predikciju narednog ugla, odnosno definiše ugao točkova u odnosu na osu vozila. Podatak o zaključenom uglu se smešta u tekstualni dokument koji se nalazi u deljenom folderu na server računaru. Izvršavanje algoritma za mašinsko učenje se vrši na računaru na kome je instaliran Linux operativni sistem i TensorFlow biblioteka za mašinsko učenje sa CUDA akceleracijom.



Slika 8.15 Algoritam zaključivanja na računaru za mašinsko učenje.



Slika 8.16 Algoritam na server računaru kojim se zadaje pomeraj robo-vozilu.



## 8.7. Rezultati verifikacije autonomne vožnje u realnim uslovima na zadatoj kolovoznoj traci

U realnom sistemu za autonomnu vožnju, izvršena je verifikacija autonomne vožnje na sličan način kao što je to bilo izvršeno u simulatoru. Tri duboke konvolucione neuralne mreže, J-Net, PilotNet i AlexNet su trenirane korišćenjem baze podataka kreirane metodom opisanom u sekciji 8.4. Ovako istrenirani modeli su korišćeni za autonomnu vožnju robo-vozila po definisanoj trajektoriji. Autonomna vožnja u našem sistemu je opisana u sekciji 8.6.

### 8.7.1. Računarska kompleksnost modela dubokih neuralnih mreža

Rezultati pokazuju da je moguće ostvariti autonomno održanje vozila u kolovoznoj traci korišćenjem sva tri modela dubokih neuralnih mreža za učenje od-kraja-do-kraja. Očekivano, za J-Net model je potrebno izvršiti najmanje operacija u realnom vremenu i trenirani model J-Net ima najmanje zauzeće memorije, 3 MB, dok je, takođe očekivano AlexNet model sa najvećim brojem operacija i trenarabilnih parametara, preko 200 miliona, i zauzima čak 2,4 GB memorijskog prostora. Broj operacija, trenarabilnih parametara i veličina istreniranih modela, osobine koje se zajednički nazivaju kapacitet modela, za J-Net, PilotNet i AlexNet su prikazani u tabeli 8.2.

**Tabela 8.2** Karakteristike treniranih modela u realnim uslovima, J-Net, PilotNet, and AlexNet.

	<b>J-Net</b>	<b>PilotNet</b>	<b>AlexNet</b>
Ukupan broj parametara *	248,373	1,314,619	200,787,281
Broj operacija (množenje i sabiranje)	248,37 k	1,31 m	200,78 m
Veličina istreniranog modela	3 MB	15.8 MB	2.4 GB

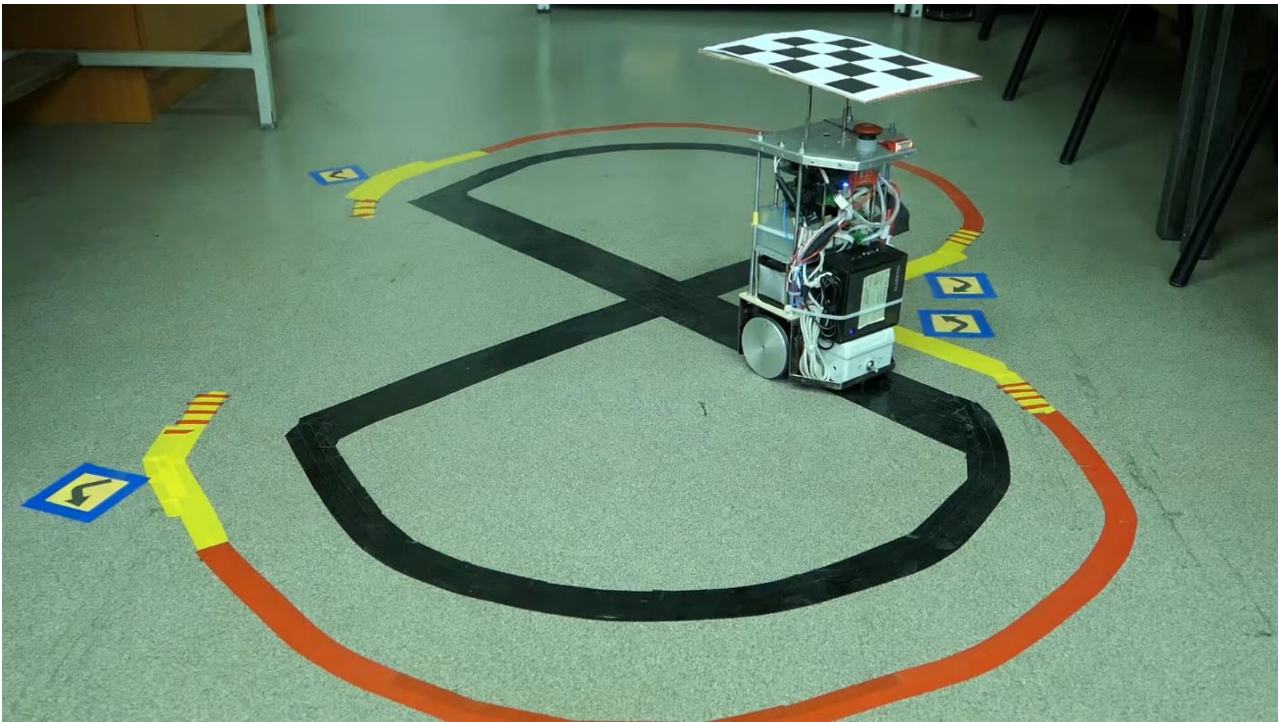
\* Ukupan broj trenarabilnih parametara se računa u odnosu na veličinu ulazne slike, odnosno broj piksela. Veličina ulazne slike za ove modele je  $120 \times 240 \times 3$  piksela.

Može se primetiti da se kapacitet samih modela razlikuje od vrednosti dobijenih za treniranje modela bazom podataka kreiranom u uslovima simulatora. Razlika potiče iz veličine ulaznog signala. U simuliranim uslovima veličina ulaza u neuralnu mrežu, slika nakon odsecanja, iznosi  $65 \times 320 \times 3$ , a u uslovima realnog sistema za autonomnu vožnju ulazni podatak, što predstavlja sliku nakon smanjenja veličine, iznosi  $120 \times 240 \times 3$  piksela. Razlika u veličini ulaznog podatka direktno utiče na broj čvorova neuralne mreže i njihovih međusobnih konekcija u svakom sloju te mreže. Pri tome, ulazni podatak s većim brojem piksela će dovesti do više čvorova u skrivenim slojevima neuralne mreže za jednako definisane parametre svih unutrašnjih slojeva, konvolucije, udruživanja, ravnajućih ili potpuno povezanih slojeva.

### 8.7.2. Kvalitativni rezultati autonomne vožnje

Autonomno održanje vozila u kolovoznoj traci korišćenjem J-Net modela, kao i referentnih modela AlexNet i PilotNet je verifikovano u sistemu za autonomnu vožnju realizovanom u laboratorijskim uslovima. Može se utvrditi da J-Net uspešno obavlja autonomnu vožnju na definisanoj trajektoriji, odnosno robo-vozilo korišćenjem J-Net modela se održava u kolovoznoj traci bez skretanja sa definisane trajektorije puta. Kao što je objašnjeno u sekciji 8.3, trajektorija se sastoji od različitih delova puta, ravan put, blage krivine u obliku polukruga i četiri oštre krivine, po dve u levo i u desno. Opservacijom autonomne vožnje se može konstatovati da autonomna vožnja korišćenjem J-Net modela, osim što vrši uspešno održanje vozila u kolovoznoj traci, dobro savladava krivine. Video snimak autonomne vožnje korišćenjem J-Net modela je dat u [112]. Na slici 8.18, prikazan je jedan frejm iz snimka autonomne vožnje po zadatoj putanji korišćenjem J-Net modela.

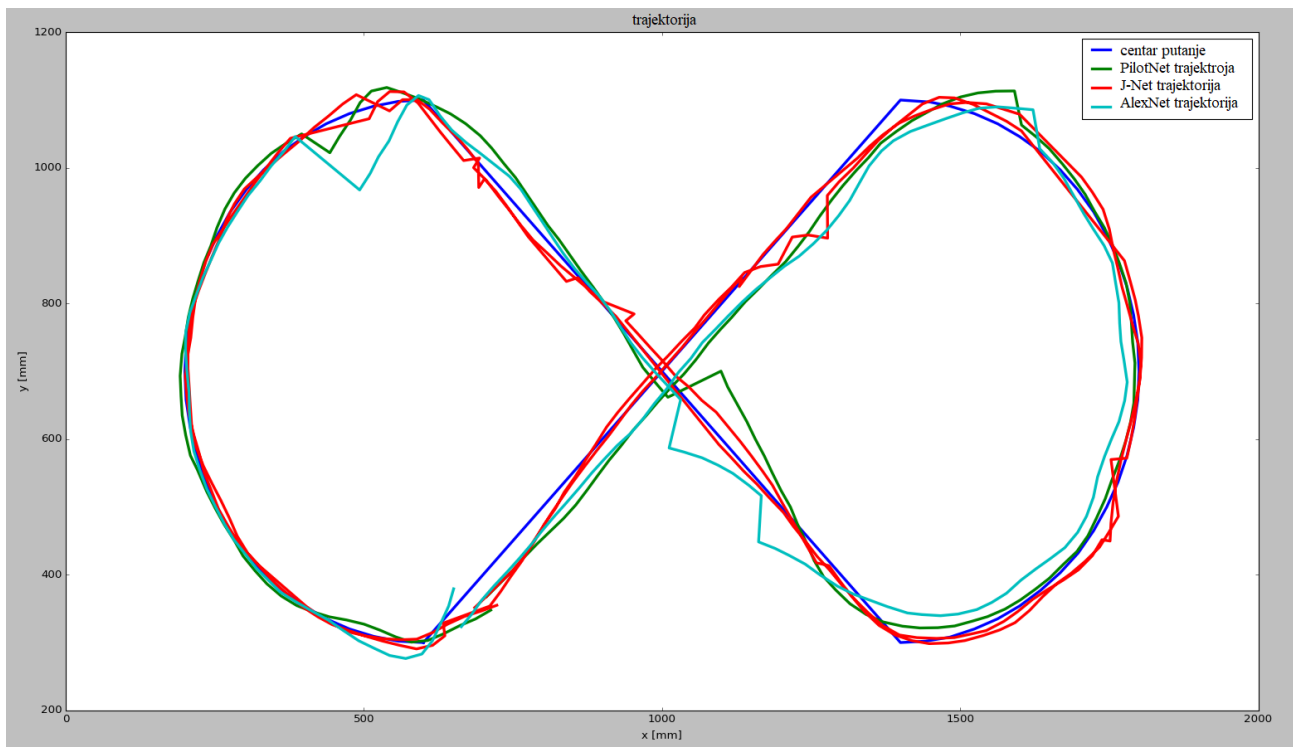




*Slika 8.18 Jedan frejm video snimka autonomne vožnje po zadatoj putanji korišćenjem J-Net neuralne mreže [112].*

U svrhu kvalitativne evaluacije autonomne vožnje, izvršena je analiza odstupanja kretanja robo-vozila korišćenjem J-Net, PilotNet i AlexNet modela od zadate trajektorije. Odstupanja su merena tako što je tokom režima autonomne vožnje praćena tačna lokalizacija vozila sistemom za lokalizaciju, koji radi nezavisno od algoritma za autonomnu vožnju, a kojim se beleže pozicije robota tokom autonomne vožnje.

Trajektorije autonomne vožnje robo-vozila korišćenjem implementiranih modela date su na zajedničkom dijagramu prikazanom na slici 8.19. Plavom linijom je prikazana definisana trajektorija kretanja, odnosno centar puta, koja odgovara tačnoj informaciji. Kretanje vozila korišćenjem J-Net modela je prikazano crvenom linijom. Uočava se da osim par sporadičnih odstupanja na pravom delu putanje i na polukružnici, a koji spadaju u dozvoljena odstupanja, jer vozilo nije skrenulo s puta već se ispravilo i dalje nastavilo autonomno kretanje po zadatoj trajektoriji, autonomna vožnja korišćenjem J-Net modela daje veoma zadovoljavajuće rezultate. Snimljena trajektorija vožnje korišćenjem J-Net modela u najvećem delu puta prati zadatu trajektoriju. Trajektorije autonomne vožnje korišćenjem PilotNet modela i AlexNet modela su prikazane zelenom i tirkiznom bojom, respektivno. Autonomna vožnja korišćenjem PilotNet i AlexNet modela takođe prati definisanu trajektoriju, međutim, mogu se uočiti blaga odstupanja u oštroj krivini u levo i na ravnom delu puta u blizini centralne raskrsnice, dok na ostatku puta robo-vozilo uglavnom dobro prati definisanu trajektoriju.



*Slika 8.19 Trajektorija kretanja robo-vozila u realnom vremenu korišćenjem dubokih neuralnih mreža: plava – centar zadate trajektorije; zelena – PilotNet; crvena – J-Net; svetlo plava – AlexNet.*

Poput kvalitativne analize autonomnog održanja vozila u kolovoznoj traci koje je vršena za autonomnu vožnju u simuliranim uslovima, ista analiza je primenjena i za uslove autonomne vožnje u realnim uslovima. Metode merenja odstupanja vozila od sredine kolovozne trake su srednja kvadratna greška i integralna apsolutna greška, opisane u sekciji 5.5. Kako su poznate tačne koordinate projektovane trajektorije puta, koje smo koristili u postupku automatizovanog kretanja robo-vozila, opisano u sekciji 8.4, a takođe su zabeležene tačne koordinate vozila tokom same autonomne vožnje, opisano u sekciji 8.6, odstupanje položaja vozila tokom autonomne vožnje od sredine kolovozne trake se dobija merenjem rastojanja od trenutne koordinate vozila do najbliže tačke projektovane trajektorije.

**Tabela 8.3** Kvalitativna evaluacija autonomne vožnje korišćenjem modela AlexNet, PilotNet i J-Net.

<b>Autonomna vožnja</b>	<b>AlexNet</b>	<b>PilotNet</b>	<b>J-Net</b>
Autonomno održanje vozila u kolovoznoj traci	uspešno	uspešno	uspešno
MSE [ $mm^2$ ]	945,5	964,6	820,7
IAE [ $mm$ ]	23,8	24,6	23,4

U tabeli 8.3, data je kvalitativna evaluacija autonomne vožnje, uspešnost autonomne vožnje i srednja kvadratna greška (MSE) i integralna apsolutna greška (IAE) za sva tri modela, J-Net, PilotNet i AlexNet. Tokom opservacije kretanja robo-vozila, autonomno održanje vozila u kolovoznoj traci korišćenjem J-Net modela je dalo dobre rezultate, pri čemu se čak i u ostrim krivinama vozilo kreće po putanji. Kod druga dva modela, PilotNet i AlexNet, autonomna vožnja je takođe uspešna, s tim



što se dešava da vozilo u krivini napravi veće odstupanje od sredine kolovozne trake, da bi se u narednom koraku vratilo na dobar pravac. Rezultati merenja odstupanja putanje robo-vozila korišćenjem J-Net, PilotNet i AlexNet, i izračunavanje srednje kvadratne greške i integralne apsolutne greške su u skladu sa prethodnim zapažanjima. Autonomna vožnja korišćenjem J-Net modela ima  $820,7 \text{ mm}^2$  srednju kvadratnu grešku i  $23,4 \text{ mm}$  integralnu apsolutnu grešku, AlexNet ima  $945,5 \text{ mm}^2$  srednju kvadratnu grešku i  $23,8 \text{ mm}$  integralnu apsolutnu grešku, dok su ove vrednosti za PilotNet  $964,6 \text{ mm}^2$  i  $24,6 \text{ mm}$  za MSE i IAE respektivno.

Objašnjenje za situaciju da se korišćenjem J-Net modela postigao najbolji kvalitet autonomne vožnje u smislu posmatranja odstupanja putanje robo-vozila od sredine projektovane kolovozne trake, a ne korišćenjem AlexNet i PilotNet modela kao što je to bio slučaj u simuliranim uslovima, je u tome što je prilikom treniranja modela dubokih neuralnih mreža korišćen relativnom mali skup podataka, svega 897 uzoraka. Iz ovog razloga, model sa manjim brojem parametara, u ovom slučaju J-Net, se bolje prilagodio nego model koji ima značajno veći broj parametara poput AlexNet-a kod koga može doći do nedovoljnog prilagođenja modela usled korišćenja relativnom male baze podataka.

### 8.7.3. Kvantitativni rezultati autonomne vožnje

Tokom autonomne vožnje robo-vozila po definisanoj trajektoriji, beležene su kvantitativne karakteristike vožnje, vreme izvršavanja zaključivanja o uglu kretanja vozila (predikcija) na osnovu slike sa vozila, čime je moguće odrediti kašnjenje u realnom vremenu, ali i dobiti informaciju o broju predikcija/zaključivanja koji se dobijaju kontinualno na osnovu jedne slike.

Kvantitativni rezultati su zabeleženi za autonomnu vožnju korišćenjem tri modela J-Net, PilotNet i AlexNet, tabela 8.4. Kao što se moglo očekivati na osnovu analize kompleksnosti implementiranih modela neuralnih mreža koja je prikazana u sekciji 8.7.1, autonomna vožnja korišćenjem J-Net modela u realnom vremenu daje najmanje kašnjenje. Prosečno vreme između dve predikcije iznosi  $22,67 \text{ ms}$ . PilotNet model ima kašnjenje od  $23,4 \text{ ms}$ , dok je AlexNet modelu potrebno najviše vremena da izvrši predikciju,  $30,2 \text{ ms}$ . Ovi rezultati su u skladu sa kompleksnošću modela dubokih neuralnih mreža i proporcionalni su ukupnom broju matematičkih operacija koje je potrebno izvršiti od trenutka kada se slika kao ulazni podatak pojavi na ulazu modela duboke neuralne mreže, dok model mreže ne izvrši predikciju, odnosno donese zaključak o narednom uglu kretanja vozila na osnovu slike.

**Tabela 8.4** Rezultati kvantitativne evaluacije autonomne vožnje korišćenjem modela AlexNet, PilotNet i J-Net u sistemu autonomne vožnje u realnim uslovima.

Autonomna vožnja	J-Net	PilotNet	AlexNet
Kašnjenje	22,7 ms	23,4 ms	30,2 ms
Broj predikcija po jednoj slici	16	11	8

Broj predikcija po jednom ulaznom signalu bi u idealnom slučaju iznosio jedan izlazni podatak po slici. Međutim, u projektovanom sistemu autonomne vožnje, kretanje robo-vozila je donekle usporeno, što je uslovljeno kompleksnošću samog sistema za autonomnu vožnju i korišćenjem više softverskih blokova koji upravljaju ovim sistemom. Iz ovog razloga, akvizicija slike sa robota se, iako se izvršava kontinuirano, izvršava u određenim vremenskim intervalima.

Algoritam za mašinsko učenje je projektovan da kontinuirano vrši predikcije na osnovu ulaznog podatka. Iz ovog razloga, za jednu sliku koja je prikupljena u datom trenutku se vrši i po nekoliko identičnih predikcija, sve dok sistem ne akvizira narednu sliku. Poboljšanje sistema za autonomnu vožnju je moguće vršiti u pravcu ostvarivanja bolje sinhronizacije između različitih komponenti

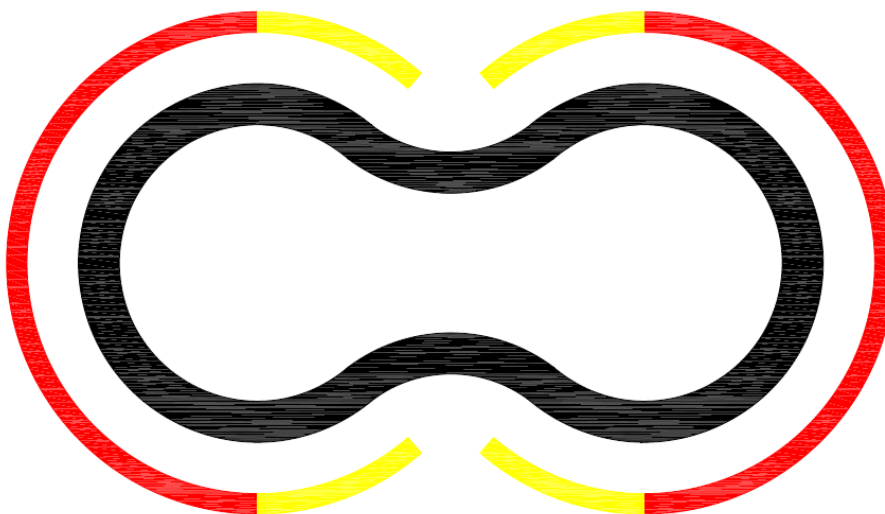
sistema i omogućiti da se slika sa robo-vozila prikuplja bez zadržske. Dodatno, podatak o broju predikcija po jednoj ulaznoj slici govori i o brzini samih modela mašinskog učenja. Akvizicija slike se obavlja u istim ili sličnim vremenskim intervalima za sva tri modela, dok je broj predikcija direktno proporcionalan brzini samih modela mašinskog učenja. Može se zaključiti, što je i očekivano s obzirom na izračunatu kompleksnost modela, da najbolje rezultate u smislu brzine odgovora modela mašinskog učenja ima J-Net model, 16 predikcija, u odnosu na druga dva implementirana modela, PilotNet i Aleks net, koji ostvaruju prosečno po 11 i 8 predikcija, respektivno, u datoj postavci sistema za autonomnu vožnju.

Zaključak je da je ostvarena uspešna autonomna vožnja korišćenjem kako projektovanog rešenja za J-Net duboku neuralnu mrežu, takođe i korišćenjem kontrolnih modela poznatih arhitektura PilotNet i AlexNet. Takođe, utvrđeno je da autonomna vožnja korišćenjem J-Net modela ostvaruje najmanje kašnjenje u realnom vremenu, istreniran model ima najmanje zauzeće memorijskog prostora u odnosu na druga poznata postojeća rešenja, pa se može zaključiti da je J-Net model uspešno odgovorio na zahteve postavljene za projektovanje računarski manje zahtevnog modela neuralne mreže koji će biti u stanju da obavlja zadatak autonomne vožnje.

## 8.8. Evaluacija modela J-Net na novoj kolovoznoj traci

U prethodnoj sekciji dati su rezultati verifikacije autonomne vožnje u realnim uslovima korišćenjem J-Net modela, pri čemu je korišćen sistem za autonomnu vožnju koji je opisan u sekciji 8.1. Međutim, potrebno je verifikovati da je J-Net model dovoljno generalizovan i da je moguće ostvariti autonomnu vožnju po novoj putanji, koja nikada nije viđena u postupku treniranja modela. Trajektorija za autonomnu vožnju, opisana u sekciji 8.3 korišćena je u postupku treniranja modela neuralnih mreža. Sada se vrši verifikaciji autonomne vožnje na novoj putanji.

Za evaluaciju modela J-Net na novoj kolovoznoj traci izabrana je putanja prikazana na slici 8.20. Razlika u odnosu na trajektoriju vožnje korišćene u prethodnom postupku, prikazane na slici 8.4, je to što sada nema ravnih delova puta, postoje tri veoma bliske, ali blaže krivine u gornjem i iste takve tri krivine u donjem delu putanje, kada se putanja posmatra kao na slici 8.20. Sličnost sa prethodnom trajektorijom su dva veća dela puta u obliku polukružnice. Zadržani su spoljašnji markeri trajektorije, odnosno „bankine“, crvene polukružnice koje označavaju krivinu na putu, i žute oznake koje upućuju na to da predstoji deo puta na kojem će biti potrebno izvršiti oštiri zaokret vozila.



Slika 8.20 Nova putanja za autonomnu vožnju u realnim uslovima

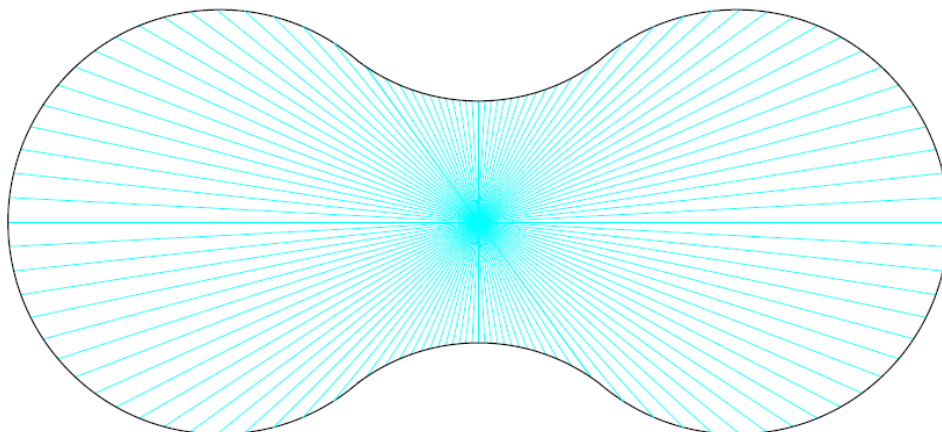
U prethodnoj postavci sistema, kolovozna traka se nalazi na podu laboratorije u kojoj je projektovan sistem za autonomnu vožnju. Kako bi se sačuvala ovakva postavka sistema i za buduće eksperimente, nova putanja je štampana na papiru velikog formata, koji je postavljen na pod laboratorije. Dodatna izmena u odnosu na originalnu putanju je to što je pozadina po kojoj je iscrtana crna kolovozna traka bele boje, za razliku od prvog slučaja u kojem je pozadina bila jednaka podu laboratorije, sive boje.

Na slici 8.21 prikazana je postavka sistema sa novom putanjom za autonomnu vožnju u realnim uslovima. Uz samu kolovoznu traku i horizontalnu saobraćajnu signalizaciju, dodati su i modeli automobila. Međutim, kako je kamera robo-vozila usmerena na put, objekti sa strane ne ometaju previše ponašanje vozila.



*Slika 8.21 Postavka sistema sa novom putanjom za autonomnu vožnju u realnim uslovima, [113].*

Autonomna vožnja je ostvarena korišćenjem J-Net modela istreniranog u postupku opisanom u sekciji 8.5, pri čemu su kompleksnost i karakteristike istreniranog J-Net modela date u sekciji 8.7. U slučaju evaluacije modela po novoj kolovoznoj traci, jedina izmena je nova putanja po kojoj se kreće ranije opisani J-Net model. Upravo generalizacija modela i postizanje autonomnog održanja vozila u kolovoznoj traci u novim uslovima je i predmet ove evaluacije.



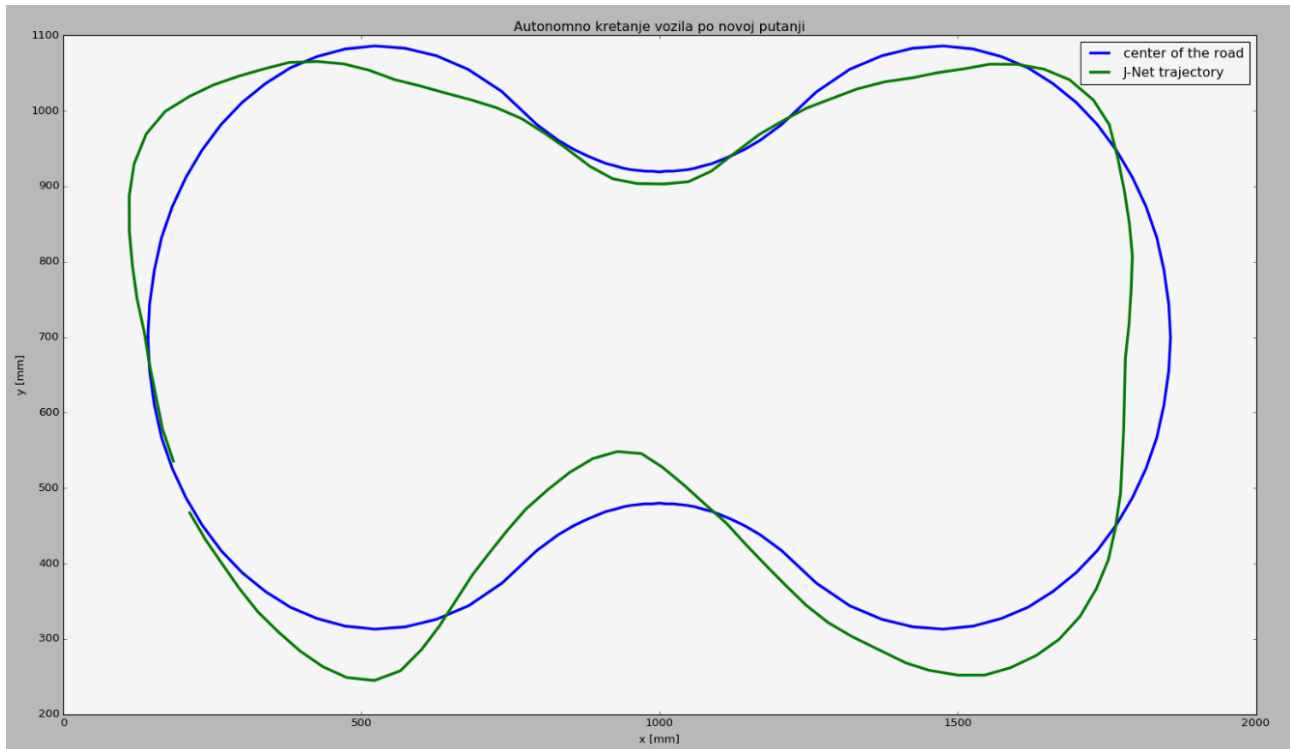
Slika 8.22 Izbor koordinata putanje za eksperimentalnu verifikaciju autonomne vožnje.

Postignuta je uspešna autonomna vožnja po novoj putanji korišćenjem J-Net modela. Zabeležene su pozicije robo-vozila prilikom svakog pomeraja korišćenjem sistema za lokalizaciju. Izračunato je odstupanje pozicije robo-vozila tokom autonomne vožnje u odnosu na sredinu novoprojektovane kolovozne trake. Na slici 8.22 nalazi se prikaz izbora koordinata nove kolovozne trake. Računato je odstupanje pozicije robo-vozila od najbliže tačke na novoj putanji, i na osnovu ovog odstupanja izračunate su srednja kvadratna greška (MSE) i integralna apsolutna greška (IAE). Rezultati su dati u tabeli 8.5. Na osnovu merenja greške odstupanja kretanja vozila od sredine kolovozne trake, zaključuje se da je kretanje vozila po novoj kolovoznoj traci uspešno, ali da su odstupanja od sredine kolovozne trake nešto veća u odnosu na prvi eksperiment. Greške odstupanja J-Net modela tokom kretanja po prvoj kolovoznoj traci, koja je ujedno korišćena i za treniranje neuralne mreže, su 820,7 mm<sup>2</sup> i 23,4 mm za MSE i IAE respektivno, dok su ove vrednosti 1891,5 mm<sup>2</sup> i 37,6 mm za MSE i IAE respektivno pri kretanju J-Net modela tokom kretanja po novoj kolovoznoj traci. Rezultati upućuju na to da model dobro generalizuje, a da su odstupanja od sredine kolovozne trake nešto veća nego u prvom slučaju, što je u skladu sa očekivanjima.

Tabela 8.5 Rezultati kvantitativne evaluacije autonomne vožnje na novom modelu putanje

Autonomna vožnja J-Net model	Originalna kolovozna traka	Nova kolovozna traka
Autonomno održanje vozila u kolovoznoj traci	uspešno	uspešno
MSE [mm <sup>2</sup> ]	820,7	1891,5
IAE [mm]	23,4	37,6

Trajektorija kretanja robo-vozila u realnom vremenu korišćenjem J-Net modela na novoj kolovoznoj traci je prikazana na slici 8.23, a video snimak autonomne vožnje kojom se vrši autonomno održanje vozila na novoj kolovoznoj tački je dat u [113]. Zaključuje se da J-Net model nove duboke neuralne mreže od-kraja-do-kraja projektovane za autonomnu vožnju uspešno generalizuje i korišćenjem ovog modela je moguće izvršiti autonomno održanje vozila u kolovoznoj traci i na potpuno novim putevima.



*Slika 8.23 Trajektorija kretanja robo-vozila u realnom vremenu korišćenjem J-Net modela na novoj kolovoznoj traci.*

Korišćenjem modela duboke konvolucione neuralne mreže, J-Net, projektovane za autonomnu vožnju moguće je ostvariti autonomno održanje vozila u kolovoznoj traci i u simuliranim i u realnim uslovima. Verifikacija sistema u realnim uslovima je izvršena na dva različita modela kolovoznih traka, tako da je samo na jednoj od njih vršena akvizicija uzoraka za treniranje modela neuralnih mreža, a druga putanja je poslužila za verifikaciju dobre generalizacije novoprojektovanog modela.

Na oba modela kolovoznih traka, ostvareno je uspešno održanje vozila u kolovoznoj traci korišćenjem modela nove duboke neuralne mreže. Arhitektura J-Net modela je projektovana za rad na ugradnim hardverskim platformama, te stoga ima manju kapacitivnost, broj parametara, matematičkih operacija, trenirani model zauzima manje memorijskog prostora, a tokom faze zaključivanja u realnom vremenu postiže ste manje kašnjenje u odnosu na druge poznate modele koji se mogu koristiti za autonomno održanje vozila u kolovoznoj traci.



## 9. Zaključak

Tehnike mašinskog učenja, i približnije dubokog učenja, postale su nerazdvojan činilac razvoja autonomnih vozila. Tehnike dubokog učenja se koriste u segmentima autonomne vožnje kao što su prepoznavanje i praćenje objekata, lokalizacija i mapiranje, semantička segmentacija i kloniranje ponašanja vozača učenjem od-kraja-do-kraja. Centralni problem koji se rešava u ovom radu je kloniranje ponašanja vozača učenjem od-kraja-do-kraja kroz zadatak autonomnog održanja vozila u kolovoznoj traci analizom informacija sa vizuelnih senzora. Projektovana je duboka neuralna mreža kojoj su ulaz slike sa kamere postavljene na vozilu, a izlaz je informacija o uglu okretanja upravljača. Ograničenje koje je uvedeno tokom definisanja problema je postavljanje brzine vozila na konstantnu vrednost, tako da je jedino ugao okretanja upravljača informacija kojom se u potpunosti upravlja vozilom.

U radu je primenjena originalna metodologija hijerarhijskog projektovanja duboke neuralne mreže kroz procese selekcije osnovnih elemenata arhitekture i specifičnih parametara implementacije u cilju zadovoljavanja unapred zadatih kriterijuma i performansi. Najpre su analizirane globalne karakteristike arhitekture neuralne mreže koje dominantno čine broj i tipovi slojeva, što ujedno predstavlja i dubinu neuralne mreže. Nakon toga se na sledećem hijerarhijskom nivou analiziraju uticaji veličine samih slojeva, konkretno broj mapa odlika svakog konvolucionog sloja. Na kraju se u najnižem hijerarhijskom nivou razmatra i uticaj veličine konvolucionih kernela.

Korišćenjem predložene metodologije, razvijena je nova originalna arhitektura duboke neuralne mreže za održanje vozila u kolovoznoj traci korišćenjem informacija sa vizuelnih senzora koja ima manju računarsku kompleksnost, broj trenabilnih parametara i potrebnih matematičkih operacija u jednoj iteraciji, u odnosu na postojeće arhitekture dubokih neuralnih mreža, dok performanse autonomne vožnje korišćenjem ovog modela ne degradiraju previše. Na kompleksnost duboke neuralne mreže direktno utiče sama arhitektura neuralne mreže, broj i vrsta slojeva i operacija koje se izvršavaju nad čvorovima ove neuralne mreže, kao i veličina ulaznog podatka u neuralnu mrežu. Model projektovane i implementirane nove arhitekture duboke neuralne mreže nazvan je J-Net. Osnovni cilj je bio da se istreniran model ovakve neuralne mreže primeni na hardverskoj platformi autonomnog robo-vozila, koja je obično namenski računarski sistem sa ograničenim resursima u pogledu računarske snage i memorijskog kapaciteta.

Verifikovana je autonomna vožnja korišćenjem nove duboke neuralne mreže za učenje od-kraja-do-kraja u dva scenarija, u simuliranim uslovima i u realnim uslovima. Upoređeni su rezultati verifikacije autonomne vožnje korišćenjem tri arhitekture dubokih neuralnih mreža: J-Net, PilotNet i AlexNet. Za verifikaciju autonomne vožnje u simuliranim uslovima korišćen je simulator autonomne vožnje otvorenog koda, dok je za verifikaciju vožnje u realnim uslovima projektovan i razvijen originalni sistem za autonomnu vožnju na Elektrotehničkom fakultetu Univerziteta u Beogradu u laboratoriji za elektroniku. U oba scenarija, autonomna vožnja korišćenjem J-Net modela je zadovoljila polazne hipoteze. Ostvarena je autonomna vožnja korišćenjem J-Net modela i u simulatoru i u realnim uslovima. Kvalitativne osobine vožnje korišćenjem J-Net modela su upoređene sa karakteristikama autonomne vožnje korišćenjem poznatih modela PilotNet i AlexNet. Karakteristike vožnje kao što su praćenje centralne linije putanje, savladavanje blagih i oštih krivina, vožnja po različitim teksturama puta su upoređive po kvalitetu sa druga dva rešenja. Sa druge strane, u kvantitativnom smislu, korišćenje J-Net modela je dalo bolje rezultate u poređenju sa drugim modelima. Postignuto je manje kašnjenje u realnom vremenu i u simuliranim uslovima, 23,8 ms je kašnjenje za J-Net, dok 24,1 ms i 26,0 ms imaju PilotNet i AlexNet, respektivno, i u realnim uslovima kašnjenje za J-Net je 22,7 ms, dok je za PilotNet i AlexNet 23,4 ms i 30,2 ms, respektivno. Takođe, poređenje kompleksnosti samih mrežnih arhitektura upućuje na manju računarsku zahtevnost J-Net modela u poređenju sa PilotNet i AlexNet modelima.

Za izvršavanje algoritma mašinskog učenja sa J-Net modelom u simuliranim uslovima potrebno je 150,84 hiljada matematičkih operacija, a sam istreniran model zauzima 1,8 MB memorijskog prostora. Izvršavanje algoritma mašinskog učenja sa PilotNet modelom zahteva 347,82 hiljada operacija i sam model zauzima 4,2 MB memorijskog kapaciteta, a korišćenjem AlexNet modela koristi se 42,45 miliona operacija i model zauzima 509,5 MB. Za autonomnu vožnju u realnim uslovima, J-Net model koristi 248,37 hiljada operacija i model zauzima 3 MB, dok su ove vrednosti korišćenjem PilotNet i AlexNet modela 1,31 milion operacija i 15,8 MB i 200,78 miliona operacija i 2,4 GB, respektivno.

Zaključuje se da je korišćenjem J-Net modela moguće postići autonomno održanje vozila u kolovoznoj traci, pri čemu je ovo rešenje manje zahtevno sa stanovišta računarske snage i zahteva za memorijskim resursima u odnosu na druga poznata rešenja. Dodatno, ostvareno je manje kašnjenje u uslovima autonomne vožnje u realnom vremenu, i u simuliranim i u realnim uslovima. Stoga zaključujemo da je J-Net model pogodan za implementaciju na namenskim računarskim sistemima sa ograničenim hardverskim resursima, poput robo-vozila. Praktična primena ovakvog rešenja je raznovrsna, od autonomnih dostavnih mobilnih vozila koja su manjeg gabarita od regularnih vozila koja su pogodna za široku upotrebu.

Nastavak istraživanja u oblasti kojom se bavi ova disertacija odvijace se u smeru razvoja novih algoritama za autonomnu vožnju, uključujući algoritme mašinskog učenja za učenje podsticanjem. Novo istraživanje ce obuhvatiti i uvođenje brzine kao drugog izlaza iz duboke neurlane mreže, tako da se slikom sa kamere montirane na vozilu upravlja i uglom okretanja upravljača i brzinom vozila. Dalje, postojeće rešenje za autonomnu vožnju moguće je nadograditi podsistemom za detekciju objekata na putu koja bi rezultirala zaustavljanjem vozila. Konačno, moguće je unaprediti projektovani sistem za autonomnu vožnju uvođenjem više različitih scenarija, uključujući dodavanje raskrsnica, semafora, različitih objekata i drugih učesnika u saobraćaju.



# Literatura

- [1] Wallace, R., Stentz, A., Thorpe, C. E., Moravec, H. P., Whittaker, W., Kanade, T. „*First results in robot road-following*,” In Proc. of the 9th International Joint Conference on Artificial intelligence (IJCAI), Los Angeles, CA, USA, 18–23 August 1985, pp. 1089-1095.
- [2] Kanade, T., Thorpe, C., Whittaker, W. „*Autonomous land vehicle project at CMU*,” In Proc. of the 1986 ACM fourteenth annual conference on Computer science. ACM, Cincinnati, Ohio, USA, 1986. pp. 71-80.
- [3] Dickmanns, E.D., Zapp, A. „*Autonomous High Speed Road Vehicle Guidance by Computer Vision*,” IFAC Proceedings Volumes, 1987, 20.5: 221-226.
- [4] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Homann, G., et al. „*Stanley: The Robot That Won the DARPA Grand Challenge*,” Journal of field Robotics, 2006, 23.9: 661-692.
- [5] Montemerlo, M., Thrun, S., Dahlkamp, H., Stavens, D., Strohband, S. „*Winning the DARPA grand challenge with an AI robot*,” In Proceedings of the 21st national conference on Artificial intelligence (AAAI), Boston, MA, USA, 16–20 July 2006; pp. 982–987.
- [6] Buehler, M., Iagnemma, K., Singh, S. „*The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*,” In Springer Tracts in Advanced Robotics, Vol. 59, Springer: Berlin/Heidelberg, Germany, 2009.
- [7] Orponen, P. „*Computational complexity of neural networks: A survey*,” Nordic Journal of Computing, 1994, 94–110.
- [8] Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., Sohl Dickstein, J. „*On the expressive power of deep neural networks*,” In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2847–2854.
- [9] Simonyan, K., Zisserman, A. „*Very deep convolutional networks for large-scale image recognition*,” arXiv 2004, arXiv:1409.1556.
- [10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. „*Going deeper with convolutions*,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
- [11] He, K., Zhang, X., Ren, S., Sun, J. „*Deep residual learning for image recognition*,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- [12] Sze, V., Chen, Y., Yang, T., Emer, J.S. „*Efficient Processing of Deep Neural Networks: A Tutorial and Survey*,” Proc. IEEE 2017, 105, 2295–2329.
- [13] Iandola, F., Keutzer, K. „*Keynote: small neural nets are beautiful: enabling embedded systems with small deep-neural-network architectures*,” In Proceedings of the 2017 International Conference on Hardware/Software Codesign and System Synthesis, Seoul, Korea, 15–20 October 2017; pp. 1–10.
- [14] SAE Standard J3016\_201806, „*Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*“, June 2018.

- [15] Ramkumar, M., „*Robotaxis: Voyage versus Ford, Waymo, and Tesla*“, Market Realist, Njujork, NY, SAD, Septembar 2019. Dostupno (na dan 14. septembar 2019): <https://articles2.marketrealist.com/2019/09/robotaxis-voyage-versus-ford-waymo-and-tesla/>
- [16] Walker, J., „*The Self-Driving Car Timeline – Predictions from the Top 11 Global Automakers*“, Emerj - Artificial Intelligence Research and Insight, Boston, MA, SAD, Septembar 2019. Dostupno (na dan 14. septembar 2019): <https://emerj.com/ai-adoption-timelines/self-driving-car-timeline-themselves-top-11-automakers/>
- [17] Boeriu, H., „*In 2021, BMW will pilot a fleet of 500 BMW iNEXT vehicles with Level 4 and 5 autonomy*“, BMWBLOG, Čikago, IL, SAD, Maj 2019. Dostupno (na dan 14. septembar 2019): <https://www.bmwblog.com/2019/05/07/in-2021-bmw-will-pilot-a-fleet-of-500-bmw-inext-vehicles-with-level-4-and-5-autonomy/>
- [18] Fridman, L., „Human-Centered Autonomous Vehicle Systems: Principles of Effective Shared Autonomy“, arXiv:1810.01835 [cs.AI], Oktobar 2018.
- [19] Horrell, P., „*The Vision iNext is BMW's first level 3 autonomous car*“, TopGear Magazine, BBC Broadcast Centre, London, Velika Britanija, Novembar 2018. Dostupno (na dan 14. septembar 2019): <https://www.topgear.com/car-news/la-motor-show-2018/vision-inext-bmws-first-level-3-autonomous-car>
- [20] Wasef, B., „*2019 Audi A8L Review | Brilliant engineering in an unassuming wrapper*“, Autoblog, Birmingham, MI, SAD, Oktobar 2018. Dostupno (na dan 14. septembar 2019): <https://www.autoblog.com/2018/10/16/2019-audi-a8-l-review-first-drive/>
- [21] Wade, M. R., „*Silicon Valley is winning the race to build the first driverless cars*“, International Institute for Management Development (IMD), Lozana, Švajcarska, Februar 2018. Dostupno (na dan 14. septembar 2019): <https://www.imd.org/research-knowledge/articles/silicon-valley-is-winning-the-race-to-build-the-first-driverless-cars/>
- [22] Cantrell, Robert, H. I. G. H. Donald, and Michael Atchley. "Autonomous Robot Delivery Systems and Methods." U.S. Patent Application No. 16/030,939. Vašington, DC, SAD, 2019.
- [23] Davey, Neil S., et al. "Automated delivery of good using autonomous robot." U.S. Patent No. 9,643,320, Myllymaki, J. (2016). *U.S. Patent No. 9,256,852*. Vašington, DC, SAD, Maj 2017.
- [24] Myllymaki, J. "Autonomous delivery platform." U.S. Patent No. 9,256,852. Vašington, DC, SAD, Februar 2016.
- [25] Kocić, J., Jovičić, N., Drndarević, V. "Sensors and Sensor Fusion in Autonomous Vehicles," 2018 26th Telecommunications Forum (TELFOR), Belgrade, 2018, pp. 420-425.
- [26] Nister, D., Naroditsky, O., and Bergen, J. "Visual odometry," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Washington, DC, USA, 2004, pp. I-I.
- [27] Sun, Z., Bebis, G., and Miller, R. "On-road vehicle detection: a review," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694-711, May 2006.
- [28] Yamazato, T. et al., "Image-sensor-based visible light communication for automotive applications," in IEEE Communications Magazine, vol. 52, no. 7, pp. 88-97, July 2014.
- [29] Dagan, E., Mano, O., Stein, G. P., and Shashua, A. "Forward collision warning with a single camera," IEEE Intelligent Vehicles Symposium, 2004, Parma, Italy, 2004, pp. 37-42.

- [30] Kurihata, H. et al., "Rainy weather recognition from in-vehicle camera images for driver assistance," IEEE Proceedings. Intelligent Vehicles Symposium, 2005., Las Vegas, 2005, pp. 205-210.
- [31] Fridman, L., „Human-Centered Autonomous Vehicle Systems: Principles of Effective Shared Autonomy“, Oct. 2018. Dostupno (na dan 3. oktobar 2019): <https://arxiv.org/abs/1810.01835>
- [32] Patole, S. M., Torlak, M., Wang D., and Ali, M. "Automotive radars: A review of signal processing techniques," in IEEE Signal Processing Magazine, vol. 34, no. 2, pp. 22-35, March 2017.
- [33] Steinbaeck, J., Steger, C., Holweg G., and Druml, N. "Next generation radar sensors in automotive sensor fusion systems," 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, 2017, pp. 1-6.
- [34] Schwarz, B. "LIDAR: Mapping the world in 3D," Nature Photonics 4.7, (2010): 429.
- [35] Davis, W. O., et al. "High performance MEMS scanner," U.S. Patent No. 7,482,730. 27 Jan. 2009.
- [36] „Consumer LiDAR“, Microvision, Inc., Redmond, WA, USA, 2019. Dostupno (na dan 29. septembar 2019): <http://www.microvision.com/consumer-lidar/>
- [37] Han, J., Kim, D., Lee M., and Sunwoo M., "Enhanced Road Boundary and Obstacle Detection Using a Downward-Looking LIDAR Sensor," in IEEE Transactions on Vehicular Technology, vol. 61, no. 3, pp. 971-985, March 2012.
- [38] Ogawa, T., Sakai, H., Suzuki, Y., Takagi K., and Morikawa K., "Pedestrian detection and tracking using in-vehicle lidar for automotive application," 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, 2011, pp. 734-739.
- [39] Thakur, R. "Scanning LIDAR in Advanced Driver Assistance Systems and Beyond: Building a road map for next-generation LIDAR technology," in IEEE Consumer Electronics Magazine, vol. 5, no. 3, pp. 48-54, July 2016.
- [40] Kutila, M., Pyykönen, P., Ritter, W., Sawade O., and Schäufele, B. "Automotive LIDAR sensor development scenarios for harsh weather conditions," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 265-270.
- [41] Zermas, D., Izzat I., and Papanikolopoulos, N. "Fast segmentation of 3D point clouds: A paradigm on Lidar data for autonomous vehicle applications," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 5067-5073.
- [42] Veličković, Z., Kocić, J., and Jovičić, N. „Pregled osnovnih tehnika kompjuterske vizije u autonomnim vozilima,“ Proc. of 17th International Symposium INFOTEH-JAHORINA, Jahorina, Bosna i Hercegovina, Mart 2018, pp. 1-6.
- [43] Shalev-Shwartz, S., Shammah, S., Shashua, A. „Safe, multi-agent, reinforcement learning for autonomous driving,“ arXiv 2016, arXiv:1610.03295.
- [44] Borrelli, F., Falcone, P., Keviczky, T., Asgari, J., and Hrovat, D. „MPC-based approach to active steering for autonomous vehicle systems,“ International Journal of Vehicle Autonomous Systems, 3(2), 2005, pp. 265-291.
- [45] Chollet, F. „Deep Learning with Python“, Manning Publications, Shelter Island, NY, USA, 2018; ISBN 9781617294433.

- [46] Goodfellow, I., Bengio, Y., Courville, A. „Deep Learning“, The MIT Press: Cambridge, MA, USA, 2017. Dostupno (na dan 3. novembar 2018): <https://www.deeplearningbook.org>
- [47] Aggarwal, C.C. An introduction to neural networks. In „Neural Networks and Deep Learning“ (pp. 1-52). Springer International Publishing: Cham, Switzerland, 2018; ISBN 978-3-319-94462-3.
- [48] Sutton, R.S., Barto, A.G. „Reinforcement Learning, 2nd ed.“, The MIT Press: Cambridge, MA, USA, 2018, p. 552, ISBN 9780262039246.
- [49] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D. „Backpropagation applied to handwritten zip code recognition“, *Neural Comput.* 1989, 1, 541–551.
- [50] Simard, D., Steinkraus, P.Y., Platt, J.C. „Best practices for convolutional neural networks applied to visual document analysis“, In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, Edinburgh, UK, 6 August 2003; pp. 958–963.
- [51] Shin, H., Roth, H., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R. „Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning“, *IEEE Trans. Med. Imaging* 2016, 35, 1285–1298.
- [52] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A. „Context Encoders: Feature Learning by Inpainting“, In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
- [53] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Li, F.-F. „Large-Scale Video Classification with Convolutional Neural Networks“, In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
- [54] Chi, J., Kim, H.-C. „Prediction of Arctic Sea Ice Concentration Using a Fully Data Driven Deep Neural Network“, *Remote Sens.* 2017, 9, 1305.
- [55] Tetko, I. V., Livingstone, D. J. and Luik, A. I. „Neural network studies. 1. Comparison of overfitting and overtraining“, *Journal of chemical information and computer sciences*, 1995, 35.5: 826-833.
- [56] Lawrence, S. and Giles, C. L. "Overfitting and neural networks: conjugate gradient and backpropagation," *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, Italy, 2000, pp. 114-119 vol.1.
- [57] Dietterich, T. „Overfitting and undercomputing in machine learning“, *ACM computing surveys (CSUR)*, 1995, 27.3: 326-327.
- [58] Caruana, R., Lawrence, S., Giles, C.L. „Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping.“ In: *Advances in neural information processing systems*. 2001. p. 402-408.
- [59] Prechelt, L. „Early stopping-but when?“, In: *Neural Networks: Tricks of the trade*. Springer, Berlin, Heidelberg, 1998. p. 55-69.
- [60] Yao, Y., Rosasco, L., and Caponnetto, A. „On early stopping in gradient descent learning“, *Constructive Approximation*, 2007, 26.2: 289-315.
- [61] van der Aalst, W. M. P., Rubin, V., Verbeek, H. M. W. , van Dongen, B. F., Kindler, E., and Günther, C. W. "Process mining: a two-step approach to balance between underfitting and overfitting." *Software & Systems Modeling* 9.1 (2010): 87.

- [62] Ding, J., Chen, B., Liu, H., and M. Huang, "Convolutional Neural Network With Data Augmentation for SAR Target Recognition," in *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 364-368, March 2016.
- [63] Krizhevsky, A., Sutskever, I., Hinton, G.E. „Imagenet classification with deep convolutional neural networks“ In *Advances in Neural Information Processing Systems*, 2012; pp. 1097–1105.
- [64] ImageNet Large Scale Visual Recognition Challenge, ILSVRC-2012. Dostupno (na dan 5. oktobar 2019): <http://image-net.org/challenges/LSVRC/2012/index>
- [65] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Berg, A.C. „Imagenet large scale visual recognition challenge“, *Int. J. Comput. Vis.* 2015, 115, 211–252.
- [66] Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A., Bengio, Y. „Renet: A recurrent neural network based alternative to convolutional networks“, *arXiv* 2015, arXiv:1505.00393.
- [67] Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V. „Learning Transferable Architectures for Scalable Image Recognition,“ In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.
- [68] Acuna, D., Ling, H., Kar, A., Fidler, S. „Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++,“ In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 859–868.
- [69] Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B. „Video-to-video synthesis,“ In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, QC, Canada, 3–8 December 2018; pp. 1152–1164.
- [70] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, J., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. „Mastering the game of go without human knowledge,“ *Nature* 2017, 550, 354.
- [71] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lillicrap, T. „Mastering chess and shogi by self-play with a general reinforcement learning algorithm,“ *arXiv* 2017, arXiv:1712.01815.
- [72] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Chen, J. „Deep speech 2: End-to-end speech recognition in English and Mandarin,“ In *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 19–24 June 2016; pp. 173–182.
- [73] Chen, Z., Zhang, T., Ouyang, C. „End-to-End Airplane Detection Using Transfer Learning in Remote Sensing Images,“ *Remote Sens.* 2018, 10, 139.
- [74] Yao, Y., Wang, H., Li, S., Liu, Z., Gui, G., Dan, Y., Hu, J. „End-To-End Convolutional Neural Network Model for Gear Fault Diagnosis Based on Sound Signals,“ *Appl. Sci.* 2018, 8, 1584.
- [75] Kocić, J., Jovičić, N., Drndarević, V. „Driver behavioral cloning using deep learning,“ In *Proceedings of the 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Republika Srpska, 21–23 March 2018; pp. 1–5.
- [76] Kocić, J., Jovičić, N., Drndarević, V. „An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms,“ *Sensors* 2019, 19, 2064.
- [77] Chen, C., Seff, A., Kornhauser, A., Xiao, J. „Deepdriving: Learning affordance for direct perception in autonomous driving,“ In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. pp. 2722-2730.

- [78] Browning, B., Melik-Barkhudarov, N., Bagnell, J. A. „Perception system for autonomous vehicle,“ U.S. Patent Application No 15/640,370, 2018.
- [79] Rosique, F., Navarro, P. J., Fernández, C., Padilla, A. „A systematic review of perception system and simulators for autonomous vehicles research,“ *Sensors*, 2019, 19(3), 648.
- [80] Van Brummelen, J., O’Brien, M., Gruyer, D., Najjaran, H. „Autonomous vehicle perception: The technology of today and tomorrow,“ *Transportation research part C: emerging technologies*, 89, 2018, pp. 384-406.
- [81] Xu, X., Fan, C. K. „Autonomous vehicles, risk perceptions and insurance demand: An individual survey in China,“ *Transportation Research Part A: Policy and Practice*, 124, 2019. pp. 549-556.
- [82] Xu, D., Anguelov, D., Jain, A. "PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 244-253.
- [83] Chavez-Garcia, R. O. , Aycard, O. "Multiple Sensor Fusion and Classification for Moving Object Detection and Tracking," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 525-534, Feb. 2016.
- [84] Durrant-Whyte, H., Bailey, T. "Simultaneous localization and mapping: part I," in *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, June 2006.
- [85] Wang, S., Clark, R. , Wen H., Trigoni, N. "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 2043-2050.
- [86] Oh, S., Kang, H. "Fast Occupancy Grid Filtering Using Grid Cell Clusters From LIDAR and Stereo Vision Sensor Data," in *IEEE Sensors Journal*, vol. 16, no. 19, pp. 7258-7266, Oct.1, 2016.
- [87] Shelhamer, E., Long J., Darrell, T. "Fully Convolutional Networks for Semantic Segmentation," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651, 1 April 2017.
- [88] Girshick, R., Donahue, J., Darrell, T., Malik, J. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 580-587.
- [89] Noh, H., Hong S., Han, B. "Learning Deconvolution Network for Semantic Segmentation," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1520-1528.
- [90] Chen, L., Papandreou, G., Kokkinos, I., Murphy K., Yuille, A. L. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 1 April 2018.
- [91] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., et al. „End to end learning for self-driving cars,“ arXiv 2016, arXiv:1604.07316.
- [92] Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U. „Explaining how a deep neural network trained with end-to-end learning steers a car,“ arXiv 2017, arXiv:1704.07911.
- [93] Mehta, A., Adithya, S., Anbumani, S. „Learning end-to-end autonomous driving using guided auxiliary supervision,“ arXiv 2018, arXiv:1808.10393.

- [94] Chen, Y., Wang, J., Li, J., Lu, C., Luo, Z., Xue, H., Wang, C. „LiDAR-Video Driving Dataset: Learning Driving Policies Effectively,“ In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5870–5878.
- [95] Ramezani Dooraki, A., Lee, D.-J. „An End-to-End Deep Reinforcement Learning-Based Intelligent Agent Capable of Autonomous Exploration in Unknown Environments,“ Sensors 2018, 18, 3575.
- [96] Bansal, M., Krizhevsky, A., Ogale, A. „Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst,“ arXiv preprint 2018, arXiv:1812.03079.
- [97] Udacity, Inc. Self-Driving Car Simulator. Available online: <https://github.com/udacity/self-driving-car-sim> (accessed on 5 November 2018).
- [98] Kingma, D.P., Ba, J. „Adam: A Method for Stochastic Optimization,“ In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
- [99] Glorot, X., Bengio, Y. „Understanding the difficulty of training deep feedforward neural networks,“ In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010.
- [100] Kocić, J. „Model Shallow,“ Dostupno (na dan 22 februar 2019): <https://www.youtube.com/watch?v=w4UUz-gI7yw>
- [101] Gulli, A., Pal, S. „Deep Learning with Keras,“ Packt Publishing: Birmingham, UK, 2017.
- [102] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. „Gradient-Based Learning Applied to Document Recognition,“ Dostupno (na dan 1.5.19): <http://yann.lecun.org/exdb/publis/pdf/lecun-01a.pdf>
- [103] Kocić, J. „Model AlexNet“. Dostupno (na dan 5 November 2018): [https://www.youtube.com/watch?v=ICxRQfm5a\\_4](https://www.youtube.com/watch?v=ICxRQfm5a_4)
- [104] Kocić, J. „Model PilotNet (NVIDIA Net)“. Dostupno (na dan 5 November 2018): <https://www.youtube.com/watch?v=YPZRKh4xbm4>
- [105] Kocić, J. „Model J-Net“. Dostupno (na dan 5 November 2018): <https://www.youtube.com/watch?v=BB-YljzloWI>
- [106] Kolodko, J., Vlacic, L. "Cooperative autonomous driving at the Intelligent Control Systems Laboratory," in IEEE Intelligent Systems, vol. 18, no. 4, pp. 8-11, July-Aug. 2003.
- [107] Deng, W., Lee, Y. H., Zhao, A. "Hardware-in-the-loop simulation for autonomous driving," 2008 34th Annual Conference of IEEE Industrial Electronics, Orlando, FL, 2008, pp. 1742-1747.
- [108] Okuyama, T., Gonsalves, T., Upadhyay, J. "Autonomous Driving System based on Deep Q Learnig," 2018 International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 2018, pp. 201-205.
- [109] Refaat, K. S., Ding, K., Ponomareva, N., Ross, S. „Agent Prioritization for Autonomous Navigation“, arXiv:1909.08792 [cs.RO].
- [110] Janković, N., Ćirić, S. Jovičić, N. "System for indoor localization of mobile robots by using machine vision," 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, 2015, pp. 619-622.



[111] Huili, Z. "Realization of Files Sharing between Linux and Windows Based on Samba," *2008 International Seminar on Future BioMedical Information Engineering*, Wuhan, Hubei, 2008, pp. 418-420.

[112] Kocić, J. „J-Net 2019 ETF“. Dostupno (na dan 26. jul 2019): <https://www.youtube.com/watch?v=JHD60uboy-o&list=UUHSiCyli0YrBL9ReViQUahQ&index=1>

[113] Kocić, J. „J-Net na novoj putanji – ETF“. Dostupno (na dan 21. 02. 2019): <https://www.youtube.com/watch?v=pVPA2RdRQSM&list=UUHSiCyli0YrBL9ReViQUahQ>

## Biografija autora

Jelena Kocić (devojačko Cvetković) je rođena 24.01.1982. godine u Kragujevcu. Završila je Prvu kragujevačku gimnaziju 2001. godine, specijalizovano matematičko odeljenje za talentovane učenike. Tokom gimnazijskog školovanja, bila je polaznik Istraživačke stanice Petnica na seminarima Primenjene fizike i elektronike i učesnik republičkih takmičenja iz informatike. Osnovne akademske studije je završila 2009. godine na Elektrotehničkom fakultetu u Beogradu, smer Elektronika. Doktorske akademske studije upisala je školske 2010/2011. godine na Elektrotehničkom fakultetu u Beogradu, smer Elektronika.

Od novembra 2009. godine radi u Institutu IRITEL u Beogradu u sektoru Energetske elektronike. Od 2010. do 2016. zaposlena je u Institutu Vlatacom u Beogradu kao inženjer razvoja elektronike u sektoru za razvoj elektro-optičkih sistema. Od 2016. do 2018. radi u kompaniji Comtrade Solutions Engineering u Beogradu kao inženjer razvoja softvera u domenu automobilske industrije. Od 2018. do 2019. radi u kompaniji Microvision, Inc., u Redmondu, Vašington, SAD, na poziciji nezavisnog saradnika za mašinsko učenje gde se bavi projektovanjem i razvojem novih algoritama dubokog učenja za kompjutersku viziju korišćenjem 3D lidar senzora. Od 2019. godine zaposlena je u kompaniji Bitgear Wireless Design Services u Beogradu kao stariji inženjer mašinskog učenja gde se bavi razvojem novih rešenja u oblasti kompjuterske vizije i mašinskog učenja.

U dosadašnjem istraživačkom radu je objavila: jedan rad u vrhunskom međunarodnom časopisu, sedam radova na međunarodnim konferencijama, osam radova na domaćim konferencijama, jedno tehničko rešenje i jedan rad je u pripremi za predaju u međunarodni časopis. Od novembra 2018. godine je recezent u časopisu IEEE Access za naučne radove iz domena dubokog učenja.

## Изјава о ауторству

Име и презиме аутора Јелена Коцић

Број индекса 5067/2010

### Изјављујем

да је докторска дисертација под насловом

“Аутономно одржање возила у коловозној траци анализом информација са визуелних сензора коришћењем неуралне мреже”

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

Потпис аутора

У Београду, 25. 10. 2019.

Јелена Коцић

## Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора    Јелена Коцић  
Број индекса                5067/2010  
Студијски програм        Електроника  
Наслов рада                Аутономно одржање возила у коловозној траци анализом  
информација са визуелних сензора коришћењем неуралне мреже  
Ментор                        др Ненад Јовичић, ванредни професор

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 25. 10. 2019.



## Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом: "Аутономно одржање возила у коловозној траци анализом информација са визуелних сензора коришћењем неуралне мреже" која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.  
Кратак опис лиценци је саставни део ове изјаве).

у Београду, 25. 10. 2019.

Потпис аутора



1. **Ауторство.** Дозвољава се умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално.** Дозвољава се умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада.** Дозвољава се умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима.** Дозвољава се умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада.** Дозвољава се умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима.** Дозвољава се умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.