

# Alat za označavanje linija vozne trake i usporedba modela zasnovanih na strojnom učenju za detekciju linija vozne trake

---

Križanac, Rea

Master's thesis / Diplomski rad

2024

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:200:120482>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-30**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURAJA STROSSMAYERA U OSIJEKU**

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studiji**

**ALAT ZA OZNAČAVANJE LINIJA VOZNE TRAKE I  
USPOREDBA MODELA ZASNOVANIH NA STROJNOM  
UČENJU ZA DETEKCIJU LINIJA VOZNE TRAKE**

**Diplomski rad**

**Rea Križanac**

**Osijek, 2024**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju****Ocjena diplomskog rada na sveučilišnom diplomskom studiju**

<b>Ime i prezime pristupnika:</b>	Rea Križanac
<b>Studij, smjer:</b>	Sveučilišni diplomski studij Elektrotehnika, Komunikacije i
<b>Mat. br. pristupnika, god.</b>	D-1484, 07.10.2022.
<b>JMBAG:</b>	0165077592
<b>Mentor:</b>	prof. dr. sc. Mario Vranješ
<b>Sumentor:</b>	doc. dr. sc. Denis Vranješ
<b>Sumentor iz tvrtke:</b>	Zvonimir Kaprocki
<b>Predsjednik Povjerenstva:</b>	prof. dr. sc. Marijan Herceg
<b>Član Povjerenstva 1:</b>	prof. dr. sc. Mario Vranješ
<b>Član Povjerenstva 2:</b>	izv. prof. dr. sc. Ratko Grbić
<b>Naslov diplomskog rada:</b>	Alat za označavanje linija vozne trake i usporedba modela zasnovanih na strojnom učenju za detekciju linija vozne trake
<b>Znanstvena grana diplomskog rada:</b>	<b>Obradba informacija (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	Detekcija linija vozne trake jedan je od osnovnih zadataka koje autonomno vozilo treba samostalno obavljati. Za potrebe obavljanja tog zadatka koristi se slika dobivena s prednje kamere u vozilu. Današnje metode detekcije linija vozne trake sve se više zasnivaju na primjeni modela strojnog učenja. Kako bi se modeli zasnovani na strojnom učenju obučili da obavljaju zadatak detekcije linija vozne trake, potrebno im je osigurati vrlo veliki skup podataka s označenim linijama vozne trake. U sklopu ovog diplomskog rada potrebno je načiniti novu vlastitu bazu podataka
<b>Datum ocjene pismenog dijela diplomskog rada od strane mentora:</b>	11.12.2024.
<b>Ocjena pismenog dijela diplomskog rada od strane mentora:</b>	Izvrstan (5)
<b>Datum obrane diplomskog rada:</b>	18.12.2024.
<b>Ocjena usmenog dijela diplomskog rada (obrane):</b>	Izvrstan (5)
<b>Ukupna ocjena diplomskog rada:</b>	Izvrstan (5)
<b>Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:</b>	18.12.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 18.12.2024.

<b>Ime i prezime Pristupnika:</b>	Rea Križanac
<b>Studij:</b>	Sveučilišni diplomski studij Elektrotehnika, Komunikacije i informatika
<b>Mat. br. Pristupnika, godina upisa:</b>	D-1484, 07.10.2022.
<b>Turnitin podudaranje [%]:</b>	10

Ovom izjavom izjavljujem da je rad pod nazivom: **Alat za označavanje linija vozne trake i usporedba modela zasnovanih na strojnom učenju za detekciju linija vozne trake**

izrađen pod vodstvom mentora prof. dr. sc. Mario Vranješ

i sumentora doc. dr. sc. Denis Vranješ

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

## Sadržaj:

1. UVOD .....	1
2. DETEKCIJA LINIJA VOZNIH TRAKA .....	3
2.1 Algoritmi za detekciju linija voznih traka zasnovani na tehnikama tradicionalnog računalnog vida.....	4
2.2 Algoritmi za detekciju linija voznih traka zasnovani na strojnom učenju .....	9
2.3 Skupovi podataka s označenim linijama voznih traka .....	16
2.3.1 TuSimple skup podataka .....	16
2.3.2 CULane skup podataka.....	19
2.3.3 LLAMAS skup podataka.....	20
2.3.4 KITTI skup podataka.....	21
2.3.5 BDD100K skup podataka.....	23
2.3.6 Osvrt na predstavljene podatkovne skupove .....	24
2.4 Postojeći alati za označavanje linija voznih traka .....	25
2.4.1 LabelMe alat za označavanje.....	25
2.4.2 VIA alat za označavanje .....	26
2.4.3 Labelbox alat za označavanje .....	26
2.4.4 CVAT alat za označavanje .....	27
2.4.5 Deepen AI alat za označavanje.....	28
2.4.6 Osvrt na predstavljene alate za označavanje linija vozne trake.....	29
3. IZRADA APLIKACIJE ZA OZNAČAVANJE LINIJA VOZNE TRAKE.....	31
3.1 Zahtjevi na aplikaciju za označavanje linija voznih traka.....	31
3.2 Tehnički detalji izrade aplikacije za ručno označavanje linija vozne trake .....	33
4. ANALIZA PERFORMANSI METODA ZA DETEKCIJU LINIJA VOZNE TRAKE ZASNOVANIH NA DUBOKOM UČENJU .....	43
4.1 Testiranje alata za označavanje linija vozne trake .....	44
4.2 Izrada vlastitog skupa podataka .....	47
4.3 Izgradnja i evaluacija detektora linija voznih traka .....	50
4.3.1 PyTorchAutoDrive .....	51
4.3.2 Docker i postavljanje okruženja .....	53
4.4 Evaluacija performansi modela za detekciju linija voznih traka zasnovanih na dubokom učenju .....	54
4.4.1 Analiza rezultata modela treniranih na CULane skupu podataka .....	55
4.4.2 Analiza rezultata modela treniranih na LLAMAS skupu podataka .....	63
4.4.3 Analiza rezultata modela treniranih na TuSimple skupu podataka .....	70
4.4.3 Treniranje i testiranje mreža na vlastitom skupu podataka .....	76

4.5	Završni osvrt na rezultate dobivene testiranjem i evaluacijom različitih metoda za detekciju linija vozne trake na različitim skupovima podataka.....	80
5.	ZAKLJUČAK .....	82
	LITERATURA.....	83
	SAŽETAK.....	86
	ABSTRACT .....	87
	ŽIVOTOPIS .....	88
	PRILOZI.....	89

## 1. UVOD

U posljednjih nekoliko godina broj autonomnih vozila na cesti sve više raste, dok se razina samostalnosti takvih vozila samo povećava. Automobilska industrija bilježi ubrzan razvoj, gdje je fokus stavljen na automatizaciju što većeg broja aktivnosti koje bi inače obavljao sam vozač. Razvoj autonomnih vozila i naprednih sustava za pomoć vozačima (*engl.* Advanced Driver-Assistance Systems - ADAS) predstavlja jedno od najvažnijih tehnoloških dostignuća u području automobilske industrije. ADAS koristi različite senzore za percepciju okoline. Prikupljeni podaci se obrađuju putem specifičnih hardverskih i softverskih komponenti, koje zatim prosljeđuju aktuatorima poruke o potrebnim akcijama. Time se vozaču olakšava upravljanje vozilom, a u nekim slučajevima vozač može biti u potpunosti zamijenjen. Glavni cilj ove automatizacije je smanjenje broja stradalih u prometnim nesrećama.

Jedna od ključnih komponenti ovih sustava je detekcija linija vozne trake, koja omogućava vozilima da se pravilno pozicioniraju u voznoj traci, održavaju pravac kretanja i izbjegavaju potencijalne opasnosti. Iako je detekcija linija vozne trake problem na kojem se radi duži niz godina, detekcija voznih traka je i dalje vrlo izazovno područje. Vozna traka ima vrlo jednostavan izgled, označena je s dvjema paralelnim linijama na cesti te ne pruža bilo kakve složene ili karakteristične značajke kojima se one mogu detektirati. Linije najčešće dolaze u dvjema bojama, bijeloj i žutoj, te su različitog izgleda: pune linije, isprekidane linije, dvostruke linije ili linije koje označavaju prijelaz između prometnih traka, što čini računalno modeliranje trake kompliciranim. Algoritmi koje se koriste za detekciju linija voznih traka se prema pristupu na kojem se zasnivaju mogu podijeliti u dvije grupe. Tradicionalni pristup detekciji linija vozne trake koristi klasične algoritme računalnog vida (*engl.* Computer Vision - CV), poput Hough-ove transformacije i algoritma Canny-eve detekcije rubova. Iako ovi algoritmi pružaju određeni nivo točnosti, često se suočavaju s izazovima u složenim uvjetima vožnje kao što su loši vremenski uvjeti, slaba vidljivost i složeni putevi. Moderni pristup nastaje pojavom strojnog učenja (*engl.* Machine Learning – ML), posebno dubokog učenja.

Tehnike strojnog učenja utječu na razna područja, uključujući računalni vid i obradu prirodnog jezika. Najčešće se koriste pristupi iz područja nadziranog učenja, koji zahtijevaju označene skupove podataka. U takvim skupovima, svaki podatak ima odgovarajuću oznaku (*engl.* annotation). Na primjeru detekcije linija vozne trake, svaka linija definirana je skupom određenog broja koordinata točaka koje pripadaju liniji. Modeli dubokog učenja zahtijevaju velike skupove podataka čije kreiranje se obično odrađuje ručno. Proces prikupljanja podataka za stvaranje skupa

podataka može biti skup, izazovan i dugotrajan proces te od osobe koja ju kreira zahtjeva razumijevanje konteksta kako bi točno označila skup. Kako bi se olakšao i ubrzao proces označavanja, razvijaju se razni alati koji se koriste za različite tipove podataka.

Ovaj diplomski rad se fokusira na razvoj alata za označavanje stvarnih linija vozne trake unutar prikupljenih okvira video snimki. Ovaj alat omogućava korisniku intuitivno označavanje linija vozne trake pomoću korisničkog sučelja koje podržava kreiranje i upravljanje projektima. Korisnik može kreirati novi projekt, učitati postojeći i dodavati nove fotografije ili snimke. Kroz sučelje, moguće je odabrati galeriju fotografija, ručno crtati oznake linija, brisati greške, te izvoziti datoteke s oznakama u dvama različitim formatima, uz kreiranje binarnih slika koje se mogu koristiti za daljnju analizu. Ovaj rad također uključuje izradu vlastitog skupa podataka snimki vožnje na različitim tipovima cesta, pri različitim vremenskim uvjetima i uvjetima osvjetljenja. U tu svrhu, načinjen je novi skup podataka koji sadrži snimke vožnje prikupljene u različitim uvjetima.

Drugi dio praktičnog rada usmjeren je na usporedbu performansi različitih modela zasnovanih na strojnom učenju za detekciju linija vozne trake, s posebnim fokusom na rezultate dobivene korištenjem Prostorne CNN (*engl.* Spatial Convolutional Neural Network- SCNN) mreže u kombinaciji s različitim osnovnim arhitekturama. Modeli su trenirani i testirani na postojećim skupovima podataka kako bi se dobili usporedivi rezultati, a zatim su dodatno testirani na novo-kreiranom skupu podataka s oznakama linija vozne trake. Analiza se fokusira na procjenu performansi algoritama u različitim scenarijima, s posebnim naglaskom na uvjete koji se javljaju u stvarnim situacijama na cesti. Svaki model ocjenjuje se prema metrikama kao što su odziv, preciznost i F1, koje su objašnjene u daljnjem tekstu. Ovaj pristup omogućuje procjenu korisnosti novo-kreiranog podatkovnog skupa te može izdvojiti njezine potencijalne prednosti u poboljšanju točnosti detekcije i prilagodbi modela za stvarne uvjete vožnje.

U drugom poglavlju obrađuju se metode za detekciju linija vozničkih traka, uključujući tradicionalne algoritme računalnog vida i modele zasnovane na strojnome učenju, uz pregled dostupnih skupova podataka i alata za označavanje. Treće poglavlje fokusira se na izradu aplikacije za ručno označavanje linija vozne trake, uključujući njezine funkcionalnosti i tehničke specifikacije. U četvrtom poglavlju provodi se analiza performansi metoda za detekciju linija vozne trake zasnovanih na dubokome učenju, uključujući testiranje aplikacije, kreiranje vlastitog skupa podataka i evaluaciju različitih modela. Na kraju, u petom poglavlju, iznosi se zaključak u kojem se sumiraju rezultati.

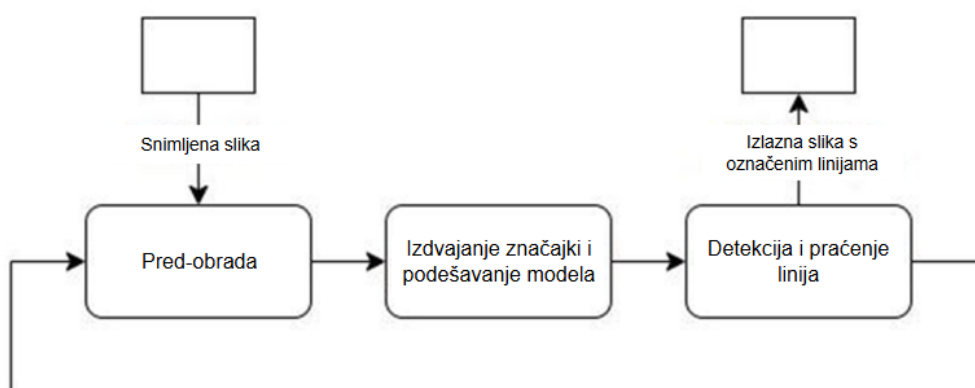


## 2. DETEKCIJA LINIJA VOZNIH TRAKA

Automobili su se od svog prvog pojavljivanja krajem 19. stoljeća značajno razvili. Prvi automobili, poput onih koje su proizvodili Karl Benz i Henry Ford, bili su luksuzni predmeti dostupni samo malom broju ljudi. S vremenom su postali pristupačniji i danas predstavljaju osnovno sredstvo prijevoza za milijune ljudi širom svijeta. Kao što su se razvijale infrastrukture cesta i pametna signalizacija, tako su se razvijali i automobili. Uz rastuću popularnost automobila na cestama, povećava se i broj prometnih nesreća, što je dovelo do potrebe za razvojem naprednih sustava za pomoć vozačima (ADAS) i autonomnih vozila. Jedna od ključnih komponenti ovih sustava je detekcija linija voznih traka, koja se istražuje već desetljećima. Detekcija linija voznih traka pomaže u održavanju sigurnosti vozača i putnika, omogućavajući vozilima da prepoznaju granice traka i ostanu unutar njih. Detekcija voznih traka je aktivna istraživačka tema za mnoge sustave, poput sustava za upozorenje na napuštanje trake, adaptivnog tempomata, asistencije pri promjeni trake, asistencije pri skretanju, procjene preostalog vremena do sigurne promjene trake i potpuno autonomnih sustava vožnje [1]. Kroz godine timovi znanstvenika različitim pristupima i metodama pokušavaju poboljšati detekciju linija vozne trake. Algoritmi detekcije vozne trake mogu se svrstati u dvije glavne kategorije: tradicionalne algoritme zasnovane na računalnom vidu (*engl.* Traditional Computer Vision) i algoritme zasnovane na strojnom učenju (*engl.* Machine Learning Based Methods). Algoritmi detekcije linija vozne trake zasnovani na tradicionalnim tehnikama računalnog vida pokazali su visoku preciznost u detekciji linija na određenim vrstama cesta. Međutim, raznolikost prometnih scenarija, vrsta prometnih oznaka i vremenskih uvjeta predstavlja izazov za ove algoritme. S napretkom tehnologije, eksponencijalnim rastom procesorske snage računala i sve većom dostupnošću velikih skupova podataka, otvorena je mogućnost primjene algoritama dubokog učenja za detekciju linija vozne trake. Korištenje opsežnih skupova podataka za treniranje i testiranje umjetnih neuronskih mreža omogućava postizanje veće preciznosti u različitim prometnim uvjetima. Većina popularnih rješenja danas ima mnogo sličnosti u svom dizajnu, organizaciji i arhitekturi, a te sličnosti proizlaze iz činjenice da su primjene različitih algoritama dobro poznate i uspostavljene [2]. Zajednička karakteristika svih algoritama za detekciju linija vozne trake je ulazni podatak tj. slika snimljena kamerom postavljenom na prednjem dijelu automobila.

## 2.1 Algoritmi za detekciju linija voznih traka zasnovani na tehnikama tradicionalnog računalnog vida

Tradicionalni pristup detekciji linija voznih traka zasnovan na računalnom vidu oslanja se na klasične tehnike obrade slike i dobro uspostavljene algoritme. Iako duboke neuronske mreže (*engl.* Deep Neural Networks – DNN) daju jako dobre rezultate, tradicionalni pristup je još uvijek rasprostranjeniji. To je prije svega zato što tradicionalne tehnike ne troše velike količine resursa u usporedbi s dubokim neuronskim mrežama, može se učinkovito izvršavati čak i na hardveru sa slabijim specifikacijama i već dugo vremena je učinkovito korišten u različitim područjima i izvan automobilske industrije [2]. Algoritmi za detekciju voznih traka imaju zadatak locirati rubove voznih traka na primljenim slikama. Algoritmi koji su zasnovani na tradicionalnom pristupu za lociranje granica traka bez prethodnog modela ceste koja se gleda moraju imati mogućnost robusno djelovati u raznim uvjetima, uključujući slučajeve kada na slici postoji velika količina smetnji[3]. Navedene smetnje mogu uključivati različite vremenske uvjete kao što su smanjenje vidljivosti pojavom magle ili kiše, kao i snježni sloj na cesti koji prekriva linije. Još jedan zadatak koji algoritmi detekcije linija rješavaju su zaklonjene linije od strane drugih sudionika u prometu te loše oznake trake kao što su izlizane linije. U ovakvim situacijama jako je teško detektirati linije jer značajke prema kojima algoritam detektira linije nisu vidljive. Sustavi detekcije linija zasnovani na tradicionalnim tehnikama opisan u mnogim znanstvenim radovima se najčešće sastoji od tri glavna procesa: pred-obrada slike, detekcija linija i praćenje linija [4]. Na slici 2.1 je grafički prikaz najčešće spominjane arhitekture sustava.



Slika 2.1 Općeniti prikaz arhitekture sustava za detekciju linija zasnovane na tradicionalnim tehnikama računalnog vida

Tradicionalni pristupi detekciji voznih traka zasnovani su na dobro poznatim tehnikama obrade slike i koriste nekoliko ključnih koraka kako bi se postigla pouzdana detekcija traka.

1. Odabir regije od interesa (*engl.* Region Of Interest – ROI) - omogućuje fokusiranje na dio slike gdje se očekuje da se pojave oznake linija vozne trake. Ovaj dio smanjuje opterećenje računala i poboljšava preciznost detekcije. ROI je obično trapezoidni dio ispred vozila, koji pokriva područje ceste gdje se očekuju oznake linija vozne trake. Definiranjem ROI-a eliminiraju se nepotrebni dijelovi slike, što ubrzava procesiranje i smanjuje mogućnost pogrešnih detekcija [4].
2. Pred-obrađ - ključni korak u obradi slike koji se koristi za pripremu ulaznih podataka za daljnju analizu i obradu. Kvaliteta pred-obrađe izravno utječe na učinkovitost i točnost kasnijih koraka detekcije linija vozne trake. Ona najčešće uključuje nekoliko tehnika koje imaju za cilj poboljšati kvalitetu slike, smanjiti šum i pretvoriti sliku u format koji je lakši za daljnju obradu. Smanjenje šuma ključno je za poboljšanje kvalitete slike, što se najčešće postiže primjenom Gauss-ovog filtra (*engl.* Gaussian Blur). Gauss-ov filter zamućuje sliku, čime se uklanjaju visokofrekventni šumovi, dok se važni rubovi ne deformiraju previše. Uz smanjenje šuma, pretvorba prostora boja također je važan dio pred-obrađe. Slike se često pretvaraju iz RGB formata u slike u nijansama sive boje ili druge prostore boja poput HSV-a (*engl.* Hue, Saturation, Value). Pretvorba u sliku u nijansama sive boje pojednostavljuje obradu slike smanjenjem složenosti i omogućava algoritmima da se fokusiraju na intenzitet piksela umjesto na boje.
3. Detekcija linija - nakon što se detektiraju rubovi, sljedeći korak je identifikacija linija koje predstavljaju oznake traka. Jedna od najpopularnijih metoda koja se koristi je Hough-ova transformacija (*engl.* Hough Transformation). To je robusna tehnika koja transformira točke rubova u parametarski prostor i detektira ravne linije pronalaskom vrhova u akumulacijskom prostoru. Ova metoda je posebno učinkovita za detekciju ravnih linija, ali može imati ograničenu učinkovitost kod zakrivljenih linija [5].

Hough-ova transformacija predstavlja jednu od klasičnih tehnika računalnog vida koja se koristi za detekciju linija, kružnica i drugih geometrijskih oblika u digitalnim slikama. Nakon Canny-ovog operatora završena je pred-obrađ slike, što rezultira binarnom slikom s istaknutim rubovima. Na binarnoj slici sve vrijednosti su 0 ili 255, pri čemu vrijednosti 255 predstavljaju piksele koji pripadaju rubovima. Zbog toga se rubovi objekata na slici sada mogu opisati geometrijskim oblicima. U kontekstu detekcije linija vozne trake, Hough-ova transformacija omogućuje prepoznavanje pravocrtnih segmenata unutar slike, čak i u prisutnosti šuma ili

djelomičnog prekida linija. Hough-ova transformacija je tradicionalni algoritam u računalnom vidu, a osnovni princip je transformacija točkaka iz prostora slike u prostor parametara. Svaka točka na slici generira skup mogućih parametara koji predstavljaju potencijalne oblike (npr. linije) koji prolaze kroz tu točku. Ove parametarske vrijednosti se zatim akumuliraju u akumulacijskom prostoru. Maksimalne vrijednosti u ovom prostoru odgovaraju parametrima oblika koji su najvjerojatnije prisutni u slici. Za detekciju linija, Hough-ova transformacija koristi parametarski oblik linije.

$$y = ax + b \quad (2-1)$$

Parametar  $a$  može postići beskonačne vrijednosti zbog čega primjena ove formule predstavlja problem. Češće korišteni oblik koji se koristi za prikaz ravne linije u prostoru  $(\rho, \theta)$  je

$$\rho(x, y) = x \cos \theta + y \sin \theta, \quad (0 < \theta < \pi) \quad (2-2)$$

$\rho$  predstavlja udaljenost od ishodišta koordinatnog sustava do najbliže točke na liniji, a  $\theta$  kut između pravca okomitog (normale) na liniju i osi  $x$ . Svaka točka  $(x, y)$  na slici u izvornom prostoru generira sinusoidu u  $(\rho, \theta)$  prostoru. Ta sinusoidna krivulja predstavlja sve moguće kombinacije  $(\rho, \theta)$  koje odgovaraju pravcima kroz tu točku  $(x, y)$ . Kada više točkaka leži na istom pravcu u slici, njihove sinusoidne presijecaju se u jednoj točki u  $(\rho, \theta)$  prostoru. Ta točka presjeka odgovara parametrima pravca koji prolazi kroz sve te točke u izvornom prostoru slike. Na taj način Hough-ova transformacija omogućuje detekciju pravaca u slici prepoznavanjem presjeka najvećeg broja sinusoida u prostoru  $(\rho, \theta)$ .

Proces Hough-ove transformacije započinje prikupljanjem podataka, pri čemu se za svaku točku ruba na binarnoj slici izračunavaju svi mogući pravci koji mogu prolaziti kroz tu točku. Te se mogućnosti bilježe u prostoru za bilježenje podataka. Nakon ovog postupka, traže se najveće vrijednosti u tom prostoru koje predstavljaju najvjerojatnije pravce na slici. Ove maksimalne vrijednosti identificiraju linije koje su najvjerojatnije prisutne na slici. Konačno, parametri identificiranih pravaca koriste se za crtanje linija na originalnoj slici, čime se vizualizira rezultat Hough-ove transformacije. Međutim, Hough-ova transformacija može biti računalno zahtjevna, posebno za složenije oblike i veće slike. Optimizacije, poput korištenja ograničenog prostora za bilježenje podataka ili prilagođenih verzija Hough-ove transformacije, mogu pomoći u smanjenju ovih zahtjeva. Hough-ova transformacija ostaje ključna tehnika u računalnom vidu, posebno u primjenama koje zahtijevaju stabilnu detekciju geometrijskih struktura, poput automatske detekcije linija voznih traka u autonomnim vozilima.

Objavljeno je dosta radova koji opisuju različite postojeće algoritme detekcije voznih traka koristeći tehnike zasnovane na tradicionalnom pristupu računalnom vidu. U radu [1], autori opisuju jednostavan sustav za detekciju linija voznih traka zasnovan na linearnom modelu. Sustav koristi Canny operator za detekciju rubova u fazi pred-obrađe te prethodno opisanu Hough-ovu transformaciju za pronalaženje relevantnih linija u fazi naknadne obrade, omogućavajući određivanje parametara geometrijskog modela. Algoritam predložen u ovom radu je testiran na slikama snimljenim kamerama postavljenim u vozilima, pri čemu je postigao zadovoljavajuće rezultate. Model zahtijeva jasno vidljive linije konzistentnog oblika, a širina ceste mora biti konstantna ili s minimalnim odstupanjima. Rezultati pokazuju da algoritam pouzdano detektira trake na ravnim cestama s jasno definiranim oznakama. Prednosti modela uključuju njegovu jednostavnost i sposobnost obrade u stvarnom vremenu, omogućujući preciznu detekciju voznih traka u standardnim uvjetima vožnje bez visokih zahtjeva za računalnim resursima. Sustav ne uspijeva detektirati linije na slikama koje sadrže zakrivljene linije ili ekstremne vremenske uvjete.

U radu [6] predstavljen je novi pristup detekciji voznih traka pod nazivom HistWind, koji je zasnovan isključivo na tradicionalnim metodama računalnog vida, bez primjene dubokog učenja. Ovo rješenje je novo jer omogućava detekciju traka u stvarnom vremenu na ograničenom hardveru, što ga čini pogodnim za implementaciju u nižim razredima vozila bez snažnih procesorskih zahtjeva. Arhitektura rješenja sastoji se od tri glavna bloka: pred-obrađe slike, identifikacije vrhova histograma i detekcije pomoću kliznog prozora. Histogram predstavlja raspodjelu piksela u određenim svjetlosnim ili kolorimetrijskim vrijednostima unutar slike, čime se identificiraju značajke kao što su rubovi traka. HistWind algoritam je testiran na CULane [7] skupu podataka te je pokazao rezultate usporedive s rezultatima modela LaneNet, zasnovanog na dubokom učenju, uz znatno niže vrijeme izvršavanja na središnjoj procesorskoj jedinici (*engl.* Central Processing Unit - CPU). Prednost ovog rješenja leži u brzini izvršavanja i jednostavnosti primjene, dok su nedostaci povezani s ograničenjima u prepoznavanju u kompleksnijim uvjetima, poput izbljedjelih oznaka i dugih vozila koja zaklanjaju trake.

U radu [8] predstavljen je inovativan pristup detekciji voznih traka koji kombinira linearni model s algoritmom K-means klasteriranja kako bi se izdvojile najrelevantnije trake na autocestama. Ključna novost ovog rada je primjena K-means klasteriranja u kombinaciji s Hough-ovom transformacijom za identificiranje i grupiranje linija traka u različite kategorije. Rješenje je dizajnirano da koristi Hough-ovu transformaciju za inicijalno prepoznavanje rubova traka, nakon čega se primjenjuje K-means algoritam kako bi se reducirali smetnje poput drugih vozila i objekata na cesti te istaknule glavne trake. K-means klastering algoritam djeluje tako da prvo nasumično

postavi K centara u prostoru podataka, a zatim iterativno pridružuje podatke najbližem centru, stvarajući privremene grupe [9]. Nakon svake iteracije, središta grupa ažuriraju se prema prosječnim vrijednostima podataka unutar tih grupa. Kroz nekoliko iteracija, algoritam postiže stabilne klastere (grupe) koji reflektiraju najvažnije linije traka, ignorirajući smetnje poput drugih objekata na cesti. Algoritam [8] je testiran na skup podataka koji je snimio Institut za umjetnu inteligenciju i robotiku Sveučilišta Xi'an Jiaotong na autocesti, gdje je pokazao visoku preciznost i otpornost na smetnje uzrokovane preprekama u vidnom polju, kao što su vozila i objekti uz cestu. U usporedbi s prethodnim metodama, ovaj pristup postiže značajna poboljšanja u točnosti i robusnosti detekcije traka, posebno u uvjetima složenih vizualnih smetnji. Konkretno, rezultati pokazuju da algoritam uspješno detektira i klasificira trake s preciznošću od preko 90%, smanjujući pritom lažne pozitivne i negativne detekcije. Prednosti ovog pristupa uključuju njegovu robusnost u složenim scenarijima i prilagodljivost različitim uvjetima, dok su izazovi vezani za učinkovitost algoritma u uvjetima loše vidljivosti ili kod linija niske kvalitete na cestama.

Rad [10] predstavlja sustav za detekciju linija vozne trake, koji je zasnovan na kombinaciji inverzne perspektivne transformacije i Kalmanova filtra. Novost ovog rada leži u integraciji tih dviju tehnika, čime se omogućuje učinkovitije prepoznavanje linija vozne trake čak i u uvjetima promjenjivog osvjetljenja i zakrčenosti. Arhitektura predloženog rješenja počinje odabirom ROI-a, zatim se nastavlja s primjenom inverzne perspektivne transformacije, kojom se uklanjaju perspektivne distorzije u slici te se dobiva prikaz iz ptičje perspektive, što olakšava prepoznavanje linija na kolniku. Na ovu sliku primjenjuje se Kalmanov filter za praćenje i predviđanje položaja linija vozne trake u sljedećim okvirima. Rješenje je testirano na slikama iz Caltech [11] skupa podataka s različitim cestovnim uvjetima. Eksperimentalni rezultati pokazuju da predloženi sustav postiže visok stupanj točnosti i stabilnosti u detekciji linija vozne trake, pri čemu se pokazao prikladnim za rad u stvarnom vremenu. Prednosti ovog pristupa uključuju sposobnost prilagodbe promjenama uvjeta na cesti te stabilno praćenje linija kroz više kadrova, zahvaljujući Kalmanovom filteru. Međutim, nedostatak sustava je njegova osjetljivost na složenije cestovne uvjete, poput cesta s loše vidljivim ili oštećenim oznakama, gdje dolazi do izazova u održavanju točnosti detekcije. Unatoč tome, ovaj rad predstavlja značajan doprinos razvoju metoda za detekciju vozničkih traka u autonomnim sustavima vožnje, posebno u kontekstu zahtjeva za rad u stvarnom vremenu.

U radu [12] predstavljen je sustav za detekciju linija vozne trake koji dinamički integrira dva različita algoritma te omogućuje *online* evaluaciju njihove učinkovitosti. Novost ovog rada leži u adaptivnoj kombinaciji primarnog i sekundarnog algoritma za detekciju, pri čemu se

sekundarni aktivira kada primarni pokazuje nisku razinu pouzdanosti, čime se povećava robusnost sustava u različitim uvjetima na cesti. Arhitektura predloženog rješenja sastoji se od dvaju glavnih modula: primarnog modula za detekciju zasnovanog na *steerable* filterima i Hough-ovoj transformaciji te sekundarnog modula koji kombinira Gauss-ov mješoviti model za segmentaciju slike i RANSAC algoritam za prilagodbu modela linije. Sustav dinamički prelazi između ovih dvaju modula ovisno o razini pouzdanosti detekcije. Dodatno, implementiran je sustav za *online* evaluaciju koji koristi tehniku uzorkovanja i glasovanja za procjenu točnosti detekcije u stvarnom vremenu. Rješenje je testirano na dva podatkovna skupa, jedna je video snimljen na američkoj autocesti M1 na relaciji od Cranfielda do Northamptona, a druga je Caltech Road. Eksperimentalni rezultati pokazuju da sustav postiže visoku razinu točnosti u detekciji linija vozne trake te je sposoban za rad u stvarnom vremenu. Prednosti ovog pristupa uključuju povećanu otpornost na različite uvjete na cesti zahvaljujući dinamičkoj integraciji dvaju algoritama te mogućnost *online* evaluacije koja omogućuje pravovremenu detekciju i korekciju pogrešaka. Međutim, složenost sustava može dovesti do povećanih računalnih zahtjeva, što može biti izazov za implementaciju na platformama s ograničenim resursima.

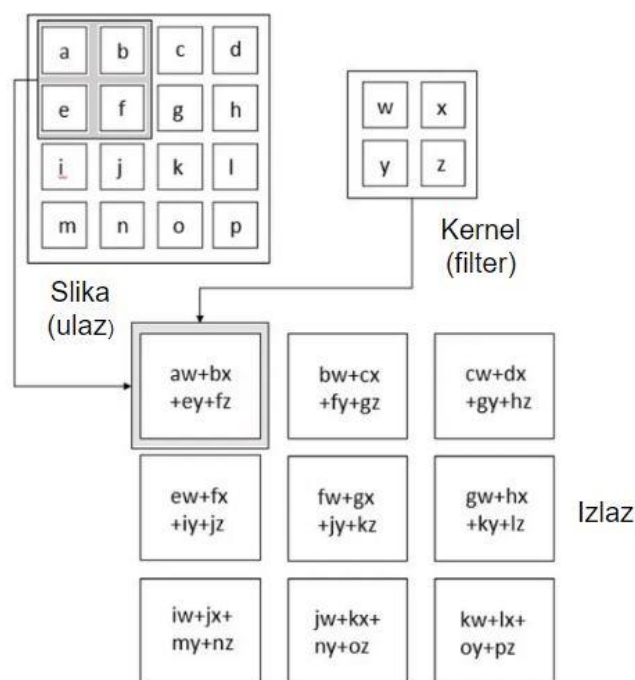
## **2.2 Algoritmi za detekciju linija voznih traka zasnovani na strojnom učenju**

Razvoj dubokog učenja označio je prekretnicu u mnogim područjima računalnog vida, uključujući detekciju linija vozne trake. Početkom 2010-ih, s pojavom snažnih grafičkih procesora (*engl.* Graphics Processing Unit - GPU) i velikih količina podataka, konvolucijske neuronske mreže (*engl.* Convolutional Neural Network - CNN) pokazale su izvanredne rezultate u zadacima prepoznavanja i klasifikacije slika. Ova tehnologija ubrzo je primijenjena u automobilskoj industriji, posebno u sustavima za pomoć vozačima (ADAS) i autonomnim vozilima, gdje je precizna i pouzdana detekcija voznih traka od ključne važnosti. Prvi koraci prema primjeni dubokog učenja u detekciji linija vozne trake bili su potaknuti uspjehom AlexNet mreže na ImageNet natjecanju 2012. godine, gdje je duboka mreža znatno nadmašila prethodne algoritme računalnog vida u zadatku klasifikacije slika. U ovom natjecanju, cilj je bio točno klasificirati slike u jednu od 1000 različitih kategorija, što je inspiriralo istraživače da istraže primjenu CNN-a u detekciji voznih traka. Tradicionalni algoritmi, poput Hough-ove transformacije i Canny-ovog detektora rubova, i dalje su bili učinkoviti, ali su se pokazali ograničenima u složenim scenarijima, poput zakrivljenih cesta, promjenjivih uvjeta osvjetljenja i prisutnosti šuma. CNN su vrsta dubokih neuronskih mreža koje su posebno učinkovite u zadacima obrade slike i prepoznavanja uzoraka. CNN su inspirirane organizacijom vizualnog korteksa životinja, gdje se neuroni aktiviraju samo u

prisutnosti određenih vizualnih značajki [13]. CNN su postale standard za zadatke poput klasifikacije slika, detekcije objekata, segmentacije slika i, naravno, detekcije linija vozne trake.

CNN se sastoje od nekoliko osnovnih slojeva, od kojih svaki ima specifičnu funkciju u procesu izdvajanja značajki i klasifikacije:

1. Konvolucijski sloj (*engl.* Convolutional Layer) - predstavlja temeljni građevni blok CNN-a gdje se odvija većina izračuna. Ključne komponente konvolucijskog sloja su filtri i aktivacijske mape. Na slici 2.2 je prikazan način rada konvolucijskog sloja. Ulazna slika u boji može se prikazati kao 3D matrica piksela, što znači da ima tri dimenzije - visinu, širinu i dubinu, koje odgovaraju RGB kanalima slike. Dio ulazne mreže koji je izdvojen za obradu se naziva lokalno receptivno polje. Detektor značajki, poznat i kao kernel ili filter, je matrica koja se pomiče preko različitih dijelova slike, tražeći prisutnost određenih značajki. Konvolucijski proces je proces gdje se svaki element receptivnog polja množi s odgovarajućim elementom kernela, a zatim se rezultati zbrajaju kako bi se dobila jedna vrijednost, kao što je prikazano na slici 2.2.



Slika 2.2 Prikaz rada konvolucijskog sloja (Slika preuzeta iz knjige [14], stranica 330)

Nakon konvolucijskog sloja, rezultat se prosljeđuje aktivacijskoj funkciji koja uvodi nelinearnost u model. Najčešće korištena aktivacijska funkcija u konvolucijskim mrežama je ReLU (*engl.* Rectified Linear Unit). Aktivacijska funkcija koja zamjenjuje sve negativne vrijednosti u aktivacijskoj mapi s nulom, dok pozitivne vrijednosti



ostavlja nepromijenjenima. Ovo omogućuje mreži da uči nelinearne odnose, što je ključno za složene zadatke prepoznavanja uzoraka.

2. Sloj sažimanja (*engl.* Pooling Layer) - koristi se za smanjenje dimenzija aktivacijske mape, čime se smanjuju broj parametara i računalni zahtjevi modela te djelomično kontrolira prekomjerno prilagođavanje (*engl.* Overfitting) na trening podatke [15]. Najčešće korištena tehnika je *max pooling*, koja za svaku podregiju aktivacijske mape zadržava maksimalnu vrijednost. Ovo omogućuje očuvanje najistaknutijih značajki dok se smanjuje količina podataka.
3. Potpuno povezani slojevi (*engl.* Fully Connected Layer) - Nakon nekoliko konvolucijskih slojeva i slojeva sažimanja, izlazne aktivacijske mape se "skupljaju" u jedan vektor i prosljeđuju kroz jedan ili više potpuno povezanih slojeva. Ovi slojevi djeluju kao tradicionalne potpuno povezane neuronske mreže i koriste se za klasifikaciju [15]. Potpuno povezani slojevi povezuju svaki neuron iz prethodnog sloja sa svim neuronima u trenutnom sloju, omogućujući mreži da kombinira značajke i donosi konačne odluke.
4. Izlazni sloj (*engl.* Output Layer) - posljednji sloj u CNN-u i daje konačne predikcije. Aktivacijska funkcija izlaznog sloja ovisi o prirodi zadatka. Na primjer, za zadatke višeklasne klasifikacije koristi se *softmax* funkcija, koja daje vektor vjerojatnosti za svaku klasu, dok se za binarnu klasifikaciju koristi aktivacijska sigmoid funkcija, koja daje vjerojatnost da ulaz pripada određenoj klasi.

U radu [16] predstavljen je algoritam detekcije linija voznih traka zasnovan na LaneNet (*engl.* Real-Time Lane Detection Networks for Autonomous Driving) arhitekturi, prilagođena za autonomnu vožnju. Novost ovog rada leži u primjeni instancijske segmentacije za prepoznavanje pojedinačnih linija trake, omogućujući sustavu da detektira promjenjiv broj linija i prilagodi se promjenama traka tijekom vožnje. Arhitektura predloženog rješenja sastoji se od dvaju glavnih modula: modula za binarnu segmentaciju koji razlikuje piksele linija trake od pozadine te modula za ugrađivanje instanci koji grupira piksele u pojedinačne linije. Nakon segmentacije, primjenjuje se prilagodljiva perspektivna transformacija, uvjetovana slikom, kako bi se parametarski opisale detektirane linije. Sustav je testiran na TuSimple [17] skupu podataka, gdje je postigao konkurentne rezultate s točnošću od 96,4% i F1 mjerom od 94,8%. Prednosti ovog pristupa uključuju sposobnost detekcije promjenjivog broja linija, prilagodbu promjenama traka te rad u stvarnom vremenu s brzinom od 50 sličica u sekundi. Međutim, sustav može biti osjetljiv na

složene cestovne uvjete, poput loše vidljivih ili oštećenih oznaka, što može utjecati na točnost detekcije.

U radu [18] autori predstavljaju novi model za detekciju linija voznih traka pod nazivom RESA (*engl.* Recurrent Feature-Shift Aggregator), koji poboljšava značajke linija nakon početnog izdvajanja koristeći CNN arhitekturu. Ovaj modul obogaćuje značajke linija nakon preliminarne ekstrakcije pomoću CNN. Novost rada leži u sposobnosti RESA modula da iskoristi snažne oblike linija i uhvati prostorne odnose piksela duž redaka i stupaca slike. RESA pomiče dijelove značajki rekurzivno u vertikalnom i horizontalnom smjeru, omogućujući svakom pikselu prikupljanje globalnih informacija, što rezultira preciznijom detekcijom linija. Sustav je testiran na dvama popularnim skupovima podataka za detekciju linija vozne trake: CULane i TuSimple. Na CULane skupu postignut je F1 rezultat od 75,3%, dok je na TuSimple skupu ostvarena točnost od 96,8%, što predstavlja vrhunske rezultate u usporedbi s postojećim metodama. Prednosti ovog pristupa uključuju poboljšanu točnost detekcije zahvaljujući globalnoj agregaciji značajki i mogućnost integracije RESA modula u druge mrežne arhitekture. Međutim, složenost modela može povećati računalne zahtjeve, što može biti izazov za implementaciju u stvarnom vremenu na uređajima s ograničenim resursima.

U radu [19] predstavljena je metoda za detekciju linija vozne trake koja kombinira strukturalnu analizu linija i CNN. Novost ovog rada leži u integraciji analize strukture linija sa značajkama dobivenim pomoću CNN-a, čime se postiže poboljšana točnost i robusnost detekcije linija vozne trake. Arhitektura rješenja uključuje preliminarnu analizu strukture linija kako bi se ograničio prostor pretrage, što smanjuje računalnu složenost i povećava učinkovitost metode. Zatim se koristi CNN za ekstrakciju značajki i identifikaciju točnih položaja linija vozne trake. Ova kombinacija omogućuje bolju detekciju i praćenje linija trake čak i u uvjetima s niskim kontrastom ili vizualnim preprekama. Ova metoda je testirana na dva već postojeća skupa podataka, Caltech podatkovni skup i Beijing skup podataka. Eksperimentalni rezultati pokazali su visoku razinu točnosti i otpornosti metode u različitim uvjetima, što potvrđuje korisnost kombinacije strukturalne analize i CNN-a. Prednosti ovog pristupa uključuju preciznu detekciju i povećanu otpornost na varijacije u cestovnim uvjetima, dok se kao potencijalni nedostatak može istaknuti povećana računalna složenost prilikom obrade velikih slika ili pri implementaciji u sustave s ograničenim resursima.

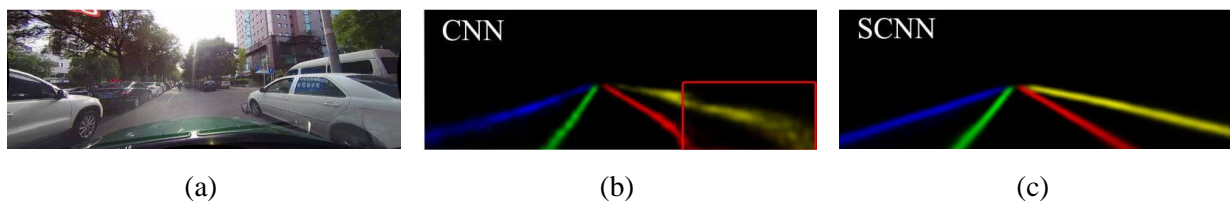
U radu [20] autori predstavljaju sustav za detekciju vozne površine i semantičku procjenu linija vozne trake u stvarnom vremenu, koristeći duboke značajke. Novost ovog rada leži u

integraciji dekonvolucijske neuronske mreže za segmentaciju vozne površine s modelom ekstra stabala za procjenu linija trake, što omogućuje simultanu detekciju vozne površine i linija trake, čak i u izazovnim uvjetima poput loše vidljivih ili odsutnih oznaka. Arhitektura predloženog sustava sastoji se od dvaju glavnih modula: dekonvolucijske mreže za segmentaciju vozne površine i modela ekstra stabala za procjenu linija trake. Dekonvolucijska mreža generira semantičku segmentaciju vozne površine, dok model ekstra stabala koristi duboke značajke iz mreže za procjenu položaja i semantičkih informacija linija trake. Sustav je testiran na dva prikupljena skupa podataka koje autori nazivaju TTI i TMD. Skupovi podataka uključujući stvarne cestovne scenarije, gdje je model postigao visoku točnost i robusnost u detekciji vozne površine i linija trake. Prosječno vrijeme obrade od 90 ms po slici omogućuje obradu približno 11 slika u sekundi, što je blizu granice za rad u stvarnom vremenu. Rad u stvarnom vremenu (*engl.* Real-Time Processing) podrazumijeva da sustav obavlja obradu podataka dovoljno brzo da rezultati budu dostupni u vremenskom okviru koji omogućuje trenutno donošenje odluka ili reakciju. Prednosti ovog pristupa uključuju sposobnost simultane detekcije vozne površine i linija trake, robusnost u različitim uvjetima vožnje te učinkovitost u stvarnom vremenu. Međutim, sustav može biti osjetljiv na složene cestovne uvjete, poput loše vidljivih ili oštećenih oznaka, što može utjecati na točnost detekcije.

CNN obično se grade sloj-po-sloj, konvolucijski sloj prima ulazne podatke od prethodnog sloja, primjenjujući potom operaciju konvolucije. Iako CNN pokazuje snažnu sposobnost ekstrakcije semantike iz sirovih piksela, nije u potpunosti istražena njihova sposobnost hvatanja prostornog odnosa piksela preko redaka i stupaca slike. Kao dio rada „*Spatial As Deep: Spatial CNN for Traffic Scene Understanding*“ [7] objavljenog 2017. predstavljena je nova vrsta konvolucijske neuronske mreže pod nazivom SCNN. Za razliku od sloj-po-sloj konvolucije koja se koristi u klasičnim CNN, SCNN koristi „isječak po isječak“ (*engl.* Slice by Slice) metodu konvolucije, gdje se operacije konvolucije, aktivacije i zbrajanja izvode preko redaka i stupaca mapi značajki, omogućujući prijenos informacija između neurona unutar istog sloja. Takav način konvolucije je pogodan za duge kontinuirane strukture s jakim prostornim odnosima, ali manje pojavnih tragova, kao što su prometne trake. SCNN generira mape vjerojatnosti koje se koriste za treniranje mreže, slično semantičkoj segmentaciji. Umjesto klasificiranja svih linija voznih traka kao jedne klase, cilj je razlikovati različite oznake linija [7].

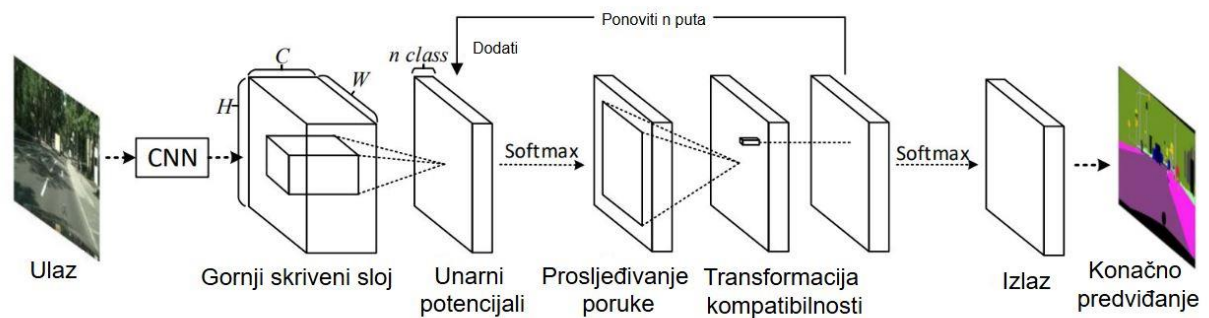
Tijekom testiranja, mape vjerojatnosti (*engl.* Probability Maps) se pretvaraju u krivulje. Za svaku oznaku linije s vjerojatnošću većom od 0.5, traži se pozicija s najvećim odgovorom svakih 20 redaka. Ove pozicije se povezuju *cubic spline* funkcijom, što rezultira konačnim predviđanjima.

Oznake linija se tretiraju kao linije širine 30 piksela, a presjek preko unije (*engl.* Intersect over Union - IoU) između predviđanja i stvarnih podataka (*engl.* Ground Truth - GT) koristi se za procjenu točnosti. Ako je IoU veći od zadanog praga, predviđanje se smatra točnim pozitivnim primjerom (*engl.* True Positive - TP). Ovakav pristup omogućuje visoku preciznost detekcije linija voznih traka u različitim prometnim scenarijima i vremenskim uvjetima. Prednost SCNN-a nad ostalim mrežama je u njegovoj sposobnosti da precizno predvidi linije čak i kada su zaklonjene, što je prikazano na slici 2.3. Pod (a) prikazan je primjer prometne situacije gdje je linija djelomično zaklonjena, preuzeta iz CULane podatkovnog skupa, a pod-slika (b) prikazuje kako klasični CNN označava linije, gdje se može primijetiti da linije nisu oštre nego pomalo raspršene. Slika pod (c) pokazuje sliku pod (a) označeno pomoću modela zasnovanog na SCNN. Kao što je vidljivo na slici, modeli zasnovani na SCNN predviđaju linije čak i kada su prekrivene te ih treba predvidjeti.

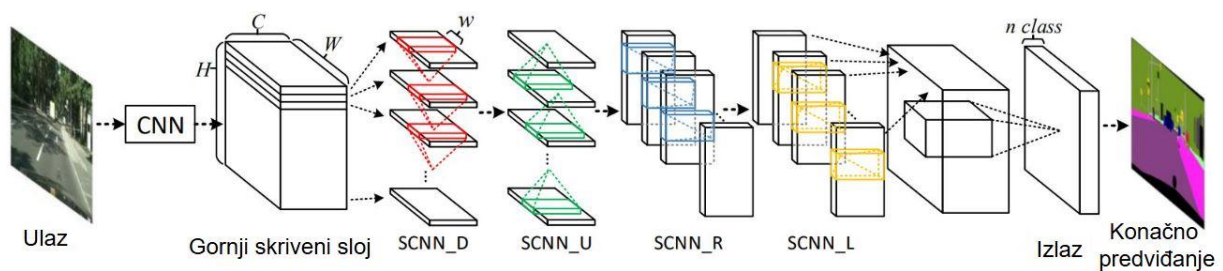


Slika 2.3 (a) slika iz CULane skupa podataka koja prikazuje sliku s djelomično prekrivenom linijom (b) predviđanje linija vozne trake za sliku (a) pomoću modela zasnovanog na CNN (c) predviđanje linija vozne trake za sliku (a) pomoću modela zasnovanog na SCNN (Slika preuzeta iz rada [7], stranica 1)

U radu [7] korištene su SCNN-VGG16 i SCNN-ResNet101 arhitekture kao osnove za razvoj modela. Tijekom testiranja modela znanstvenici su zaključili kako već postojeći skupovi podataka nisu zadovoljavajući te su kreirali vlastiti podatkovni skup nazvan CULane. Ovi modeli su trenirani i testirani na CityScapes [21] i CULane skupovima podataka, ocjenjujući ih za zadatke detekcije traka i semantičke segmentacije prometne scene. SCNN modeli su nadmašili dotadašnje algoritme, pokazavši superiornost u očuvanju kontinuiteta dugih i tankih struktura poput linija voznih traka. Usporedba arhitektura između klasične CNN arhitekture i SCNN-a, uključujući propagaciju informacija, prikazana je na slici 2.4. Kao što je prikazano u modulu 'SCNN D' na slici 2.4. (b), SCNN se primjenjuje na tenzor dimenzija  $C \times H \times W$ .  $C$  označava broj kanala,  $H$  broj redova, a  $W$  broj stupaca, tenzor se dijeli na  $H$  isječaka, pri čemu se prvi isječak šalje u konvolucijski sloj s jezgri širine  $w$ . Za razliku od klasičnih CNN-a, slika 4.2 (a), gdje izlaz konvolucijskog sloja postaje ulaz u sljedeći sloj, u SCNN-u se izlaz dodaje sljedećem isječku, generirajući novi isječak. Ovaj proces se ponavlja dok se svi isječci ne ažuriraju.



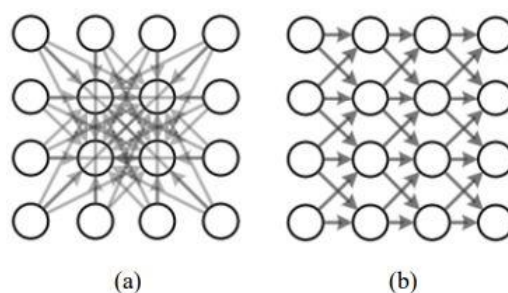
(a)



(b)

Slika 2.4 (a) Metoda zasnovana na MRF/CRF. (b) Implementacija prostornog CNN-a. (Slika preuzeta iz rada [7], stranica 3)

Prethodne metode za modeliranje prostornih odnosa, zasnovane na MRF (*engl.* Markov Random Fields) i CRF (*engl.* Conditional Random Fields), su računski intenzivne jer svaki element slike prima informacije od svih ostalih elemenata. Ova složenost čini ove metode neprikladnima za primjenu u stvarnom vremenu, posebno u automobilske industriji za zadatke kao što je detekcija linija voznih traka. SCNN, nasuprot tome, koristi učinkovitiji način prostorne propagacije informacija. Usporedba smjerova propagacije poruka između SCNN-a i MRF modela prikazana je na slici 2.5, ističući prednosti SCNN-a u smanjenju računske složenosti.



Slika 2.5 Smjerovi propagacije poruka u (a) gustim MRF/CRF modelima i (b) prostornom CNN-u (desno). Za (a), prikazana je samo propagacija poruka prema unutarnja 4 piksela radi jasnoće. (Slika preuzeta iz rada [7], stranica 4)

Kako se to može vidjeti na slici 2.5, SCNN ne samo da poboljšava točnost detekcije linija voznih traka, već također značajno smanjuje računске zahtjeve u usporedbi s MRF i CRF metodama. Ova efikasnost čini SCNN atraktivnim rješenjem za aplikacije koje zahtijevaju brzu i preciznu analizu, kao što su sustavi autonomne vožnje.

Nedostatak pristupa zasnovanog na dubokom učenju je potreba za velikim skupom podataka koji se koristi za treniranje duboke neuronske mreže (DNN) s informacijama koje bi model na kraju trebao biti sposoban prepoznati [2]. DNN zahtijevaju velike skupove podataka za treniranje kako bi postigle visoku točnost i robusnost. Razlog za to leži u prirodi dubokog učenja, koje se oslanja na učenje složenih obrazaca i značajki iz podataka. Što više podataka model ima na raspolaganju, to bolje može naučiti prepoznati raznolike scenarije i varijacije, čime se poboljšava njegova sposobnost generalizacije na novim, neviđenim podacima. Podaci koji pokrivaju različite uvjete, poput promjena u osvjetljenju, vremenskim uvjetima, kutovima snimanja i drugim varijacijama, omogućuju modelu da bude robusniji i pouzdaniji u stvarnim uvjetima.

## **2.3 Skupovi podataka s označenim linijama voznih traka**

Prikupljanje velikih i raznolikih skupova podataka može biti vrlo zahtjevno i skupo. Potrebno je mnogo resursa za snimanje, obradu i pohranu podataka, posebno ako su potrebni podaci visoke kvalitete. Osim toga, većina podataka mora biti ručno označena kako bi bila korisna za treniranje. Označavanje je proces u kojem ljudski stručnjaci identificiraju i označavaju značajke u podacima, što je vremenski zahtjevno i skupo. Treniranje DNN-a na velikim skupovima podataka zahtijeva značajne računalne resurse, uključujući moćne GPU-ove i puno memorije, što može biti skupo i tehnički zahtjevno. Prikupljanje i korištenje velikih količina podataka može dovesti do problema s privatnošću i etikom, posebno ako se prikupljaju osjetljivi ili osobni podaci. Zbog ovih izazova, istraživači i inženjeri stalno traže načine za smanjenje potrebe za velikim skupovima podataka, kao što su algoritmi prenesenog učenja, umjetno generirani podaci i napredne tehnike augmentacije podataka. Ovi algoritmi mogu pomoći u smanjenju količine potrebnih podataka, a da se pritom ne žrtvuje točnost i robusnost modela. U nastavku su opisani najpoznatiji i najčešće korišteni skupovi podataka za treniranje i razvoj modela za detekciju linija voznih traka.

### **2.3.1 TuSimple skup podataka**

TuSimple skup podataka je objavljena 2017. godine povodom TuSimple izazova detekcije linija voznih traka (*engl.* TuSimple Lane Detection Challengera) [17]. Ovaj skup podataka sadrži

ukupno 6.408 slika cesta s američkih cesta i autocesta. 3626 slika pripada trening skupu, 358 validacijskom skupu dok 2782 slika pripada skupu za testiranje, što omogućuje temeljitu obuku, validaciju i evaluaciju modela za detekciju linija voznih traka. Slike su snimljene u rezoluciji od 1280x720 piksela, pružajući dovoljno detalja za preciznu analizu i identifikaciju linija na cesti. Videozapisi iz kojih su izdvojene slike snimljeni su kamerom koja je postavljena na kontrolnoj ploči vozila u različitim voznim okruženjima, uključujući urbana, ruralna i prigradska područja. Snimke su zabilježene tijekom različitih vremenskih stanja, uključujući sunčano, oblačno i kišno vrijeme, što doprinosi robusnosti modela u stvarnim uvjetima vožnje. Nadalje, skup podataka uključuje snimke različitog doba dana, pružajući modelima mogućnost da se nose s različitim uvjetima osvjetljenja. Primjeri slika iz TuSimple skupa podataka prikazani su na slici 2.6.



*Slika 2.6 Prikaz slika iz TuSimple skupa podataka (a) brza i zakrivljena cesta, tri različita tipa linija uključujući i žutu liniju, malo prometa i dobra vidljivost (b) ravna cesta dobre vidljivosti, tri vozne trake, dosta prometa na cesti.*

Prema repozitoriju [17], TuSimple skup podataka koristi JSON (*engl.* JavaScript Object Notation) format za pohranu oznaka podataka vezanih uz detekciju linija voznih traka. Svaki redak u datoteci „*label\_data\_(date).json*“ predstavlja podatke za posljednji (20.) okvir u određenom isječku videa. U tablici 2.1 prikazani su nazivi i strukture koji opisuju slike iz TuSimple skupa podataka. U TuSimple oznakama može biti prisutno do pet linija voznih traka. Obično se očekuje da četiri linije odgovaraju trenutnoj voznoj traci te lijevim i desnim trakama, dok se peta linija može pojaviti u slučajevima promjene trake, gdje može biti teško odrediti koja je linija trenutna traka. Uređeni par elemenata listi *lanes* i *h\_samples* određuje položaj točke koja predstavlja oznaku linije na slici. JSON datoteka prikazana na slici 2.7 je primjer formata zapisa za slike iz TuSimple skupa podataka. Četiri linije unutar liste *lanes* označavaju da na cesti postoje četiri trake. U ovom primjeru, -2 označava da na određenoj visini (*h\_sample*) nema postojeće oznake linije trake. Prva linija trake prvi put postaje vidljiva na poziciji (632, 280). Unutar prve liste *lanes* vrijednost 632 je peti element u nizu te predstavlja poziciju x koordinate.



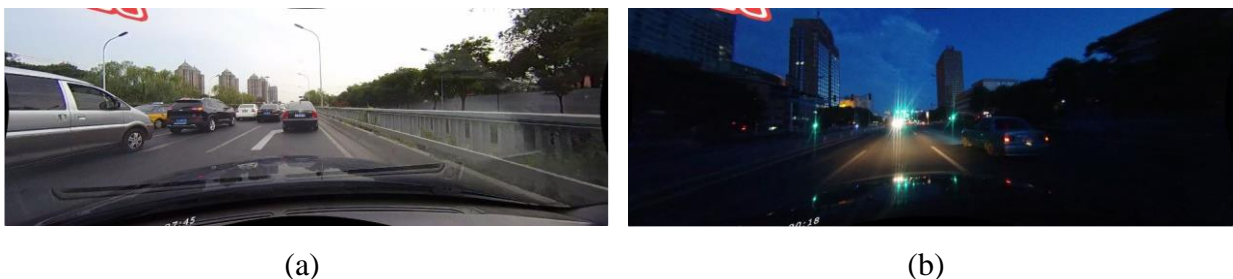




### 2.3.2 CULane skup podataka

Kinesko sveučilište u Hong Kongu je s ciljem stvaranja kompleksnog podatkovnog skupa za akademsko istraživanje u području detekcije linija voznih traka 2018. godine objavilo CULane skup podataka. Ona je jedna od najvažnijih skupova podataka za istraživanje i razvoj u području autonomne vožnje i detekcije linija voznih traka. CULane skup podataka sadrži ukupno 133235 slika visoke rezolucije 1640x590 piksela [7]. Skup podataka je podijeljena na tri kategorije od čega je 88880 slika za trening, 9675 slika za validaciju te 34680 slika za testiranje. Skup podataka je podijeljen u 9 kategorija na temelju vremenskih uvjeta, cestovnih uvjeta i oblika linija voznih traka na slici. Ovaj velik i raznolik skup podataka omogućuje temeljitu obuku modela, osiguravajući da su algoritmi sposobni prepoznati linije voznih traka u širokom rasponu vozačkih scenarija.

Podaci su prikupljeni u različitim voznim okruženjima, uključujući urbano okruženje, autoceste, prigradske ceste i ruralna područja. Dodatno, slike su snimljene u različitim vremenskim uvjetima, uključujući sunčano, oblačno, kišno i snježno vrijeme, što doprinosi robusnosti modela, omogućujući im prilagodbu na promjene u uvjetima vožnje. Skup podataka također uključuje i dnevne i noćne snimke, što je ključno za razvoj modela koji mogu pouzdano funkcionirati u različitim uvjetima osvjetljenja, uključujući vožnju po mraku. Na slici 2.8 prikazana su 2 primjera slika koje se nalaze u CuLane skupu podataka snimljenih u različitim uvjetima.



*Slika 2.8 Prikaz slika iz CULane skupa podataka snimljenih u različitim scenarijama: (a) gradska vožnja, ravna cesta, sunčano, gužva (b) gradska vožnja, ravna cesta, noć, slaba vidljivost linija voznih traka, odsjaj*

Oznake u CULane skupu podataka pohranjene su u formatu .txt datoteka, gdje svaka datoteka sadrži precizne koordinatne točke koje definiraju položaj linija voznih traka na svakoj slici. Na slici 2.9 prikazan je sadržaj .txt datoteke za jednu sliku iz CULane skupa podataka. Svaki redak unutar datoteke označava jednu liniju na cesti. Svaku točku linija označava uređeni par koji se dobiva pomoću *cubic spline* funkcije. Nakon dobivene jednadžbe funkcije za svaku desetu  $y$  koordinatu se računa odgovarajuća  $x$  koordinata. Na primjeru sa slike prvi par koordinata za prvu liniju slike je (256.114, 590), dok je drugi par (276.365, 580).

```

256.114 590 276.365 580 294.654 570 312.07 560 329.481 550 347.757 540 365.154 530 382.542 520 400.414 510 417.777 500 435.127 490
452.46 480 469.776 470 487.072 460 504.346 450 521.234 440 538.461 430 554.803 420 571.977 410 588.264 400 604.522 390 621.277 380
637.488 370 653.722 360 671.098 350 687.497 340 705.996 330 724.49 320 743.584 310 763.229 300 783.104 290

1220.67 590 1210.05 580 1196.42 570 1182.79 560 1169.95 550 1156.31 540 1142.65 530 1128.97 520 1116.03 510 1102.32 500 1088.59 490
1074.84 480 1061.07 470 1046.45 460 1033.26 450 1018.59 440 1004.7 430 989.957 420 976.002 410 961.19 400 947.166 390 931.833 380
917.756 370 902.451 360 888.476 350 874.625 340 860.697 330 848.19 320 836.537 310 825.54 300 813.812 290

1652.75 440 1598.91 430 1544.83 420 1490.75 410 1436.67 400 1382.58 390 1328.85 380 1273.77 370 1219.68 360 1165.28 350 1111.37 340
1058.3 330 1006 320 955.482 310 904.455 300 853.32 290

```

*Slika 2.9 Prikaz .txt datoteke za sliku iz CULane skupa podataka*

Unatoč mnogim prednostima, CULane skup podataka ima i određene nedostatke. Jedan od glavnih izazova je nedostatak snimaka u ekstremnim vremenskim uvjetima, poput guste magle ili vrlo intenzivne kiše, što može predstavljati problem za modele kada se suoče s takvim uvjetima u stvarnom svijetu. Osim toga, iako skup podataka obuhvaća različite vrste vozačkih okruženja, postoji relativna dominacija urbanih scenarija, što može ograničiti učinkovitost modela u manje zastupljenim ruralnim ili vrlo specifičnim voznim uvjetima. Ipak, CULane skup podataka ostaje jedan od najvažnijih i najkorištenijih resursa u zajednici za istraživanje autonomne vožnje, pružajući bogat i raznolik skup podataka koji omogućuje napredak u razvoju tehnologija za sigurnu i pouzdanu autonomnu vožnju.

### **2.3.3 LLAMAS skup podataka**

LLAMAS (*engl.* Labeled Lane Markers) podatkovni skup predstavlja jedan od važnijih skupova podataka u području autonomnih vozila i računalnog vida, posebno u kontekstu detekcije oznaka na cestama. Ovaj je skup podataka prvi put predstavljen 2019. godine od strane istraživača iz tvrtke Bosch, s ciljem poboljšanja performansi algoritama za prepoznavanje i praćenje traka na cestama [22]. Oznake linija voznih traka označene su pomoću automatiziranog algoritma za označavanje, koristeći precizne mape napravljene pomoću LIDAR-a. LLAMAS sadrži slike podijeljene u tri skupa: trening, validacija i test. Skup za trening obuhvaća oko 58269 slika, dok skup za validaciju sadrži 20844 slika i skup za testiranje sadrži 20929 slika, što ukupno čini 100042 slike. Za testiranje i evaluaciju modela za detekciju linija voznih traka koristi se validacijski skup jer oznake linija vozne trake koje predstavljaju *ground truth* za testni skup nisu javno dostupne. Što se tiče rezolucije slika, sve slike u LLAMAS skupu imaju rezoluciju od 1276 x 717 piksela te su spremljene u PNG (*engl.* Portable Network Graphic) formatu. Ove slike su snimane u različitim voznim okruženjima, uključujući urbana, ruralna i autoceste. Time je osigurana raznovrsnost scena koje odražavaju stvarne uvjete u kojima se autonomna vozila kreću. LLAMAS slike su snimane u različitim vremenskim uvjetima, uključujući sunčano, oblačno i djelomično oblačno vrijeme, što dodatno doprinosi varijabilnosti skupa podataka. Na slici 2.10 prikazana su 2 primjera slika koje se nalaze unutar LLAMAS podatkovnog skupa.



(a)



(b)

*Slika 2.10 Prikaz slika iz LLAMAS podatkovnog skupa (a) brza i ravna cesta, malo prometa, sunčani uvjeti i dobra vidljivost (b) brza i ravna cesta, djelomična naoblaka, jako dobra vidljivost bez prometa na cesti*

Oznake u LLAMAS skupu podataka predstavljene su u JSON formatu, koji sadrži precizne informacije o pozicijama traka i vrstama oznaka na voznim trakama. Svaka slika je označena tako da uključuje koordinate linija i njihove klasifikacije, što čini skup podataka vrlo pogodnim za treniranje modela detekcije traka.

LLAMAS skup podataka, iako vrijedan u kontekstu istraživanja računalnog vida i autonomnih vozila, ima nekoliko nedostataka koji mogu ograničiti njegovu upotrebljivost u određenim situacijama. Prvo, sve snimke u ovom skupu podataka napravljene su isključivo tijekom dana, što znači da nedostaju noćni uvjeti vožnje. Ova činjenica može predstavljati problem prilikom treniranja modela koji trebaju biti učinkoviti u uvjetima slabog osvjetljenja. Također, vremenski uvjeti u kojima su snimke napravljene su ograničeni. LLAMAS skup podataka sadrži snimke snimljene u sunčanim, oblačnim i djelomično oblačnim uvjetima, ali nedostaju snimke pod ekstremnijim vremenskim uvjetima kao što su kiša, snijeg ili magla. Konačno, iako su snimke zabilježene u različitim voznim okruženjima poput urbanih, ruralnih i autocesta, geografska raznolikost skupa podataka je relativno ograničena, što je vidljivo na slici 2.10. To može dovesti do problema prilikom primjene modela u različitim regijama koje nisu dovoljno zastupljene u skupu podataka. Sve ove mane ukazuju na potrebu za dodatnim skupovima podataka ili proširenjima LLAMAS podatkovnog skupa kako bi se osigurala bolja robusnost i generalizacija modela u stvarnim uvjetima vožnje.

### **2.3.4 KITTI skup podataka**

KITTI (*engl.* Karlsruhe Institute of Technology and Toyota Technological Institute) skup podataka jedan je od najpoznatijih i najkorištenijih u području računalnog vida, posebno u kontekstu istraživanja i razvoja autonomnih vozila. KITTI podatkovni skup sadrži niz podskupova podataka prilagođenih za različite zadatke, uključujući detekciju objekata, stereoskopsko gledanje,

vizualnu odometriju i detekciju linija voznih traka. Upravo ovaj posljednji segment, detekcija linija voznih traka, ima značajnu ulogu u omogućavanju sigurnog i preciznog upravljanja autonomnim vozilima.

Podskup za detekciju linija u KITTI skupu podataka sadrži ukupno 579 slika [23]. Ove slike su podijeljene na dva skupa: 289 slika pripada trening skupu, dok preostalih 290 slika čini testni skup. Rezolucija slika iznosi 1242x375 piksela, što osigurava dovoljno detalja za precizno identificiranje i označavanje linija voznih traka. Snimke su napravljene u urbanim područjima grada Karlsruhe-a i njegove okolice, obuhvaćajući različite prometne situacije, od neoznačenih i jednostavno označenih cesta do kompleksnijih prometnih traka s višestrukim oznakama. Snimanje je provedeno u dnevnim uvjetima, uz varijacije u osvjetljenju koje uključuju sunčano i oblačno vrijeme. Nema snimaka u ekstremnim vremenskim uvjetima poput kiše ili snijega, niti noćnih snimaka, što predstavlja jedno od ograničenja ovog skupa podataka. Podaci su prikupljeni vožnjom po stvarnim prometnim situacijama, što doprinosi vjerodostojnosti skupa, no također implicira i određenu neujednačenost u vrstama i kvaliteti snimljenih scena. Na slici 2.11 prikazani su primjeri scenarija unutar slika za KITTI podatkovni skup.

Oznake linija voznih traka generirane su ručnim označavanjem na slikama, a zatim su doručene u dvodimenzionalnom BEV (*engl.* Bird's Eye View) prostoru kako bi se postigla veća preciznost. Ove oznake uključuju informacije o poziciji i tipu linija (pune ili isprekidane), kao i njihovoj orijentaciji u odnosu na vozilo. Međutim, KITTI podskup za detekciju linija voznih traka ima nekoliko značajnih nedostataka. Prvo, ograničen je na dnevne uvjete snimanja, što može predstavljati problem u razvoju modela koji trebaju raditi u uvjetima slabog osvjetljenja ili tijekom noći. Osim toga, nedostatak snimaka u različitim vremenskim uvjetima, poput kiše ili snijega, ograničava sposobnost modela da pouzdano radi u svim realnim voznim situacijama. Konačno, relativno mala veličina skupa podataka, s ukupno 579 slika, može biti nedovoljna za treniranje pouzdanih modela, što može zahtijevati kombiniranje s drugim skupovima podataka.



*Slika 2.11 Prikaz slika iz KITTI podatkovnog skupa podataka (a) ravna cesta kroz urbano područje, nedostatak linija na cesti, parkirana vozila na kolniku (b) ravna dionica na brzjoj cesti, različiti tipovi linija, dosta sudionika prometa.*



### 2.3.5 BDD100K skup podataka

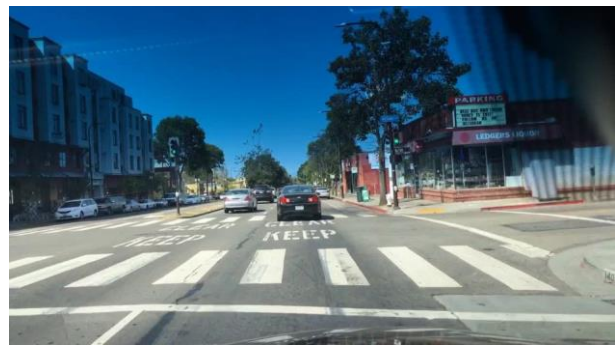
BDD100K je opsežan skup podataka koji je razvijen na UC Berkeley i objavljen 2020. godine s ciljem podrške istraživanju autonomnih vozila [24]. Skup sadrži 100000 videozapisa snimljenih s više od 50000 vožnji koje pokrivaju regije New York-a, San Francisco Bay-a i ostalih regija diljem Sjedinjenih Američkih Država. Obuhvaćajući širok raspon uvjeta vožnje, vremenskih uvjeta i geografskih lokacija.

Detekcija linija voznih traka u BDD100K podatkovnom skupu od posebne je važnosti jer omogućuje autonomnim vozilima da precizno održavaju svoju putanju na cesti. Ovaj skup podataka sadrži slike bogato označene informacijama o linijama voznih traka, koje su ručno označene kako bi se osigurala visoka razina točnosti. 70000 slika pripada trening skupu, 20000 testnom skupu i 10000 validacijskom skupu. Oznake uključuju pozicije i tipove linija grupirane su u 8 glavnih kategorija: rubnjak ceste, pješački prijelaz, dvostruka bijela linija, dvostruka žuta linija, dvostruke linije ostalih boja, jednostruka bijela linija, jednostruka žuta linija, jednostruka linija ostalih boja.

BDD100K obuhvaća slike snimljene u različitim uvjetima vožnje, uključujući urbane, prigradske i ruralne sredine. Osim toga, slike su zabilježene pod različitim vremenskim uvjetima, uključujući sunčano, oblačno, kišno i maglovito vrijeme, kao i noćne uvjete. Skup podataka također uključuje ekstremne vremenske uvjete kao što je prikazan na slici 2.7 (a), i različite scenarije na cestama koji se nalaze u stvarnom životu prikazano na slici 2.7 (b). Ova raznolikost uvjeta omogućuje treniranje modela koji mogu raditi u širokom rasponu stvarnih situacija, što je ključno za razvoj robusnih sustava autonomne vožnje.



(a)



(b)

*Slika 2.12 Prikaz slike iz BDD100K skupa podataka snimljenih u različitim scenarijima: (a) ekstremni uvjeti, kiša, slaba vidljivost ceste, gradska vožnja (b) gradska vožnja, sunčano, jako dobra vidljivost, oznake na cesti*

Jedna od glavnih prednosti BDD100K skupa podataka je njezina raznolikost, koja omogućuje modelima da generaliziraju na različite uvjete vožnje. Međutim, s obzirom na veliku količinu podataka, treniranje modela na cijelom skupu može biti računalno zahtjevno, što zahtijeva značajne resurse. Osim toga, iako skup podataka pokriva mnoge uobičajene situacije vožnje, neke specifične ili rijetke situacije možda nisu dovoljno zastupljene, što može ograničiti sposobnost modela da se prilagodi u svim mogućim uvjetima.

### 2.3.6 Osvrt na predstavljene podatkovne skupove

Na kraju ovog poglavlja, za lakšu usporedbu i preglednost, još jednom su sažete najvažnije karakteristike predstavljenih skupova podataka. Zbirni pregled ovih ključnih značajki prikazan je u Tablici 2.2, što omogućuje brzo i jasno uočavanje glavnih razlika i sličnosti među pojedinim skupovima podataka.

Tablica 2.2 Zbirni pregled glavnih karakteristika postojećih skupova podataka za analizu i vrednovanje metoda za detekciju linija vozne trake

SKUP PODATAKA	TuSimple [17]	CULane [7]	LLAMAS [22]	KITTI [23]	BDD100K [24]
UKUPAN BROJ SLIKA	6408	133235	100042	579	1000000
BROJ SLIKA U TRENING SKUPU	3626	88880	58269	289	70000
BROJ SLIKA U VALIDACIJSKOM SKUPU	358	9685	20844	/	10000
BROJ SLIKA U TESTNOM SKUPU	2782	34680	20929	290	20000
REZOLUCIJA SLIKE	1280 x 720	1640 x 590	1276 x 717	1242 x 375	1280 x 720
FORMAT SLIKE	JPG	JPG	PNG	PNG	JPG
FORMAT OZNAKA LINIJA	Polilinije (JSON)	<i>Cubic Splines</i>	Oznake na razini piksela, 2D isprekidane linije	<i>Ego Lane Area</i>	Klase, 2D Poligoni s vrhovima
DOBA DANA	Dan	Dan, noć	Dan	Dan	Dan, noć
VREMENSKI UVIJETI	Sunčano, oblačno, sjena,	Sunčano, oblačno, odsjaj, sjena, kiša	Sunčano, oblačno	Sunčano, oblačno	Sunčano, oblačno, kiša, snijeg, magla
TIPOVI CESTA	Autoceste	Ruralna područja, autoceste, tuneli, gradska vožnja	Autoceste	Ruralna područja, autoceste, gradska vožnja	Ruralna i gradska područja

U praktičnom dijelu ovog diplomskog rada odabrane su mreže trenirane na TuSimple, CULane i LLAMAS podatkovnim skupovima. Rezultati dobiveni na tim podatkovnim skupovima uspoređeni su s rezultatima dobivenih na novokreiranom skupu podataka u sklopu ovog rada. Razlozi korištenja ovih podatkovnih skupova leže u njihovoj raznolikoj primjenjivosti i širokoj uporabi u znanstvenim istraživanjima vezanim uz računalni vid i autonomne sustave vožnje. CULane i TuSimple skupovi postali su svojevrsni standard za zapis oznaka linija voznih traka zbog njihove precizne strukture oznaka i bogatstva različitih scenarija vožnje.

## **2.4 Postojeći alati za označavanje linija voznih traka**

Alati za označavanje linija voznih traka igraju ključnu ulogu u stvaranju kvalitetnih skupova podataka za autonomna vozila. Njihova primjena omogućava precizno označavanje voznih traka u slikama i videozapisima, što je od izuzetne važnosti za razvoj i treniranje modela za detekciju i praćenje traka. U ovom dijelu predstaviti će se postojeći alate koji se koriste za ovu namjenu, ističući njihove glavne značajke, tehnologije koje koriste te njihove prednosti i nedostatke.

### **2.4.1 LabelMe alat za označavanje**

LabelMe je jedan od prvih i najpoznatijih alata za ručno označavanje slika, razvijen na *Massachusetts Institute of Technology - MIT*. Ovaj alat, koji je slobodno dostupan i otvorenog koda, osmišljen je kako bi omogućio istraživačima i inženjerima jednostavan način za kreiranje oznaka na slikama [25]. Primarna svrha LabelMe alata je pružiti platformu za označavanje objekata u slikama kroz crtanje poligona, linija ili drugih oblika, čime se stvara visokokvalitetan skup podataka za treniranje modela strojnog učenja. Jedna od glavnih prednosti LabelMe-a je njegova jednostavnost i fleksibilnost. Alat se koristi putem web-preglednika, što znači da korisnici mogu pristupiti i raditi na svojim projektima s bilo kojeg mjesta. Korisnici mogu ručno crtati poligone oko objekata, poput vozila, pješaka ili linija voznih traka, kako bi precizno označili relevantne dijelove slike. Ove oznake se zatim spremaju u obliku JSON datoteka, koje se lako mogu integrirati u različite sustave za računalni vid i strojno učenje [26].

LabelMe je posebno koristan u istraživačkim projektima gdje je potrebna visoka razina preciznosti u označavanju, ali i u slučajevima gdje su dostupni resursi ograničeni. Međutim, ovaj alat također ima i svoje nedostatke. Budući da je alat ručni, proces označavanja može biti vrlo dugotrajan, posebno kada je riječ o velikim skupovima podataka. Osim toga, kvaliteta oznaka ovisi o preciznosti i dosljednosti korisnika, što može dovesti do varijabilnosti u podacima. Unatoč tim

izazovima, LabelMe ostaje jedan od najkorištenijih alata za označavanje slika u zajednici računalnog vida, pružajući osnovu za mnoge napredne aplikacije i istraživanja u ovom području.

## 2.4.2 VIA alat za označavanje

VIA (*VGG Image Annotator*) je jednostavan, lagan i svestran alat za ručno označavanje slika i videozapisa, razvijen od strane *Visual Geometry Group* - VGG na Sveučilištu Oxford. Namijenjen je istraživačima i inženjerima u području računalnog vida, koji trebaju kreirati visokokvalitetne oznake za treniranje modela strojnog učenja. Alat je otvorenog koda i besplatno dostupan, što ga čini pristupačnim za širok spektar korisnika, od akademske zajednice do industrije [27]. VIA se ističe po svojoj jednostavnosti i minimalnim zahtjevima za instalaciju. Budući da je alat potpuno web-baziran, može se pokrenuti izravno u web-pregledniku bez potrebe za bilo kakvom instalacijom ili vanjskim ovisnostima. To omogućuje korisnicima da započnu s radom gotovo odmah, bez komplikacija koje često prate druge, složenije alate za označavanje.

Glavna funkcionalnost VIA alata uključuje ručno označavanje slika i videozapisa pomoću različitih oblika kao što su poligoni, pravokutnici, krivulje ili točke. Korisnici mogu precizno označiti objekte, kao što su vozila, pješaci ili linije voznih traka, te dodati pripadajuće attribute kao što su oznake klase ili komentari. Oznake se spremaju u jednostavnom JSON formatu, koji je lako prenosiv i može se integrirati u različite sustave i radne tokove za računalni vid. Unatoč jednostavnom sučelju, VIA omogućuje složene prilagodbe i proširenja, čime zadovoljava potrebe raznih korisnika. Osim toga, njegova mala veličina, cijeli alat dolazi u jednoj HTML datoteci, omogućuje jednostavno dijeljenje i pokretanje na različitim uređajima, čak i bez internetske veze. Međutim, kao i svaki alat, VIA ima i svoje nedostatke. Ručno označavanje može biti dugotrajan proces, pogotovo kod većih skupova podataka. Također, alat ne pruža napredne mogućnosti automatizacije ili poluautomatizacije koje nude neki drugi moderni alati za označavanje. Unatoč tome, VIA je izvrstan izbor za projekte koji zahtijevaju prilagodljivo i lako dostupno rješenje za ručno označavanje.

## 2.4.3 Labelbox alat za označavanje

Labelbox je napredna platforma za označavanje podataka koja je dizajnirana kako bi ubrzala i olakšala proces kreiranja visokokvalitetnih skupova podataka za modele strojnog učenja. Ovaj alat nudi cjelovito rješenje koje integrira alate za označavanje, upravljanje podacima i treniranje modela, čineći ga jednim od najpopularnijih alata u industriji za projekte računalnog vida, obrade prirodnog jezika (*engl.* Natural Language Processing - NLP) i drugih područja umjetne inteligencije. Jedna od glavnih prednosti Labelbox-a je njegova sposobnost podrške za



razne vrste oznaka, uključujući označavanje slika, videozapisa, tekstualnih podataka i 3D podataka [28]. Alat omogućuje korisnicima da precizno označe objekte koristeći različite tehnike, kao što su poligoni, pravokutnici, točke i linije, što je posebno korisno u zadacima poput detekcije objekata, semantičke segmentacije i detekcije linija voznih traka. Svi podaci se spremaju u standardiziranim formatima, što olakšava njihovu upotrebu u raznim alatima za treniranje modela.

Labelbox se ističe i svojim mogućnostima timske suradnje i upravljanja projektima. Platforma omogućuje korisnicima da rade zajedno na istim projektima, dijele zadatke označavanja i prate napredak u stvarnom vremenu. Ovo je posebno korisno za velike timove i organizacije koje trebaju koordinirati rad više ljudi na kompleksnim projektima. Osim toga, Labelbox nudi integraciju s raznim alatima za analizu podataka i treniranje modela, omogućujući iteracije koje poboljšavaju i unapređuju modele strojnog učenja. Jedna od ključnih značajki Labelbox-a je mogućnost automatizacije procesa označavanja pomoću modela strojnog učenja. Korisnici mogu trenirati vlastite modele ili koristiti unaprijed trenirane modele za automatsko označavanje podataka, što znatno smanjuje vrijeme i trud potreban za ručno označavanje. Ova mogućnost poluautomatizacije omogućuje brže stvaranje velikih skupova podataka uz održavanje visoke razine točnosti. Međutim, Labelbox, kao i svaki alat, ima i određene izazove. Platforma je komercijalni alat, što znači da može biti skuplja od nekih drugih rješenja, posebno za manje timove ili projekte s ograničenim budžetom. Također, unatoč svojim naprednim značajkama, kompleksnost alata može zahtijevati određeno vrijeme za učenje i prilagodbu, posebno za korisnike koji nisu upoznati s naprednim tehnikama označavanja.

#### **2.4.4 CVAT alat za označavanje**

CVAT (*engl.* Computer Vision Annotation Tool) je jedan od najmoćnijih i najpopularnijih alata za označavanje podataka u području računalnog vida. Razvijen od strane tvrtke Intel, CVAT je otvorenog koda i besplatno dostupan, što ga čini pristupačnim izborom za širok spektar korisnika, od istraživača i inženjera do velikih organizacija koje se bave razvojem modela strojnog učenja [29]. CVAT je dizajniran kako bi omogućio učinkovito i precizno označavanje slika i videozapisa za različite zadatke računalnog vida, uključujući detekciju objekata, semantičku segmentaciju, praćenje objekata i detekciju linija voznih traka. Alat podržava razne vrste oznaka, poput pravokutnika, poligona, linija i točaka, što ga čini svestranim alatom za širok spektar primjena. Korisnici mogu jednostavno kreirati i upravljati oznakama, prilagođavati postavke projekta te izvoziti podatke u različitim formatima, kao što su XML, JSON ili COCO, koji su kompatibilni s popularnim okvirima za strojno učenje.

Jedna od ključnih prednosti CVAT-a je njegova podrška za rad s velikim skupovima podataka i složenim projektima. Alat omogućuje višekorisnički pristup, što znači da više korisnika može raditi na istom projektu istovremeno, što je posebno korisno za veće timove ili organizacije. CVAT također nudi napredne značajke kao što su automatsko označavanje pomoću unaprijed treniranih modela, čime se smanjuje vrijeme potrebno za ručno označavanje, te mogućnost prilagodbe sučelja i funkcionalnosti prema specifičnim potrebama korisnika [30]. CVAT je poznat po svojoj skalabilnosti i fleksibilnosti, što ga čini idealnim za projekte svih veličina, od manjih istraživačkih zadataka do velikih industrijskih aplikacija. Njegovo intuitivno korisničko sučelje omogućuje lako označavanje i upravljanje podacima, dok napredne opcije omogućuju duboku prilagodbu i optimizaciju radnih tokova. Integracija s popularnim alatima za verzioniranje podataka, kao što su Git, dodatno poboljšava mogućnosti upravljanja projektima i kolaboracije.

Iako je CVAT izuzetno moćan alat, njegova kompleksnost može predstavljati izazov za korisnike koji su novi u području računalnog vida ili koji nemaju iskustva s naprednim tehnikama označavanja. Također, budući da je riječ o otvorenom kodu, korisnici ponekad mogu naići na tehničke poteškoće koje zahtijevaju dodatne resurse za rješavanje.

#### **2.4.5 Deepen AI alat za označavanje**

Deepen AI je napredna platforma za označavanje podataka, specijalizirana za autonomna vozila i druge aplikacije u području računalnog vida [31]. Osnovan s ciljem ubrzanja i optimizacije procesa označavanja podataka, Deepen AI koristi kombinaciju ručnog označavanja i naprednih algoritama strojnog učenja kako bi omogućio brzo i precizno kreiranje visokokvalitetnih skupova podataka. Platforma je posebno dizajnirana za zadatke poput detekcije objekata, segmentacije, praćenja objekata i detekcije linija voznih traka, s fokusom na upotrebu u složenim i sigurnosno kritičnim aplikacijama kao što su autonomna vozila.

Jedna od glavnih prednosti ove platforme je njezina sposobnost automatizacije velikog dijela procesa označavanja. Koristeći unaprijed trenirane modele strojnog učenja, platforma može automatski generirati oznake koje se zatim mogu ručno pregledati i doraditi. Ovaj pristup poluautomatizacije znatno smanjuje vrijeme potrebno za označavanje velikih skupova podataka, istovremeno održavajući visoku razinu točnosti. To je posebno korisno u scenarijima gdje je brzina ključna, kao što su brzi razvojni ciklusi u industriji autonomnih vozila. Deepen AI također nudi snažne mogućnosti timske suradnje, omogućujući višekorisnički pristup i paralelno označavanje podataka. Ovo je idealno za velike timove koji trebaju koordinirati rad na složenim projektima. Platforma podržava različite formate podataka i integracije s popularnim alatima za strojno učenje,

što olakšava prijelaz od označavanja do treniranja modela. Uz to, Deepen AI nudi detaljne analitičke alate koji omogućuju praćenje napretka projekta, procjenu kvalitete oznaka i identifikaciju područja koja zahtijevaju dodatnu pažnju. Platforma je također prilagođena za rad s različitim tipovima podataka, uključujući 2D slike, 3D oblake točaka i videozapise, što je čini vrlo fleksibilnom u različitim aplikacijama računalnog vida. Na primjer, u kontekstu autonomnih vozila, Deepen AI se može koristiti za precizno označavanje voznih traka, prometnih znakova, pješaka i drugih objekata u složenim urbanim okruženjima. Iako Deepen AI nudi brojne prednosti, uključujući automatizaciju i timsku suradnju, njegova složenost i komercijalna priroda mogu biti prepreka za manje timove ili projekte s ograničenim resursima. Međutim, za organizacije koje traže robusno rješenje za označavanje podataka u industrijskim aplikacijama, Deepen AI pruža iznimnu vrijednost kroz svoju učinkovitost i preciznost.

#### **2.4.6 Osvrt na predstavljene alate za označavanje linija vozne trake**

Nakon detaljnog objašnjenja značajki svakog alata, u nastavku je prikazana usporedna tablica koja sumira ključne karakteristike i mogućnosti ovih alata. Tablica 2.3 pruža pregled informacija koje su bitne za donošenje odluka o izboru alata u različitim kontekstima primjene, poput jednostavnosti korištenja, fleksibilnosti, podržanih formata oznaka i mogućnosti integracije s drugim sustavima. VIA je jednostavan alat koji se ističe svojom intuitivnošću i mogućnošću brze ručne anotacije. Njegova jednostavnost dolazi s ograničenjima, poput nemogućnosti podrške za naprednije značajke ili složene projekte, čineći ga prikladnim za manje, brže zadatke. S druge strane, CVAT se nameće kao moćniji alat, s podrškom za raznovrsne formate oznaka, mogućnost suradnje više korisnika te napredne opcije upravljanja projektima. Ova fleksibilnost i opsežnost čine ga idealnim za veće projekte s višestrukim suradnicima. Deepen AI nudi niz naprednih značajki, uključujući automatizirane anotacije i prilagodljive modele, čime značajno ubrzava proces označavanja podataka. Međutim, ove mogućnosti dolaze uz višu cijenu, što može ograničiti njegovu primjenu u projektima s ograničenim proračunom. Labelbox se ističe kao sveobuhvatan alat, koji kombinira upravljanje podacima i projektima s mogućnostima integracije. Njegova funkcionalnost čini ga idealnim za veće timove koji zahtijevaju strukturiranu organizaciju podataka, iako može biti resursno zahtjevan. Nasuprot tome, LabelMe predstavlja jednostavno rješenje otvorenog koda, koje omogućuje brzu implementaciju i osnovnu funkcionalnost. Iako mu nedostaju mogućnosti za kolaborativni rad ili naprednije značajke, njegova otvorenost i fleksibilnost čine ga privlačnim za projekte s ograničenim budžetom ili tehničkim potrebama. Svi ovi alati imaju svoje prednosti i nedostatke, pa odabir odgovarajućeg alata prvenstveno ovisi o specifičnim potrebama projekta. Ključni faktori uključuju složenost zadatka, broj suradnika,

dostupne resurse i zahtjeve za integraciju s postojećim sustavima. U konačnici, temeljita analiza zahtjeva projekta i dostupnih mogućnosti svakog alata omogućit će donošenje informirane odluke koja će optimizirati proces označavanja podataka i osigurati efikasnost u daljnjoj primjeni.

*Tablica 2.3 Zbirni pregled glavnih karakteristika postojećih alata za analizu i vrednovanje metoda za detekciju linija vozne trake*

ZNAČAJKA	LabelMe [26]	VIA [27]	Labelbox [28]	CVAT [29]	Deepen AI [31]
<b>PLATFORMA</b>	Web-baziran, otvorenog koda	Web-baziran, otvorenog koda	Komercijalna, web-bazirana	Web-baziran, otvorenog koda	Komercijalna, web-bazirana
<b>PODRŽANE VRSTE PODATAKA</b>	Slike	Slike, videozapisi	Slike, videozapisi, tekst, 3D podaci	Slike, videozapisi	Slike, videozapisi, 3D točkasti oblaci
<b>VRSTE OZNAKA</b>	Poligoni, linije, točke	Poligoni, pravokutnici, linije, točke	Poligoni, pravokutnici, linije, točke	Poligoni, pravokutnici, linije, točke	Poligoni, pravokutnici, linije, točke
<b>AUTOMATIZACIJA</b>	/	/	Poluautomatizacija pomoću ML	Poluautomatizacija pomoću ML	Napredna automatizacija pomoću ML
<b>PODRŠKA ZA TIMSKI RAD</b>	Ograničena, individualna upotreba	Ograničena, individualna upotreba	Da, s naprednim alatima za suradnju	Da, višekorisnički pristup	Da, višekorisnički pristup
<b>FORMAT IZLAZNIH PODATAKA</b>	JSON	JSON	JSON, COCO, Pascal VOC	XML, JSON, COCO	JSON, COCO, Pascal VOC
<b>TROŠAK</b>	Besplatan, otvorenog koda	Besplatan, otvorenog koda	Komercijalan, različiti planovi cijena	Besplatan, otvorenog koda	Komercijalan, različiti planovi cijena
<b>INTEGRACIJA S OKVIRIMA STROJNOG UČENJA</b>	Ograničena, može se koristiti s TensorFlow i PyTorch putem prilagođenih skripti	Ograničena, korisnici mogu ručno izvoziti podatke za TensorFlow i PyTorch	Da, integracija s TensorFlow, PyTorch, i drugim popularnim okvirima	Da, integracija s TensorFlow, PyTorch, OpenVINO	Da, integracija s TensorFlow, PyTorch, Keras, i drugim popularnim okvirima
<b>API ZA UPRAVLJANJE PODACIMA</b>	/	/	Da	Da	Da

### **3. IZRADA APLIKACIJE ZA OZNAČAVANJE LINIJA VOZNE TRAKE**

U potpoglavlju 2.4 opisani su postojeći alati za označavanje linija vozne trake. Međutim, postoje situacije u kojima nijedan alat ne nudi potrebnu funkcionalnost, što prisiljava istraživače da kreiraju nove alate ili modificiraju postojeće [32]. Iako se nekada preporučuje modifikacija postojećih alata kako bi se izbjegli problemi poput visokih troškova održavanja i rizičnih dizajnerskih grešaka, često je zbog loše dokumentacije, nedostatka podrške i softverskih problema jednostavnije i brže razviti novi alat s potrebnim specijaliziranim značajkama. U ovom poglavlju opisan je proces izrade nove jednostavne aplikacije za ručno označavanje linija vozne trake koja će se koristiti za prikupljanje i pripremu podataka potrebnih za treniranje modela strojnog učenja za detekciju linija vozne trake. Aplikacija je u sklopu ovog diplomskog rada osmišljena s ciljem olakšavanja i ubrzanja postupka označavanja, što je ključno za razvoj sustava autonomne vožnje i napredne analize prometnih podataka.

#### **3.1 Zahtjevi na aplikaciju za označavanje linija voznih traka**

Prije samog početka razvoja aplikacije ključno je definirati i razumjeti sve zahtjeve aplikacije. Ovi zahtjevi predstavljaju osnovu na kojoj će se graditi cijeli projekt i osiguravaju da konačni proizvod ispunjava potrebe korisnika te da je funkcionalan i pouzdan. Zahtjevi sustava mogu se podijeliti u dvije glavne kategorije: funkcionalni i nefunkcionalni zahtjevi. Nefunkcionalni zahtjevi obuhvaćaju aspekte poput performansi, pouzdanosti, skalabilnosti i interoperabilnosti aplikacije. Funkcionalni zahtjevi, s druge strane, definiraju konkretne funkcije koje aplikacija mora obavljati. Neki od tih zahtjeva su:

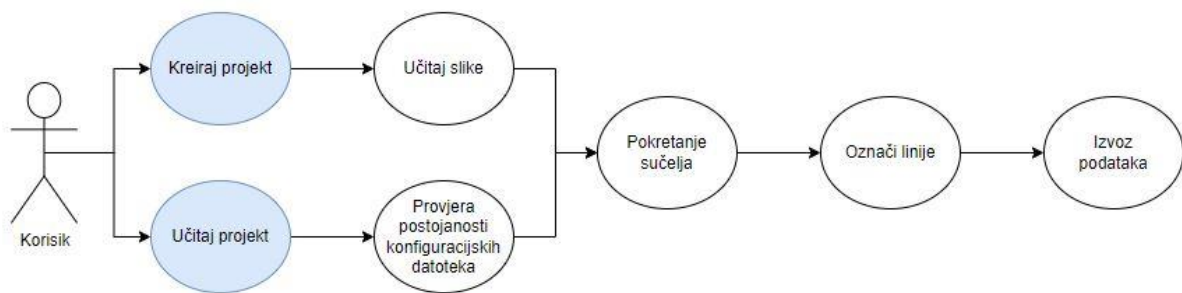
1. Kreiranje i upravljanje projektima - osnovni zahtjev za aplikaciju je omogućiti korisnicima intuitivno kreiranje i efikasno upravljanje projektima. Korisnik bi trebao imati mogućnost kreiranja različitih projekata koji su prilagođeni specifičnim potrebama i zahtjevima. Ova funkcionalnost uključuje definiranje naziva projekta, odabir odgovarajuće lokacije na disku za pohranu podataka te mogućnost odabira relevantne galerije slika koja će biti korištena unutar projekta. Nakon inicijalnog kreiranja projekta, aplikacija treba automatski generirati odgovarajuću mapu s nazivom projekta na odabranoj lokaciji na disku. Ova mapa treba sadržavati odabranu galeriju slika i pridruženu konfiguracijsku datoteku koja će pohranjivati sve ključne parametre vezane za projekt. Dodatno, aplikacija bi trebala omogućiti korisnicima pregled i upravljanje postojećim projektima. To uključuje

možnost učitavanja, uređivanja i brisanja projekata. Ova funkcionalnost osigurava visoku fleksibilnost i prilagodljivost aplikacije potrebama korisnika.

2. Rukovanje galerijom slika - aplikacija mora pružiti korisnicima intuitivno i učinkovito rukovanje galerijom slika, čime se omogućava jednostavan pregled i manipulacija slikama. Ključni aspekti rukovanja galerijom uključuju pregledavanje, pomicanje i dodavanje slika. Pregledavanje slika treba biti glatko i korisnicima omogućiti lak prelazak s jedne slike na drugu. Korisničko sučelje mora biti dizajnirano tako da navigacija kroz slike bude jednostavna, bilo pomoću navigacijskih tipki ili gesti. Aplikacija mora podržavati jednostavno dodavanje novih slika u galeriju. Korisnici bi trebali moći učitati slike s lokalnog diska. Ova funkcionalnost omogućava korisnicima da brzo prošire svoju kolekciju slika i osigurava da su svi relevantni materijali lako dostupni za označavanje.
3. Ručno označavanje - glavna funkcionalnost aplikacije je omogućiti korisnicima ručno označavanje linija vozne trake na učitanim slikama. Korisnici trebaju imati pristup alatu za crtanje linija koji omogućava precizno označavanje, prilagođavajući se točno linijama vozne trake. Proces ručnog označavanja zasnovan je na označavanju točaka duž linije vozne trake, a aplikacija ih automatski povezuje u kontinuiranu liniju. Linije koje korisnici označavaju trebaju biti lagano transparentne. Ova transparentnost omogućava korisnicima jasan pregled nad označenim dijelovima slike, olakšavajući precizno pozicioniranje linija bez ometanja vizualnog prikaza same slike. Ovaj alat mora podržavati crtanje više linija na istoj slici, omogućujući označavanje svih traka na cesti, uključujući središnje i rubne linije. Alat mora podržati minimalno pet linija dok maksimalan broj nije određen. Aplikacija mora omogućiti jednostavno uklanjanje postojećih linija i njihovo ponovno crtanje bez gubitka ostalih podataka na slici, osiguravajući tako visoku razinu preciznosti i fleksibilnosti u procesu označavanja.
4. Spremanje označenih podataka - spremanje označenih podataka osigurava da svi korisnički unosi budu pohranjeni na način koji omogućava daljnju analizu i korištenje za treniranje modela strojnog učenja. Aplikacija mora omogućiti pohranu svih označenih podataka u strukturiranom formatu koji je pogodan za daljnju obradu. Podržani formati uključuju JSON, XML, i druge standardne formate koji se koriste u području računalnog vida i strojnog učenja. Ovi formati omogućavaju jednostavno prebacivanje podataka između različitih alata i platformi za analizu.

### 3.2 Tehnički detalji izrade aplikacije za ručno označavanje linija vozne trake

Prva faza izrade aplikacije je definiranje zahtjeva, što je opisano u podpoglavlju 3.1, a sljedeći korak u izradi aplikacije je kreiranje dijagrama slučaja korištenja (*engl.* Use-Case). Dijagrama slučaja korištenja su važan alat u softverskom inženjeringu koji omogućuju vizualizaciju interakcija između korisnika i sustava, pomažući u boljem razumijevanju funkcionalnosti koje sustav treba pružiti. Na slici 3.1 prikazan je dijagrama slučaja korištenja za aplikaciju. Dijagram započinje s mogućnošću da korisnik izabere želi li kreirati novi projekt ili otvoriti već postojeći. Pri kreiranju novog projekta, korisnik određuje lokaciju projekta na disku i daje mu naziv. U idućem koraku korisnik učitava slike koje želi označiti, nakon čega može započeti s radom na projektu. Nakon što korisnik označi željene podatke on ih izvozi za daljnju upotrebu.

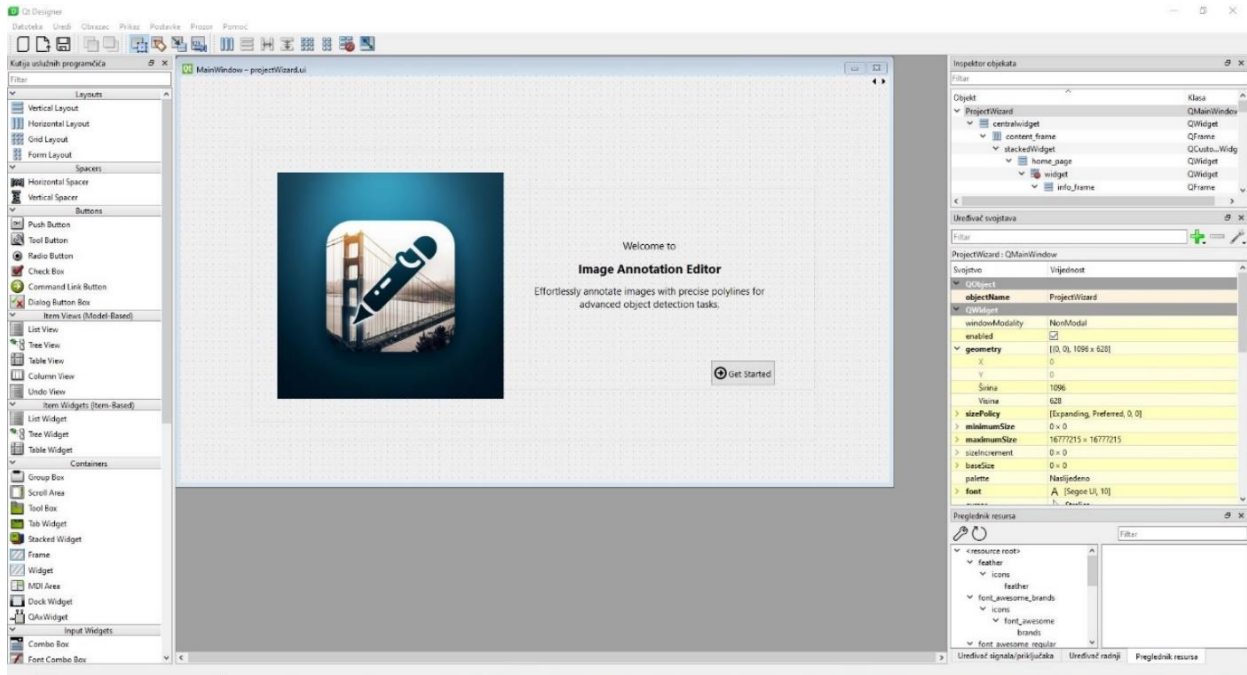


Slika 3.1 Dijagrama slučaja korištenja aplikacije za ručno označavanje linija vozne trake

Kako bi se ostvarili ciljevi projekta za izradu aplikaciju za ručno označavanje linija vozne trake, odabran je programski jezik *Python* u kombinaciji s *Qt* aplikacijskim okvirom. *Python*, verzija 3.9.13, je odabran zbog svoje jednostavnosti, čitljivosti i široke podrške za različite biblioteke. *Python*-ova fleksibilnost i bogat ekosustav omogućuju brzo prototipiranje i razvoj, a *PyCharm* kao razvojno okruženje dodatno poboljšava produktivnost zahvaljujući svojim naprednim alatima za ispravljanje pogrešaka (*engl.* Debugging) i integriranim podrškama za verzioniranje koda.

*Qt* je aplikacijski okvir otvorenog koda koji se koristi za razvoj grafičkih korisničkih sučelja i višeplatformskih aplikacija [33]. Razvijen na *C++* programskom jeziku, *Qt* omogućuje stvaranje aplikacija visokih performansi koje mogu raditi na različitim operacijskim sustavima, uključujući i različite mobilne platforme kao što su Android i iOS. *Qt Designer*, čije sučelje je prikazano na slici 3.2, dio je *Qt* alata koji omogućuje dizajniranje korisničkih sučelja putem intuitivnog sučelja za povlačenje i ispuštanje [34]. Traka s lijeve strane ekrana prikazuje elemente sučelja (*engl.*

Widgets) za koje korisnik može koristiti metodu povuci-i-ispusti (engl. Drag and Drop) na glavni prozor prikazan na sred ekrana. S desne strane ekrana nalazi se preglednik objekata (engl. Object Inspector) koji korisniku omogućuje pregled hijerarhije objekata te manipulaciju objektima. *Qt Designer* značajno olakšava proces izrade aplikacije jer programeri mogu vizualno kreirati sučelje aplikacije bez potrebe za pisanjem puno koda, što olakšava održavanje i iteracije dizajna.



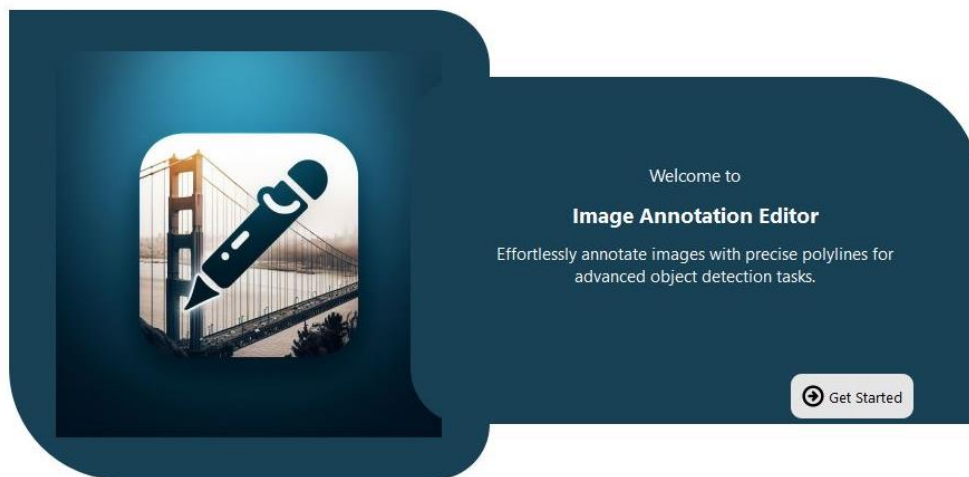
Slika 3.2 Sučelje *Qt Designer* grafičkog alata

Razumijevanje korištenih alata omogućuje bolju primjenu njihovih funkcionalnosti, što je ključno za uspješnu implementaciju aplikacije. U skladu sa zahtjevima ove aplikacije implementirano je korisničko sučelje koje je prikazano na slici 3.3. Kod pokretanja aplikacije, korisnika prvo dočekuje početno sučelje s gumbom "Get Started". Prikazano sučelje se sastoji od idućih *Qt* komponenti [34]:

- *QMainWindow* - osnovni element glavnog prozora aplikacije. Omogućava standardnu strukturu prozora s izbornikom, alatnim trakama, statusnom trakom i središnjim područjem za prikaz sadržaja. *QMainWindow* pruža okvir za organizaciju ostalih komponenti sučelja. U ovoj aplikaciji služi kao glavni kontejner za sve ostale elemente sučelja.
- *QFrame* - element sučelja koji se koristi za stvaranje okvira oko drugih grafičkih elemenata. Može se koristiti za grupiranje elemenata zajedno, pružajući vizualnu strukturu i organizaciju.

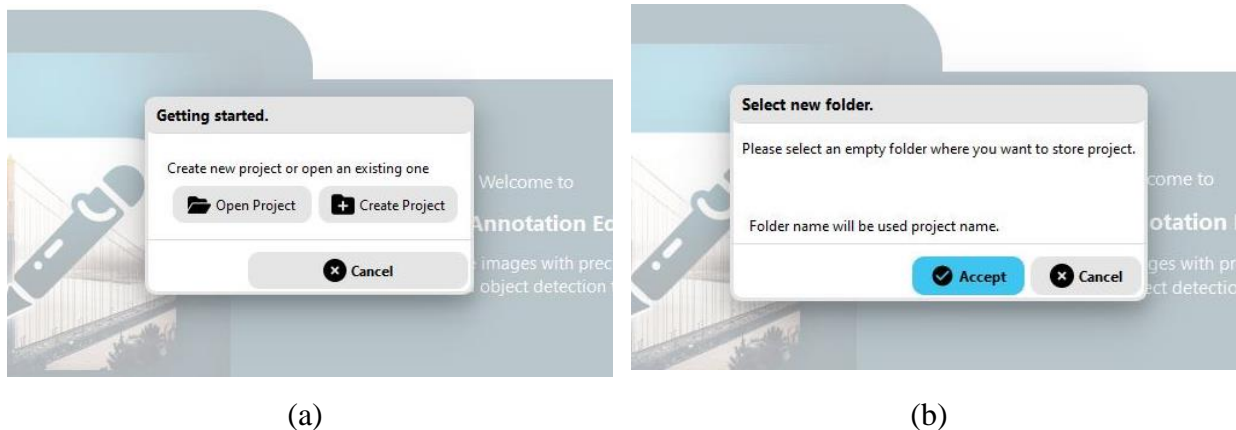


- *QWidget* - osnovni gradivni blok za sve grafičke objekte u *Qt*-u. Sve grafičke komponente su izvedene iz *QWidget* klase. Pruža osnovne funkcionalnosti za crtanje i rukovanje događajima. *QWidget* se koristi kao osnovna klasa za izradu svih prilagođenih elemenata sučelja i može se koristiti za stvaranje jednostavnih komponenti sučelja.
- *QLabel* - koristi se za prikazivanje statičkog teksta ili slike. Obično se koristi za dodavanje opisa, naslova ili drugih informativnih elemenata u sučelje.
- *QPushButton* - predstavlja gumb koji korisnik može pritisnuti za izvršavanje određene akcije. Može imati tekstualnu oznaku, ikonu ili oboje. Koristi se za aktiviranje različitih funkcionalnosti unutar aplikacije, kao što su učitavanje slika, spremanje označenih podataka ili brisanje oznaka.



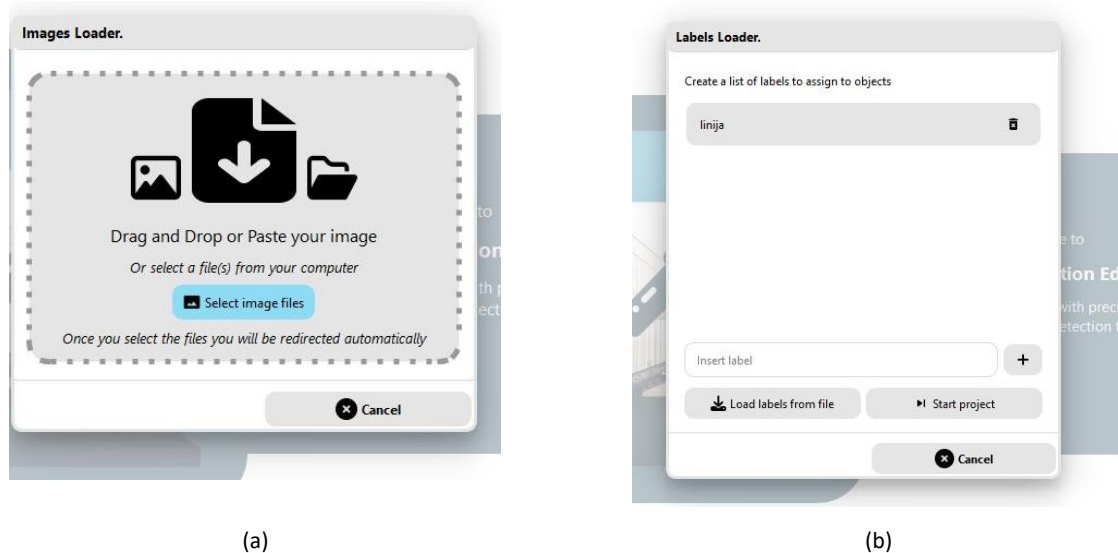
*Slika 3.3 Dizajn početnog sučelja aplikacije za označavanje linija vozne trake*

Nakon pritiska na gumb "*Get Started*" aplikacija korisnika preusmjerava na konfiguraciju projektnih parametara. Na slici 3.4 pod (a) je prikazan skočni prozor s dvjema opcijama: "*Open Project*" i "*Create Project*", koji korisniku daje mogućnost odabira između otvaranja već postojećeg projekta ili kreiranja novog. Nakon odabira kreiranja novog projekta, na slici 3.4 pod (b) je prikazan drugi skočni prozor u kojem se od korisnika traži da odabere praznu mapu gdje će projekt biti spremljen. Korisniku se također objašnjava da će naziv mape biti korišten kao naziv projekta. Za izradu skočnih prozora koristili su se elementi sučelja *QFrame*, *QLabel* i *QPushButton*. Nakon što korisnik odabere mapu u kojoj će se projekt nalaziti, pojavljuje se novi skočni prozor, prikazan na slici 3.5 a), koji omogućuje korisniku da odabere slike koje želi učitati iz mape ili ih doda metodom povuci-i-ispusti.



Slika 3.4 (a) prozor za odabir projekta (b) prozor za odabir mape

Nakon učitavanja slika, pojavljuje se još jedan skočni prozor, prikazan na slici 3.5 b), koji korisniku omogućuje kreiranje oznaka. U ovom dijelu mogu se kreirati nazivi za vrste linija ili pozicije koje se označavaju te će to biti zapisano u datoteku u JSON formatu.

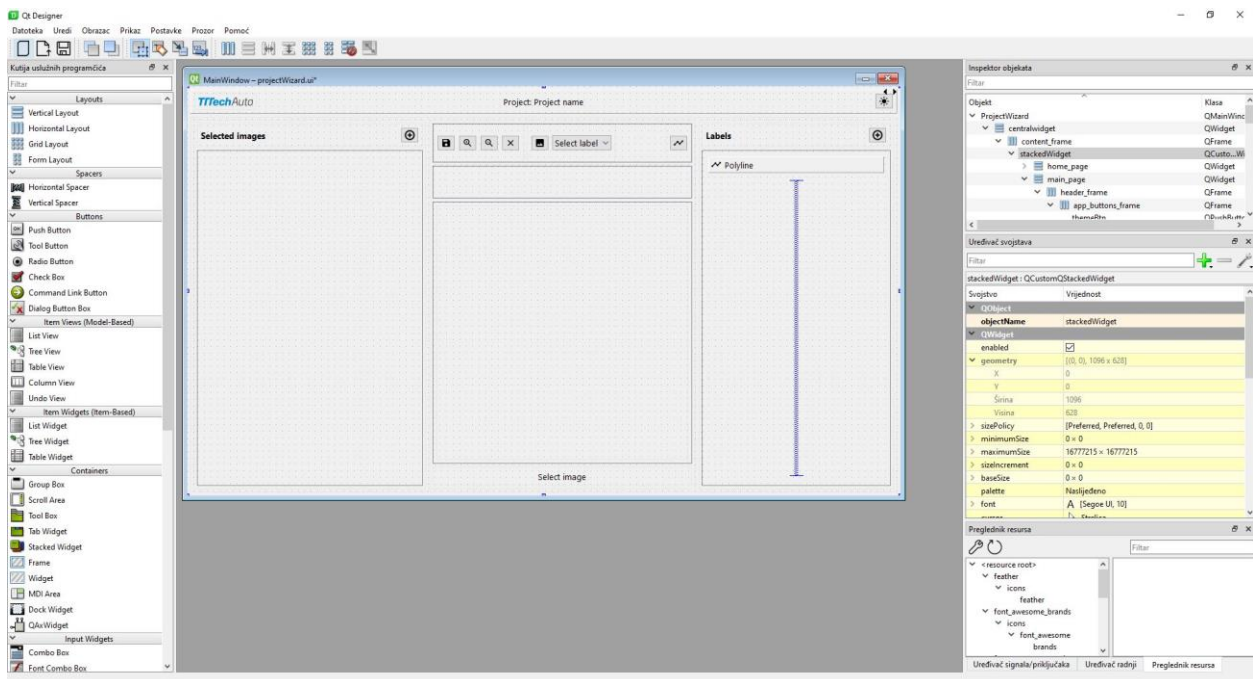


Slika 3.5 (a) skočni prozor za odabir galerije slika (b) skočni prozor za kreiranje oznaka

Za izradu ovih dvaju skočnih prozora koristili su se prethodno opisani elementi sučelja *QFrame*, *QPushButton*, *QLabel*, *QWidget*. Dodatno su se koristili još [34]:

- *QLineEdit* - korisnicima aplikacije omogućava unos jednostrukog teksta. Koristi se za kreiranje tekstualnih polja u aplikacijama gdje korisnici mogu unijeti ili urediti tekstualni sadržaj.
- *QScrollArea* - omogućava prikazivanje sadržaja koji premašuje vidljivi prostor. Omogućava horizontalno i vertikalno pomicanje sadržaja koji je prevelik da bi se prikazao u zadanom području.

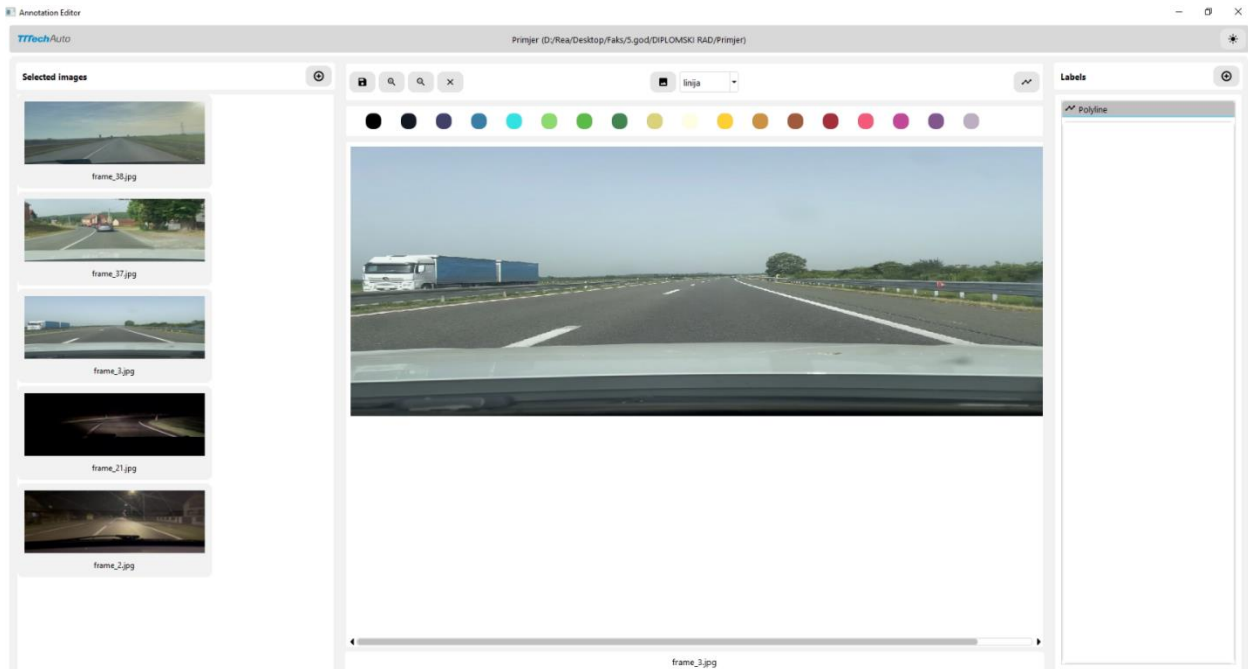
Sučelje za označavanje objekata korisniku omogućava prikaz slika iz projektne galerije, dodavanje novih slika, odabir opisa linije, kreiranje i brisanje kreiranih oznaka, te izvoz datoteka s oznakama na disk. Na slici 3.6 je prikazano sučelje za označavanje slika unutar *QDesigner* grafičkog alata. Elementi sučelja korišteni za izradu ovog sučelja su: *QFrame*, *QPushButton*, *QLabel*, *QWidget*, *QScrollArea*. Dodatni elementi sučelja koji su se još koristili su *QToolBox* i *Spacer* vidljivi na desnoj strani sučelja za označavanje linije na slici 3.6. *QToolBox* je komponenta korisničkog sučelja u *Qt*-u koja omogućuje organizaciju sadržaja u više kartica koje se mogu proširiti ili sažeti, slično kao sklopivi paneli. *Spacer* (razmaknica) se koristi za stvaranje prostora između drugih komponenti unutar korisničkog sučelja, omogućujući dinamično prilagođavanje rasporeda elemenata prilikom promjene veličine prozora [34].



Slika 3.6 Dizajn sučelja unutar *QDesigner* grafičkog alata

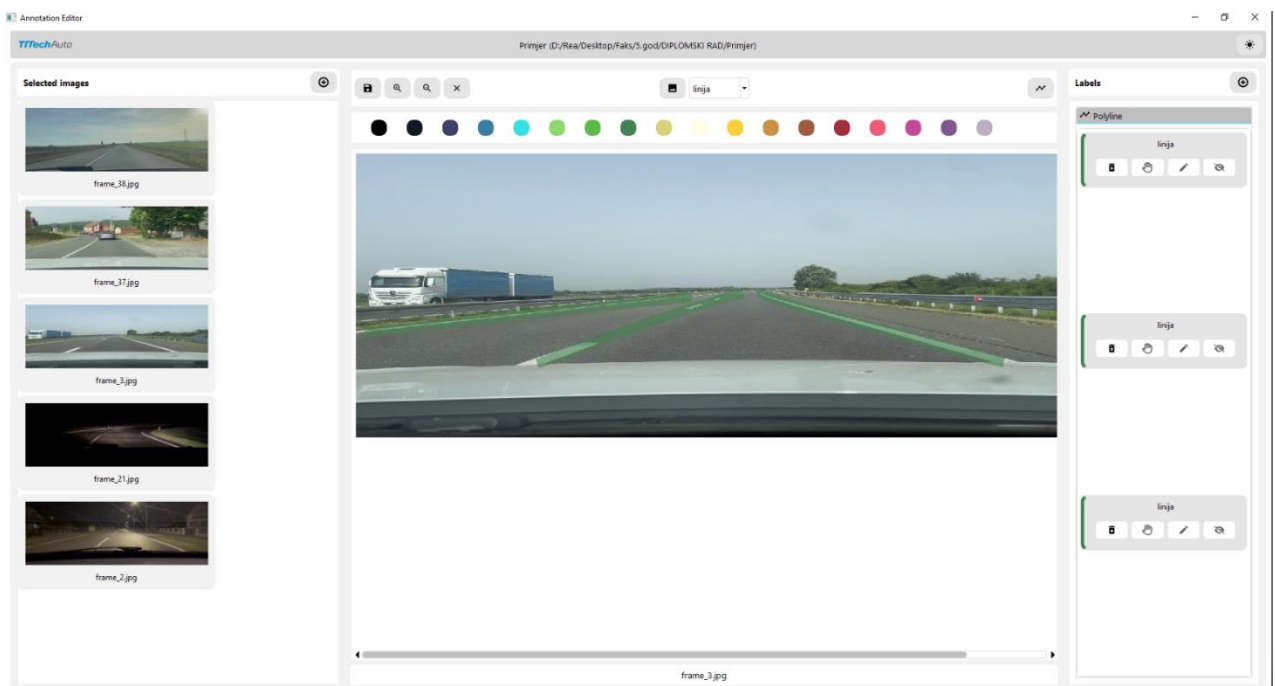
Razlike između prikaza korisničkog sučelja u *Qt Designeru* i stvarnog prikaza prilikom pokretanja aplikacije mogu biti posljedica nekoliko faktora. *Qt Designer* prikazuje statički izgled i osnovni raspored elemenata bez dinamičkih promjena koje se događaju tijekom izvršavanja aplikacije. Tijekom pokretanja aplikacije mogu se primijeniti dodatne stilizacije i prilagodbe putem koda koje nisu vidljive u statičkom prikazu u *Qt Designeru*. Slika 3.7 prikazuje sučelje za označavanje slika nakon pokretanja aplikacije. Razlog za razliku u prikazu leži u činjenici da su, prilikom razvoja aplikacije, nakon pretvorbe UI (*engl.* User Interface) dokumenta, koji u ovom slučaju predstavlja XML datoteku generiranu u *QDesigneru* i koja definira izgled i strukturu

korisničkog sučelja, u Python kod, dodane razne stilizacije i prilagodbe koje omogućuju dinamično prilagođavanje korisničkog sučelja.



*Slika 3.7 Primjer korisničkog sučelja s učitanim slikom*

S lijeve strane sučelja (slika 3.8) prikazana je galerija projektnih slika. Prilikom pritiska na gumb „+“ korisniku se otvara skočni prozor prikazan na slici 3.5 a), što omogućava proširenje projektne galerije.

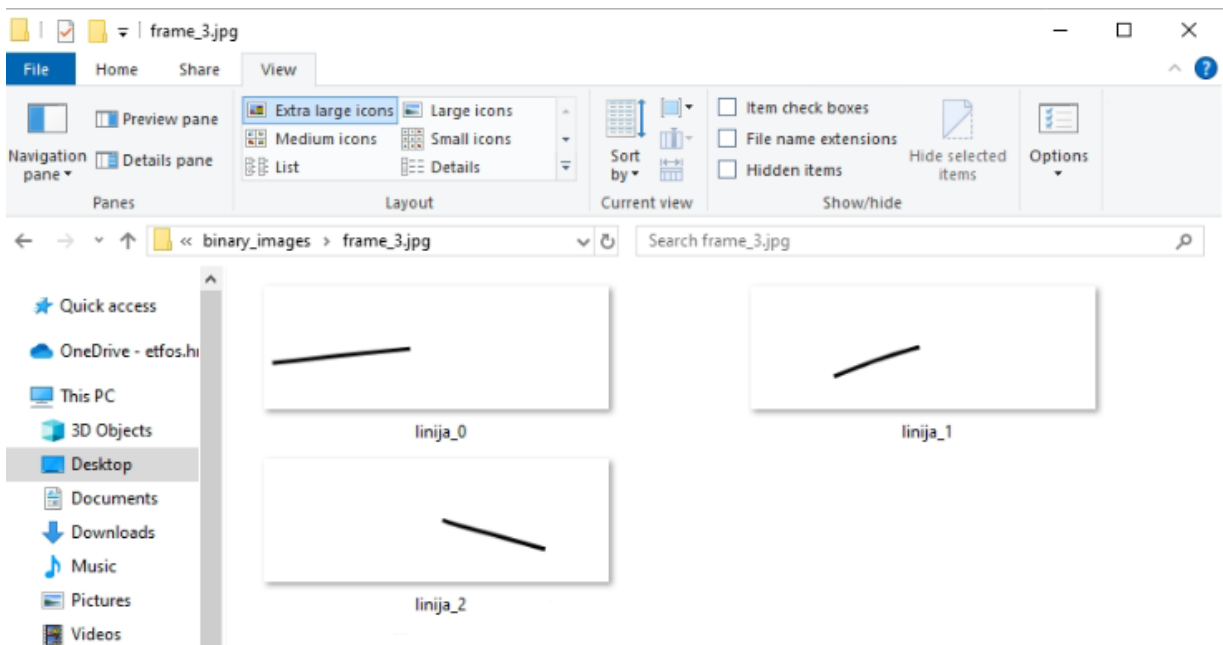
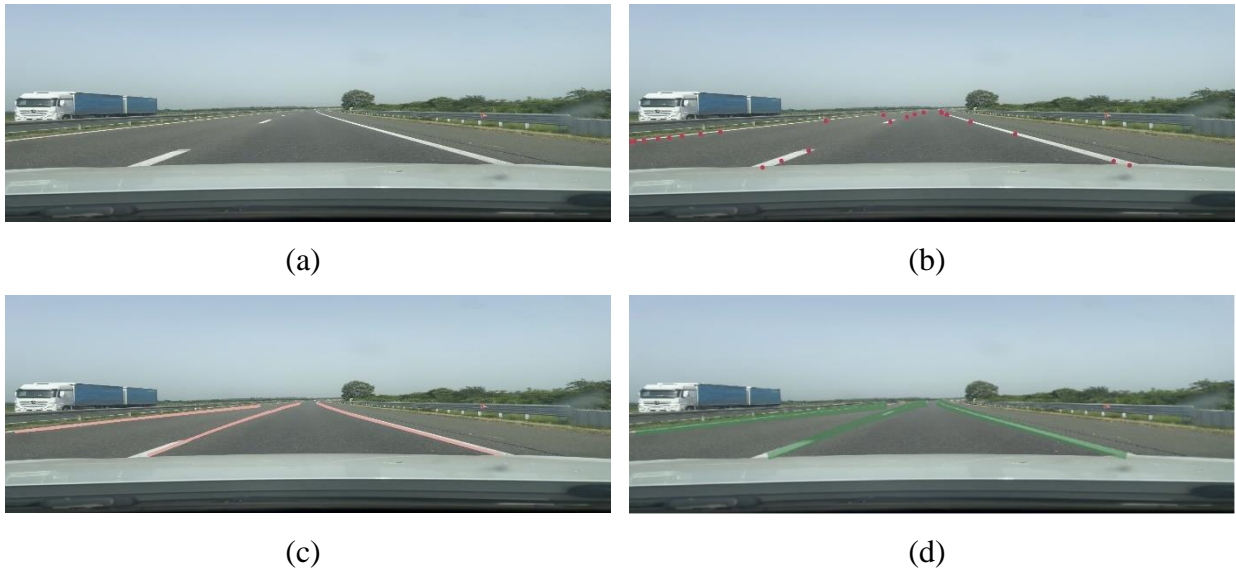


*Slika 3.8 Primjer prikaza linija nakon označavanja slike*

Nasred sučelja je prikazana slika koju korisnik trenutno označava, dok se iznad slike nalaze opcije za odabir boje linije i izbor naziva. Također se nalaze opcije za uvećavanje i smanjivanje rezolucije prikazane slike kao i mogućnost za izvoz datoteka s oznakama na disk. S desne strane nalaze se označene linije koje se mogu izbrisati. Korištenje aplikacije je intuitivno i prilagođeno korisniku. Desnim klikom na željenu sliku iz projektne galerije, koja se nalazi s lijeve strane sučelja, korisnik dobiva mogućnost odabira između brisanja slike iz galerije ili postavljanja slike na poziciju za označavanje. Označavanje kreće tako da korisnik krene s označavanjem točaka unutar slike koju želi označiti. Na slici 3.9 prikazan je proces označavanja, pod (a) nalazi se izvorna slika dok se pod (b) nalaze označene točke. Za kreiranje linija koristi se funkcija *polyline*. Ova funkcija omogućava korisniku crtanje linije definiranjem niza točaka. *Polyline*, ili višelinijski poligon, je geometrijski entitet koji se sastoji od niza povezanih ravnih segmenata. Korisnik dodaje točke na željene pozicije, pri čemu se svaka dodana točka automatski povezuje s prethodnom točkom putem ravnog segmenta. Desnim klikom miša završava se crtanje *polyline*-a, definirajući konačni oblik linije što je prikazano na slici 3.9 (c). Funkcija *polyline* ne uključuje zakrivljene segmente; svi segmenti su ravni i povezani u točkama, što ne omogućava najvjerniji prikaz linija na cestama, osobito u situacijama gdje postoje krivine.

Nakon što korisnik završi crtanje linije, aplikacija pomoću *cubic spline* algoritma prilagođava kreiranu liniju onoj na slici. *Cubic spline* je matematička funkcija koja se koristi za interpolaciju i aproksimaciju podataka [35]. Ova funkcija koristi niz zadanih točaka za stvaranje glatke krivulje koja prolazi kroz sve te točke, oblikujući se po krivuljama na cesti što je prikazano na slici 3.9 (d). Svaki segment *spline*-a je polinom trećeg stupnja, a susjedni segmenti dijele zajedničke točke što rezultira glatkim prijelazima. Jedan od najvažnijih uvjeta *cubic spline* funkcije je da sekvenca x koordinata mora biti strogo uzlazna, zbog čega aplikacija sadrži testiranje uzlaznosti niza i sortiranje u slučaju da niz ne zadovoljava uvijete.

Nakon kreiranja oznaka na slici, korisnik može spremi svoje podatke pritiskom na gumb s ikonom diska koji se nalazi iznad slike. Ova radnja izvozi datoteku s oznakama na lokaciju projektne mape. Važno je napomenuti da, ukoliko korisnik prijeđe na sljedeću sliku bez prethodnog spremanja podataka, te oznake neće biti spremljene i bit će izgubljene. Prilikom pritiska na gumb za izvoz podataka, aplikacija automatski generira strukturu mapa u projektnoj mapi, što je prikazano na slici 3.10. U okviru ovih mapa, svaka slika ima svoju zasebnu mapu koja sadrži datoteke s imenom slike i popratnim informacijama. Na taj način, svi podaci su organizirani i lako dostupni unutar projektne mape, osiguravajući da korisnik može jednostavno upravljati i pristupati svojim oznakama.



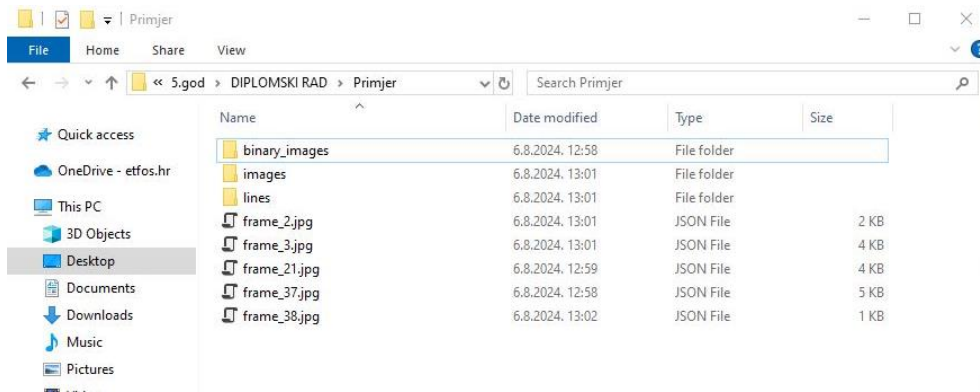
(e)

*Slika 3.9 Proces označavanja linija voznih traka kroz različite faze obrade: (a) Originalna slika ceste bez oznaka, koja predstavlja početni ulazni podatak (b) korisnik ručno označava ključne točke na slici (c) Prikaz linija voznih traka kao poligonalnih linija (d) Prikaz linija voznih traka izglađenjenih pomoću cubic spline interpolacije (e) prikaz generiranih binarnih slika za pojedinačnu označenu liniju*

Binarne slike su digitalne slike koje se sastoje samo od dviju boja, najčešće crne i bijele. Svaki piksel u binarnoj slici može imati jednu od dvije vrijednosti, što pojednostavljuje vizualne informacije i omogućuje lakšu obradu. Na slici 3.9 (e) prikazana je mapa binarnih slika za sliku prikazanu pod (a). Mapa se sastoji od triju binarnih slika jer na slici postoje tri linije koje su označene unutar aplikacije. U računalnom vidu, binarne slike su ključne jer smanjuju računalne

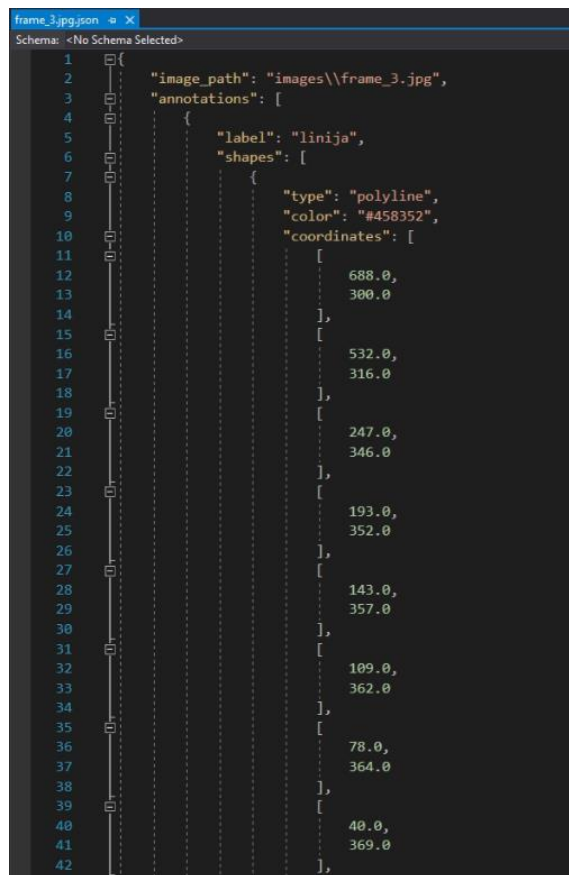


zahtjeve, omogućujući bržu i efikasniju analizu podataka. Njihova jednostavnost poboljšava robusnost algoritama za prepoznavanje uzoraka, uklanjajući varijacije u boji i osvjetljenju.



Slika 3.10 Primjer mape projekta nakon označavanja

JSON je jednostavan format za razmjenu podataka, poznat po svojoj čitljivosti i jednostavnosti uporabe. Funkcionira tako da strukturira podatke u obliku parova ključ-vrijednost, omogućujući lako čuvanje i prijenos informacija između servera i web aplikacija. Na slici 3.11 prikazano je kako se JSON koristi u sklopu aplikacije na slici „frame\_3.jpg“ koja je prikazana na slici 3.9 (a).



Slika 3.11 Primjer JSON datoteke za označenu sliku

Ključ „*image\_path*“ označava putanju do slike i osiguravaju da se svaka slika može ispravno učitati. Ključ „*annotations*“ sadrži druge ključ-vrijednost parove koji specificiraju podatke o oznakama kao što su naziv linije, boja, te parovi koordinata. Koordinate unutar JSON formata označavaju svaki klik miša na ekranu prilikom označavanja. Ovaj dokument služi kao sigurnosna kopija u slučaju problema s aplikacijom ili potrebe za nadogradnjom, čime se osigurava da se slike ne moraju ponovo označavati.

Tijekom izrade aplikacije, određeni postupci i odluke morali su biti doneseni s obzirom na zahtjeve drugog dijela ovog rada koji se bavi usporedbom modela zasnovanih na strojnom učenju. Konkretno, prilikom strukturiranja i implementacije aplikacije za ručno označavanje linija vozne trake, radi pokretanja jedne od prethodno definiranih mreža, SCNN [7], koja će se predstaviti u poglavlju 4.2, morala je biti kreirana tekstualna datoteka prikazana na slici 3.12. Datoteka je kreirana uz pomoć *cubic spline* interpolacije koja kreira jednadžbu linije. U rasponu od najmanjeg do najvećeg broja na ordinati slike uzima se svaka deseta y koordinata te se uz pomoć *cubic spline* interpolacije računa odgovarajući x na osi apscisa. Na slici 3.12 prikazane su tri linije koordinata koje predstavljaju tri linije na cesti. Tekstualna datoteka je strukturirana tako da koordinate slijede redoslijed x y x y ..., pri čemu svaki par koordinata odgovara jednoj točki na liniji. Na primjeru slike 3.12 koji predstavlja „*frame\_3.jpg*“ prikazanu na slici 3.9 pod (a), prvi par koordinata za prvu liniju je (421, 270), dok je drugi par koordinata (477, 280), i tako dalje.



```
frame_3.jpg.lines - Notepad
File Edit Format View Help
688 300 591 310 491 320 392 330 299 340 212 350 127 360 28 370
797 293 757 303 727 313 693 323 661 333 631 343 602 353 574 363 546 373 520 383 493 393 467 403 441 413 414 423 387 433
845 292 870 302 901 312 933 322 969 332 1006 342 1044 352 1080 362 1117 372 1153 382 1189 392 1225 402 1262 412 1299 422 1338 432
```

Slika 3.12 Primjer tekstualne datoteke za „*frame\_3.jpg*“

U ovom podpoglavljju detaljno su opisane funkcionalnosti i rad aplikacije za ručno označavanje linija vozne trake. Kroz prikaz ključnih elemenata korisničkog sučelja te objašnjenje načina na koji se bilježe i obrađuju podaci, prikazano je kako aplikacija omogućuje precizno i efikasno označavanje i pohranu informacija. Posebna pažnja posvećena je korištenju naprednih algoritama poput *cubic spline* interpolacije za prilagodbu linija, čime se osigurava visoka razina točnosti i kvalitete rezultata. Opisan je i postupak generiranja dodatnih datoteka koje sadrže koordinate označenih točaka i binarne slike, što je ključno za daljnju analizu i primjenu unutar modela zasnovanih na strojnom učenju. U sljedećem podpoglavljju bit će detaljnije opisan proces kreiranja podatkovnog skupa.



## 4. ANALIZA PERFORMANSI METODA ZA DETEKCIJU LINIJA VOZNE TRAKE ZASNOVANIH NA DUBOKOM UČENJU

U ovom poglavlju detaljno se analiziraju performanse metoda za detekciju linija vozne trake koje su zasnovane na dubokom učenju. Cjelokupni zadatak diplomskog rada obuhvaća razvoj aplikacije za označavanje linija vozne trake, kreiranje vlastitog skupa podataka i usporedbu različitih modela zasnovanih na strojnome učenju. Pritom se analizira učinkovitost algoritama treniranih na postojećim skupovima podataka te njihovo prilagođavanje vlastitom skupu podataka kroz proces dodatnog treniranja (dotreniranja). Cilj ovog rada jest procijeniti kako odabrani modeli reagiraju na stvarne cestovne scenarije i osigurati prilagodljivost modela za stvarne uvjete vožnje. Ovo poglavlje sadrži tri glavna dijela: testiranje aplikacije za označavanje linija vozne trake, izradu vlastitog skupa podataka i evaluaciju modela treniranih na različitim skupovima podataka. Za analizu ispravnosti rada alata za označavanje linija opisanog u prethodnom poglavlju odabran je skup od 100 slika, kako bi se potvrdila točnost i konzistentnost označavanja. Testiranje uključuje ručno označavanje slika i provjeru ispunjavanja svih postavljenih zahtjeva kako bi se osiguralo da oznake odgovaraju stvarnim linijama vozne trake. Nakon provjere ispravnosti označavanja, izrađen je vlastiti skup podataka koji sadrži slike voznih traka prikupljene u različitim uvjetima vožnje. Ovaj skup podataka koristi se za testiranje modela treniranih na postojećim skupovima podataka te za dotreniranje kako bi se osigurala njihova prilagodba specifičnim uvjetima koji se javljaju na lokalnim cestama.

Za evaluaciju su odabrane SCNN mreže s različitim arhitekturama, odnosno različitim temeljima mreža (*backbone*): VGG16, ResNet34 i ERFNet. Arhitektura VGG16 (*engl.* Visual Geometry Group) sastoji se od 13 konvolucijskih slojeva i 3 potpuno povezana sloja, što ukupno čini 16 slojeva [36]. Svaki konvolucijski sloj koristi filtre veličine 3x3, omogućujući preciznu ekstrakciju značajki. Rezidualna mreža ResNet34 (*engl.* Residual Network) sastoji se od 34 sloja, pri čemu su ključne komponente rezidualni blokovi. Ovi blokovi omogućuju prijenos podataka kroz mrežu bez degradacije performansi, čime se omogućuje dublje treniranje i bolja stabilnost. ERFNet (*engl.* Efficient Residual Factorized Network) je dizajnirana za segmentaciju slike, kombinirajući efikasnost i visoku preciznost. Arhitektura koristi rezidualne blokove koji su optimizirani za smanjenje računalne složenosti, čime se postiže brzina obrade prikladna za stvarnovremenske aplikacije. Ova mreža je često korištena kada je potrebno postići balans između točnosti i performansi na uređajima s ograničenim resursima.

Svaki od ovih modela treniran je na tri različita podatkovna skupa: CULane, TuSimple i LLAMAS. Ovi skupovi detaljnije su objašnjeni u podpoglavlju 2.3. CULane je opsežan skup podataka koja sadrži različite scenarije vožnje u urbanim i prigradskim uvjetima. TuSimple se specijalizira za jednostavnije scenarije vožnje na autocestama, s ciljem precizne detekcije linija traka. LLAMAS nudi visoku razinu detalja s raznolikim uvjetima osvjetljenja i cestovnih konfiguracija.

U ovom radu provedena su tri različita pristupa kako bi se ocijenila učinkovitost modela na novom skupu podataka. Prvo, modeli trenirani na trima postojećim bazama podataka testirani su izravno na testnom skupu vlastitog skupa podataka, bez ikakvog dodatnog treniranja, kako bi se procijenila njihova sposobnost generalizacije. Drugo, modeli trenirani na postojećim bazama podataka dodatno su trenirani na trening skupu vlastitog skupa podataka, a potom testirani na istom skupu kao i u prethodnom slučaju. Treće, svi modeli su trenirani od početka koristeći isključivo trening skup vlastitog skupa podataka, bez oslanjanja na prethodno treniranje na drugim bazama, kako bi se ispitao njihov kapacitet učenja isključivo iz lokalnih podataka. Potom su i tako trenirani modeli testirani na testnom skupu vlastitog skupa podataka.

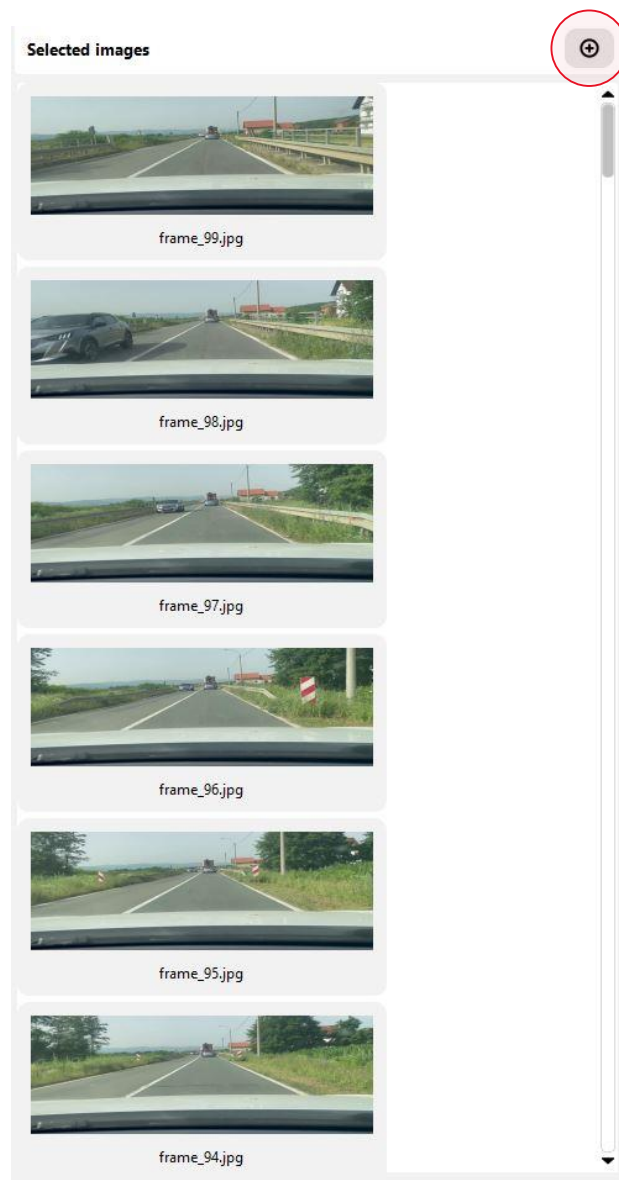
#### **4.1 Testiranje alata za označavanje linija vozne trake**

Kako bi se osigurala pouzdanost razvijene aplikacije za označavanje linija vozne trake potrebno je provesti testiranje njezine funkcionalnosti. Testiranje je provedeno na uzorku od nasumično odabranih 100 slika. Slike su ručno označene unutar aplikacije, a zatim je svaka oznaka provjerena kako bi se potvrdilo da su sve linije točno postavljene. Tijekom testiranja, evaluacija se provodila prema zahtjevima navedenim u potpoglavlju 3.1.

Prvi zahtjev, kreiranje i upravljanje projektima, potvrđen je kroz inicijalno kreiranje i organizaciju projekta u kojem su slike bile pohranjene. Ova funkcionalnost omogućila je intuitivno upravljanje projektom, uključujući kreiranje mape na odabranoj lokaciji i spremanje relevantne galerije slika. Ova funkcionalnost je prikazana u podpoglavlju 3.2. Na slici 3.4 (a) prikazan je skočni prozor koji korisnicima omogućava odabir postojećeg projekta ili kreiranje novog. Skočni prozor na slici 3.4 (b) prikazuje obavijest koja se pojavljuje kada se izabere opcija za kreiranje novog projekta, uz napomenu da je potrebno otvoriti prazan folder gdje će se smjestiti sve datoteke vezane uz projekt.

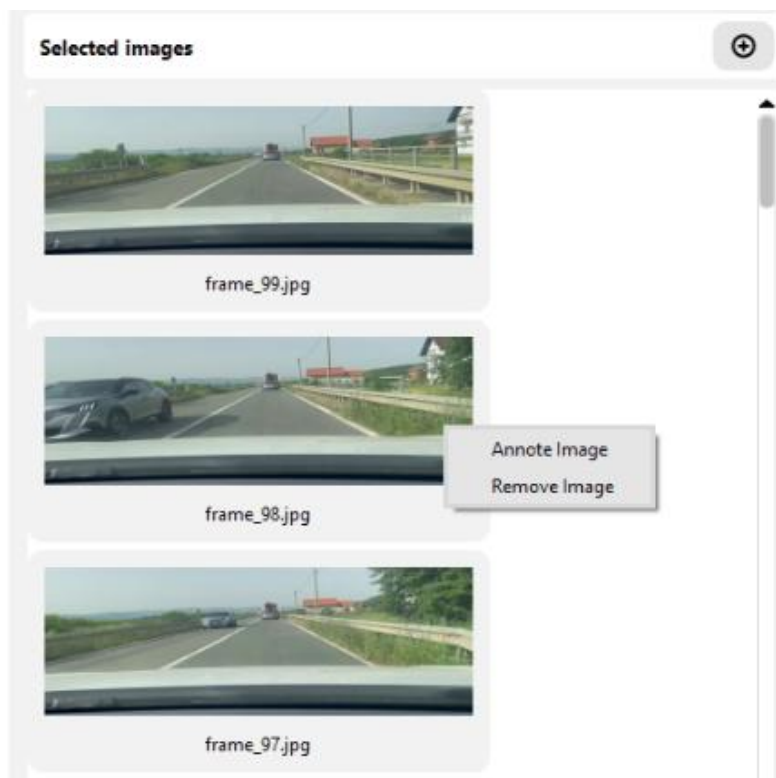
Drugi zahtjev, rukovanje galerijom slika, zadovoljio je potrebu za glatkom i intuitivnom navigacijom kroz slike. Testiranjem je utvrđeno da aplikacija omogućuje jednostavno pregledavanje slika te dodavanje novih slika u galeriju, čime se osigurava fleksibilnost

korisnicima. Ova funkcionalnost također je već objašnjena u podpoglavlju 3.2. Na slici 3.5 pod (a) prikazan je skočni prozor koji omogućuje korisnicima odabir inicijalnih slika koje želi u svom projektu, bilo putem klasičnog načina preko mape ili metodom povuci-i-ispusti. Na slici 4.1 je prikazan prozor za upravljanje galerijom slika. U gornjem desnom kutu nalazi se znak ⊕ koji je zaokružen crvenom bojom, a pritiskom na njega otvoriti će se skočni prozor sa slike 3.5 (a) koji omogućuje korisniku dodavanje novih slika u projekt.



*Slika 4.1 Prikaz prozora za upravljanje galerijom slika*

Na slici 4.2 prikazan je način upravljanja slikama unutar prozora za upravljane galerijom slika. Desnim klikom na sliku dobije se prozor za kontekstualne opcije koji omogućuje korisniku da izbriše sliku iz galerije slika ili da započne s označavanjem željene slike koja se onda prebaci na središnji prozor, prikazan na slici 3.7 unutar aplikacije za označavanje linija vozne trake.



Slika 4.2 Prikaz načina upravljanja slikama unutar aplikacije za označavanje linija vozne trake

Treći zahtjev, ručno označavanje linija vozne trake, ispitan je kroz precizno označavanje točaka duž linija trake. Aplikacija je omogućila crtanje i transparentno prikazivanje linija, čime je korisnicima olakšano precizno pozicioniranje bez ometanja vizualnog prikaza slike. Na slici 3.8 prikazan je proces označavanja nad odabranom slikom.

Četvrti zahtjev, spremanje označenih podataka, provjeren je kroz pohranu svih označenih linija u strukturiranom formatu. Oznake su pohranjene u JSON i txt formatima. JSON datoteke, primjer jedne se nalazi na slici 3.11, sadrže koordinate točaka koje su pritisnute mišem. Txt datoteke, primjer prikazan na slici 3.12, sadrže zapis koordinata nakon *cubic spline* funkcije. U sklopu testiranja rada aplikacije za označavanje linija vozne trake kreirana je kratka Python skripta koja uzima sliku, parsira podatke iz JSON i txt dokumenata te ih označava na slici. Slika 4.2 prikazuje rad aplikacije nad dvjema slikama iz odabranog skupa podataka. Slike 4.2. pod (a) i (c) prikazuju označene linije za koje je korisnik spremio podatke. Linije su širine 20 piksela. Slike pod (b) i (d) označavaju izlaz iz Python skripte. Ljubičaste linije, širine 6 piksela, označavaju podatke preuzete iz txt datoteka, dok roze točke na slikama označavaju svaku oznaku mišem koju je korisnik napravio prilikom označavanja slika tj. prikazuje parsirane podatke iz JSON datoteke. Ovim postupkom potvrđeno je da aplikacija ispunjava sve definirane zahtjeve za označavanje linija vozne trake, pružajući korisnicima alate za učinkovito i precizno označavanje.



*Slika 4.3 Prikaz načina testiranja aplikacije (a) primjer jedne označene slike unutar aplikacije (b) primjer testiranja zapisani koordinata nakon označavanja (a) slike (c) primjer druge označene slike unutar aplikacije (d) primjer testiranja zapisani koordinata nakon označavanja (c) slike*

Nakon što je potvrđena pouzdanost označavanja, zaključeno je da aplikacija zadovoljava sve postavljene zahtjeve za precizno i konzistentno označavanje linija vozne trake. Time je osigurana osnova za daljnje korake istraživanja. U sljedećem poglavlju opisuje se proces izrade vlastitog skupa podataka, koji uključuje prikupljanje slika voznih traka uz različite prometne i vremenske uvjete.

## 4.2 Izrada vlastitog skupa podataka

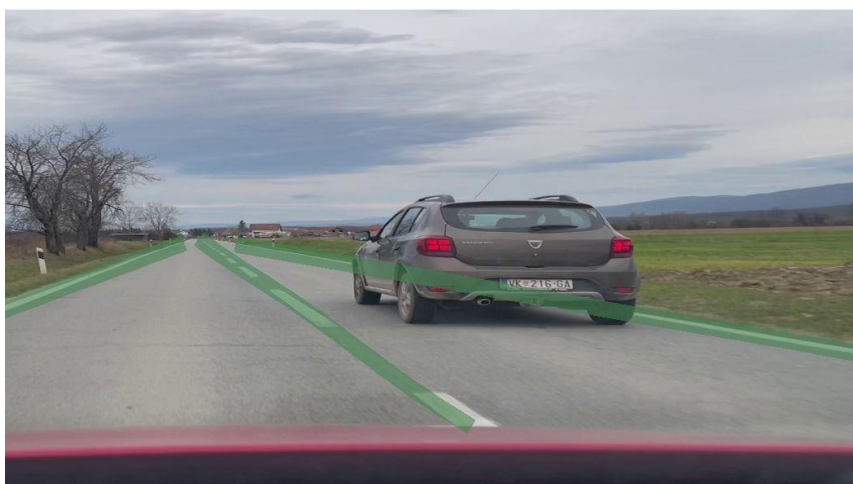
Kako bi se osigurala sveobuhvatnost i kvaliteta kreiranog skupa podataka, provedeni su brojni koraci koji su uključivali snimanje video materijala i izdvajanje pojedinačnih okvira. Snimanje je obavljeno mobilnim uređajem iPhone 12, tablica 4.1, pri čemu su obuhvaćeni različiti uvjeti vožnje kako bi se osigurala raznolikost podataka, uključujući promjene vremenskih uvjeta, doba dana i tipove cesta. Postavke na mobitelu su bile 1800p pri 30fps za video rezoluciju, a kao format video snimanja korišten je HEVC.

*Tablica 4.1 Parametri kamere mobitela iPhone 12*

Parametar	Vrijednost
Kamera	Dvostruka stražnja kamera
Rezolucija	12 megapiksela (svaka leća)
Širokokutna kamera	Otvor blende f/1.6
Ultraširokokutna kamera	Otvor blende f/2.4, vidno polje od 120 stupnjeva
Stabilizacija	Optička stabilizacija slike (OIS) za širokokutnu leću

Video rezolucije	4K pri 24, 30, 60 fps; 1080p pri 30, 60, 120, 240 fps
HDR podrška	HDR s Dolby Vision
Usporeno snimanje	1080p pri 120 fps ili 240 fps
Time-Lapse	Podrška za Time-Lapse s noćnim načinom rada
Audio Zoom	Podrška za usmjeravanje zvuka tijekom video snimanja
Formati video snimanja	HEVC i H.264

U radu [18], u kojem je opisan SCNN model, kreiran je CULane skup podataka, koji je poslužila kao osnova za kreiranje vlastitog skupa podataka. Python skripta izrađena za ovaj proces omogućila je automatsko izdvajanje okvira iz video materijala u redovitim intervalima, osiguravajući pritom dosljednost rezolucije na 1640x590 piksela. Prednost SCNN je ta što ima mogućnost pretpostaviti pozicije linija čak i kad se na slici nalazi prepreka. Prilikom izrade skupa podataka pomoću aplikacije korišten je isti princip što je prikazano na slici 4.4.



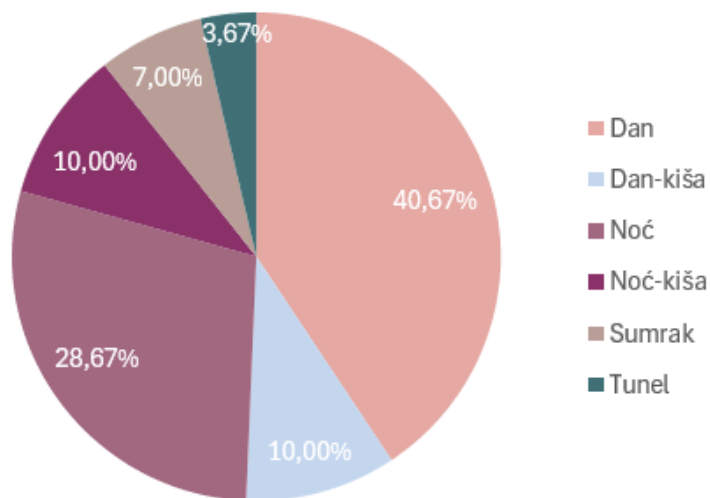
*Slika 4.4 Primjer označavanja linija vozne trake kada je linija djelomično zaklonjena*

Nakon završenog označavanja slika, provjerena je točnost podataka kako bi se osiguralo da su sve oznake ispravne. U slučaju da su uočene greške, podaci koji su se prikazivali krivo ili su krivo zapisani su ispravljeni kako bi se osigurala točnost skupa podataka. Slike koje nisu zadovoljavale kriterije kvalitete uklonjene su iz skupa podataka kako bi se održala visoka kvaliteta prikupljenih podataka. Slika 4.5 prikazuje primjer jedne od takvih slika. Slika je uklonjena jer je vjetrobransko staklo prljavo, što daje dojam da je slika mutna, a također niti jedna linija nije vidljiva. Ovaj proces prikupljanja, obrade i provjere podataka osigurao je da je skup podataka spreman za daljnju upotrebu u analizi i razvoju aplikacije. Skup podataka kreiran u sklopu praktičnog dijela ovog rada sadrži ukupno 1500 slika prikupljenih uz različite uvjete, s ciljem osiguranja raznolikosti scena za detekciju vozničkih traka u realnim i izazovnim uvjetima.



*Slika 4.5 Primjer uklonjene slike zbog prljavog vjetrobranskog stakla i manjka vidljivih linija*

Skup slika raspodijeljen je na način da je 760 slika snimljeno po danu, od čega je 150 snimljeno po kiši, 580 slika je snimljeno po noći, od kojih je 150 slika snimljeno po kiši, dok slike snimljene za vrijeme sumraka i u tunelu čine manji udio sa 105 slika za sumrak i 55 slika za tunel. Na slici 4.6 nalazi se grafički prikaz raspodjele slika prema uvjetima snimanja za kreirani skup podataka u obliku kružnog dijagrama koji ilustrira postotke slika u različitim uvjetima vožnje, naglašavajući udio svake kategorije unutar cijelog skupa podataka. Ovakva struktura skupa omogućava detaljnu analizu performansi algoritma detekcije voznih traka kroz široki spektar vizualnih uvjeta i vremenskih prilika.



*Slika 4.6 Prikaz raspodjele svake kategorije unutar cijelog skupa podataka*

Slike su izdvojene iz videozapisa snimljenih tijekom dana, noći i sumraka te u uvjetima kiše i unutar tunela. Tijekom dana, korišteno je 16 videozapisa, dok je za dnevne uvjete s kišom korišteno 6 videozapisa. Noćne scene obuhvaćene su s 10 videozapisa, a za noćne uvjete s kišom iskorišteno je dodatnih 5 videozapisa. Prikazi sumraka i tunela snimljeni su u manjem opsegu, s ukupno 3 videozapisa za sumrak i 1 za tunel.



Novokreirana baza podataka podijeljena je na tri dijela: skup za treniranje, skup za testiranje i validacijski skup, u omjerima 70%, 20% i 10%. Ukupan broj slika u bazi podataka iznosi 1500, od čega je 1050 slika uključeno u skup za treniranje, 300 slika u testni skup, a 150 slika u validacijski skup. Skupovi su kreirani tako da se iz svake kategorije uzeo proporcionalni broj slika. Ova podjela osigurava da svaka skupina sadrži reprezentativan uzorak iz cijelog skupa podataka, što pridonosi točnosti i pouzdanosti evaluacije modela. Raspodjela je napravljena po uzoru na CULane podatkovnog skupa tako da su kreirana tri tekstualna dokumenta gdje svaki označava jedan od skupova. Unutar dokumenata nalaze se putanje do slika koje pripadaju tom skupu. Označena baza podataka s uključenim tekstualnim datotekama dana je kao elektronički prilog P.4.1. u ovom radu.

Validacijski skup u procesu treniranja modela omogućuje praćenje napretka modela tijekom treniranja. Naime, tijekom svakog koraka treniranja model se testira na validacijskom skupu, što pomaže u procjeni koliko dobro model generalizira na podacima koje nije izravno „vidio“ tijekom treniranja. Na taj način se može pravovremeno prepoznati prekomjerno učenje, gdje se model previše prilagodi specifičnim podacima iz skupa za treniranje, umjesto da uči općenite značajke korisne za nove podatke.

### **4.3 Izgradnja i evaluacija detektora linija voznih traka**

U ovom podpoglavlju prikazana je izgradnja i evaluacija detektora linija voznih traka koristeći odabrane skupove podataka i modele. Korišteni su modeli trenirani na skupovima podataka CULane, TuSimple i LLAMAS, koji su odabrani zbog svoje relevantnosti i raznolikosti prometnih scenarija. Model korišten za testiranje bio je SCNN, implementiran uz pomoć repozitorija PyTorch AutoDrive, koji omogućuje korištenje različitih arhitektura poput VGG16, ResNet34 i ERFNet. Važnost ovog područja detaljno je opisana u poglavlju 2, gdje je naglašena potreba za točnom i pouzdanom detekcijom linija voznih traka u kontekstu autonomnih sustava vožnje.

U odjeljku 4.3.1 opisan je *PyTorchAutoDrive*, koji je korišten kao Python okvir za implementaciju modela za semantičku segmentaciju i detekciju linija voznih traka. Ovaj okvir, kao što mu ime kaže, koristi prednosti *PyTorch* platforme, omogućujući jednostavnu prilagodbu i proširenje postojećih modela. U odjeljku 4.3.2 detaljno je opisano korištenje *Docker*-a za postavljanje virtualnog radnog okruženja. *Docker* je korišten kako bi se osiguralo konzistentno i reproduktivno razvojno okruženje, što je od posebne važnosti za složene projekte koji zahtijevaju specifične verzije softvera i ovisnosti.



### 4.3.1 PyTorchAutoDrive

*PyTorchAutoDrive* je sveobuhvatan repozitorij [36] otvorenog koda koji omogućuje istraživačima i inženjerima implementaciju, treniranje i evaluaciju modela za autonomnu vožnju. Zasnovan na *PyTorch* platformi, ovaj repozitorij sadrži niz naprednih modela za detekciju linija vozne trake i semantičku segmentaciju, čime pokriva ključne aspekte računalnog vida u kontekstu autonomnih vozila. Jedna od glavnih značajki *PyTorchAutoDrive* repozitorija je njegova modularna struktura koja olakšava prilagodbu i proširenje postojećih modela. Repozitorij je organiziran u nekoliko glavnih direktorija koji omogućuju jasan tijek rada. Konfiguracijske datoteke definiraju parametre modela, poput broja epoha (*engl.* Epoch Size), veličine grupe podataka (*engl.* Batch Size), vrste optimizatora i putanja do skupa podataka. Korištenjem ovih datoteka, korisnici mogu lako prilagoditi eksperimente bez izmjene osnovnog koda. Ove datoteke su raspoređene unutar devet glavnih direktorija:

Cjevovod podataka (*engl.* Data Pipeline):

1. Skupovi podataka – direktorij koji omogućuje implementaciju različitih metoda za učitavanje i obradu podataka, prilagodbu ulaznih formata i pred-obradu potrebnu za treniranje.
2. Testna augmentacija – modul koji sadrži funkcije za testiranje i obradu podataka, kao što su augmentacije i druge pomoćne metode.
3. Trening augmentacija – modul koji sadrži skripte za treniranje modela i automatizaciju postupaka vezanih uz obuku i prilagodbu modela.

Cjevovod optimizacije (*engl.* Optimization Pipeline):

4. Funkcija gubitka – koristi se za izračunavanje gubitka tijekom treniranja (npr. *CrossEntropy*).
5. Optimizator – algoritam za optimizaciju modela (npr. SGD).
6. Planer stope učenja – planer stope učenja koji upravlja promjenama stope učenja tijekom treniranja

Opcije modela (*engl.* Model Specific Options):

7. Trening – postavke za treniranje koje uključuju broj epoha, veličinu grupe podataka i opcije distribuiranog treniranja (DDP).
8. Testiranje – postavke za evaluaciju, profiliranje i vizualizaciju, poput putanje do kontrolne točke (*engl.* Checkpoint Path) i veličine slike.

## 9. Model – postavke za definiranje arhitekture modela.

Dodatne opcije (ovo su novije opcije koje su u fazi testiranja i dostupne su kao *beta* verzija):

10. Vizualizacija – koristi se umjesto testiranja ako je specificirano

11. Testni skup podataka – koristi se za zamjenu osnovnog skupa podataka tijekom testnog načina rada ako je specificirano.

12. Vizualizacijski skup podataka – koristi se za zamjenu osnovnog skupa podataka tijekom vizualizacije ako je specificirano.

Cjelokupni postupak treniranja i evaluacije započinje pripremom konfiguracijskih datoteka koje sadrže sve potrebne parametre za pokretanje eksperimenta. Ovi parametri uključuju informacije o arhitekturi modela, hiperparametre, putanje do skupa podataka i druge ključne postavke. Sve konfiguracijske datoteke su već unaprijed definirane i mogu se prilagoditi prema potrebama. U ovom radu, međutim, podaci se testiraju s već istreniranim modelima, stoga se parametri ne mijenjaju prilikom dotreniranja. Kada su konfiguracijske datoteke spremne, treniranje se pokreće pomoću skripti koje integriraju učitavanje podataka, optimizaciju modela i praćenje metrika kao što su preciznost i gubitak. Također je važno napomenuti da u ovom repozitoriju validacijski skup nije korišten tijekom treniranja modela. Umjesto toga, model je evaluiran pomoću internih mjera, kao što su „*training loss*“, koji prikazuje ukupnu pogrešku modela tijekom treniranja, „*loss seg*“, koji mjeri preciznost segmentacije linija vozne trake, i „*loss exist*“, koji prati pogrešku u detekciji postojanja linija. Ove metrike omogućuju optimizaciju modela bez korištenja dodatnog skupa za validaciju, što pojednostavljuje proces treniranja.

Za pokretanje ovog repozitorija potrebne su NVIDIA grafička kartica i CUDA (*engl.* Compute Unified Device Architecture) biblioteka. NVIDIA grafičke kartice omogućuju ubrzanje obrade velikih količina podataka i treniranje modela korištenjem paralelne obrade, dok CUDA predstavlja softversku platformu koja omogućuje korištenje GPU-a za znanstvene izračune. Za postavljanje okruženja u repozitoriju su dane najniže verzije i ovisnosti potrebne za pokretanje koje su prikazane u tablici 4.2. Za postavljanje okruženja potrebno je instalirati PyTorch odgovarajuće verzije, CUDA biblioteku za ubrzanje obrade na GPU-u te ostale ovisnosti definirane u datoteci *requirements.txt*. Na korištenom računalu, opremljenom NVIDIA RTX 3080 grafičkom karticom s 12 GB VRAM-a, operacijski sustav bio je *Ubuntu* verzija 24.4. Procesor je 11th Gen *Intel(R) Core(TM) i9-11900F* s 8 jezgri i 16 niti, a ukupna količina radne memorije (RAM) iznosi 32 GB.

Tablica 4.2 Popis potrebnih ovisnosti i verzija za rad PyTorchAutoDrive repozitorija

Softverske ovisnosti	Verzije
Python	$\geq 3.6$
CUDA	$\geq 9.2$
Pytorch	$\geq 1.6$
TorchVision	$\geq 1.3.5$
MMCV	$\geq 1.3.5$
openCV	=4.5.4.58
scPy	=1.5.4
ninja	=1.8.2
tqdm	=4.62.3

Zbog neslaganja između traženih i postojećih verzija softvera, postavljanje je provedeno unutar *Docker* kontejnera opisanog u dijelu 4.3.2.

### 4.3.2 Docker i postavljanje okruženja

*Docker* je platforma za virtualizaciju koja omogućuje izolirano pokretanje aplikacija unutar kontejnera. Kontejneri su jednostavne jedinice koje sadrže sve potrebne komponente za pokretanje aplikacije, uključujući operacijski sustav, knjižnice i sve ovisnosti specifične za aplikaciju. Za razliku od virtualnih strojeva, koji emuliraju cijeli hardver i zahtijevaju više resursa, kontejneri dijele jezgru operacijskog sustava domaćina, čime se postiže bolja efikasnost i manja potrošnja resursa. Ova značajka omogućuje brzo pokretanje, veći broj istovremeno aktivnih kontejnera i lakšu prenosivost aplikacija između različitih okruženja. Upravljanje *Docker*-om obuhvaća niz osnovnih naredbi za kreiranje, pokretanje i upravljanje kontejnerima. Na primjer, naredba *docker pull* koristi se za preuzimanje gotovih slika s *Docker Hub*-a, dok *docker run* omogućuje pokretanje kontejnera s preuzetom slikom. Za izvršavanje naredbi unutar pokrenutog kontejnera koristi se naredba *docker exec*, što olakšava upravljanje i održavanje aplikacija unutar izoliranog okruženja. *Docker* također podržava kreiranje vlastitih slika pomoću *Dockerfile* datoteka, koje omogućuju definiranje specifičnih postavki i prilagodbu okruženja prema potrebama korisnika.

*Docker Hub* je internetska platforma koja omogućuje korisnicima pristup gotovim slikama za širok raspon aplikacija. Korisnici mogu preuzimati gotove slike koje su već konfigurirane za specifične potrebe ili stvarati vlastite slike kako bi prilagodili okruženje prema vlastitim zahtjevima. Slike se mogu graditi pomoću *Dockerfile* datoteka, koje specificiraju sve korake

potrebne za postavljanje željenog okruženja, uključujući instalaciju operacijskog sustava, knjižnica i ovisnosti. Za potrebe ovog rada korištena je gotova *Docker* slika s operacijskim sustavom *Ubuntu* 18.04, unutar koje je kreiran kontejner u kojem su se instalirale ključne komponente. Instalacija je uključivala *CUDA* 11.1 za ubrzanje obrade na GPU-u, što omogućuje optimizirano treniranje dubokih neuronskih mreža. Uz to, *PyTorch* v1.8 korišten je kao fleksibilan okvir za strojno učenje i implementaciju modela, dok je *MMCV* 2.2 pružao dodatne funkcionalnosti potrebne za rad s računalnim vidom. Unutar kontejnera postavljeno je i virtualno okruženje putem alata *Conda*, uz korištenje *Pythona* 3.8 kao interpretatora. Ova konfiguracija osigurala je dosljedno i ponovljivo radno okruženje, omogućujući provedbu svih potrebnih eksperimenata.

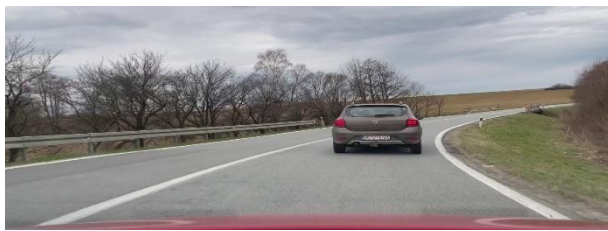
#### **4.4 Evaluacija performansi modela za detekciju linija voznih traka zasnovanih na dubokom učenju**

U ovom potpoglavlju opisuje se postupak evaluacije i analize izvedbe triju različitih modela koji su prethodno trenirani na različitim skupovima podataka. Svaki model testiran je na testnom skupu novokreiranog vlastitog podatkovnog skupa, zatim je dotreniran na trening skupu novokreiranog skupa podataka i ponovno testiran na testnom skupu novokreiranog skupa podataka kako bi se procijenila poboljšanja u performansama. Osim toga, svaki model je posebno treniran na trening skupu vlastitog podatkovnog skupa i evaluiran na testnom skupu novokreiranog vlastitog podatkovnog skupa, bez da je prethodno treniran na nekom od postojećih skupova podataka. Ova analiza obuhvaća ukupno 21 evaluaciju (18 kombinacija testiranja i dotreniravanja 3 dodatna treniranja modela na vlastitom skupu). Analiza uključuje usporedbu preciznosti, odziva i F1 mjere u različitim fazama testiranja, čime se dobiva uvid u prednosti i slabosti svakog pristupa.

Dio 4.4.1 posvećen je usporedbi rezultata dobivenih od strane modela treniranih na trening skupu *CULane* i testiranih na testnom skupu vlastitog skupa podataka, mreže su zatim dodatno trenirane na trening skupu vlastitog skupa podataka i ponovno testirane na istom skupu. U dijelu 4.4.2 prikazani su rezultati testiranja mreže na testnom skupu vlastitog skupa podataka koja je trenirana na *LLAMAS* podatkovnog skupa. Ta mreža je zatim dotrenirana na trening skupu vlastitog skupa podataka i testirana na testnom skupu vlastitog skupa podataka. U dijelu 4.4.3 analiziraju se rezultati testiranja na testnom skupu vlastitog skupa podataka mreže trenirane na trening skupu *TuSimple* podatkovnog skupa, gdje je novokreirani skup podataka prilagođena *TuSimple* skupu podataka što se tiče načina zapisa koordinata i segmentacijskih zapisa. Nakon toga mreža je dotrenirana na trening skupu novokreiranog skupa podataka, a zatim ponovo

testirana na testnom skupu novokreiranog skupa podataka. Zatim se u dijelu 4.4.4 opisuju rezultati treniranja i testiranja modela na isključivo na vlastitom skupu podataka. Svako testiranje popraćeno je primjerima vizualizacije IoU između predviđenih linija i GT oznaka.

Za prikaz rezultata u sljedećim dijelovima izabrana su četiri okvira, prikazana na slici 4.7, iz testnog skupa vlastitog podatkovnog skupa na kojima se prikazuju rezultati rada mreža. Prvi okvir, slika 4.7 (a), prikazuje cestu u ruralnom području, krivina na cesti, djelomičnu naoblaku, automobil ispred koji zaklanja liniju.



(a)



(b)



(c)



(d)

*Slika 4.7 Odabrani okviri iz testnog skupa na kojima će se prikazivati rezultati rada mreža (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Slika pod (b) prikazuje kišni dan uz slabu vidljivost, jednu liniju koja se loše vidi. Pod (c) je prikazan okvir noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo i mutan scenariji radi kapljica vode. Posljedni okvir pod (d) prikazuje vrijeme sumraka, ravnu cestu bez prometa i jako dobru vidljivost linija.

#### **4.4.1 Analiza rezultata modela treniranih na CULane skupu podataka**

Prije evaluacije rezultata modela treniranih na CULane skupu podataka, potrebno je precizno definirati parametre evaluacije kako bi se osigurala konzistentnost i objektivnost procjene. U ovom kontekstu, tri ključne metrike pružaju uvid u učinkovitost modela: preciznost, odziv i F1-mjera. Za procjenu, širina linije postavljena je na određenu vrijednost izraženu u pikselima, kako bi se simulirale stvarne dimenzije na slikama. Također, definiran je prag IoU kao minimalna granica koja određuje kada se detekcija smatra uspješnom. Prije same evaluacije,

potrebno je generirati GT linije koje predstavljaju referentne oznake na slikama. Evaluacija se provodi usporedbom GT linija s linijama koje predviđa model. Predviđanja za koja je IoU iznad postavljenog praga smatraju se točnima i bilježe se kao *true positive* (TP). Preciznost se računa kao omjer broja ispravno predviđenih pozitivnih uzoraka (TP) prema ukupnom broju predviđenih pozitivnih uzoraka (TP + FP), gdje *false positive* (FP) predstavlja broj lažno predviđenih pozitivnih uzoraka. Formula za preciznost glasi:

$$Preciznost = \frac{TP}{TP + FP} * 100\% \quad (4-1)$$

Preciznost pruža mjeru pouzdanosti modela pri identifikaciji linija vozne trake. Odziv mjeri omjer ispravno predviđenih pozitivnih uzoraka (TP) u odnosu na stvarni broj pozitivnih uzoraka (TP + FN), gdje *false negative* (FN) predstavlja broj stvarnih pozitivnih uzoraka koje model nije uspio detektirati. Formula za odziv glasi:

$$Odziv = \frac{TP}{TP + FN} * 100\% \quad (4-2)$$

Odziv ističe sposobnost modela da detektira sve relevantne značajke unutar slike. F1-mjera kombinira preciznost i odziv kako bi se dobila uravnotežena procjena modela, posebno korisna kada je potrebno razmotriti i pouzdanost i potpunost predikcija. Formula za F1-mjeru glasi:

$$F1 = 2 * \frac{Odziv * Preciznost}{Odziv + Preciznost} * 100\% \quad (4-3)$$

F1-mjera pruža cjeloviti prikaz učinkovitosti modela uzimajući u obzir obje komponente. Primjenom ovih evaluacijskih metrika, moguće je analizirati performanse modela treniranih na CULane skupu podataka. U Tablici 4.3 prikazani su rezultati testiranja mreža, SCNN mreža s VGG16, ResNet34 i ERFNet *backbone*-ovima, treniranih na CULane podatkovnom skupu i testirani na testnom skupu vlastitog skupa podataka. Podatkovni skup CULane sastoji se od 9 kategorija: *normal, crowded, night, no line, shadow, arrow, dazzle light, cross i curve*. Dokument koji definira te kategorije nalazi se na putanji: *tools/culane\_evaluation/eval.sh*. U sklopu testiranja, na primjer *eval.sh* dokumenta, vlastiti skup podataka je podijeljen na pet kategorija: dan, dan-kiša, noć, noć-kiša, sumrak i tunel. Također, prije dotreniravanja mreže pomoću repozitorija [37] su generirane segmentacijske maske, to su slike koje model koristi za trening i evaluaciju. Unutar tablice 4.3 stupac „Dotrenirana“ označava jesu li mreže prilikom testiranja trenirane na CULane skupu podataka ili su testirane nakon dodatnog treniranja na trening skupu vlastitog podatkovnog

skupa. U stupcu „Mjere“ nalaze se 3 retka koji daju vrijednosti preciznosti, odziva i F1 mjere u postotcima za svaku od navedenih kategorija.

Dotrenirani modeli postižu značajno bolje rezultate u svim metrikama (preciznost, odziv, i F1-mjera). Preciznost kod dotreniranog SCNN-VGG16 modela iznosi 97,49%, u usporedbi s 64,56% kod modela koji nije dotreniran. Slično, SCNN-ResNet34 i SCNN-ERFNet također pokazuju veće vrijednosti kod dotreniranih verzija, što ukazuje na značajnu prednost dodatnog treniranja na lokalnom skupu podataka.

*Tablica 4.3 Rezultati evaluacije modela treniranih na podatkovnom skupu CULane i testiranih na testnom skupu vlastitog podatkovnog skupa*

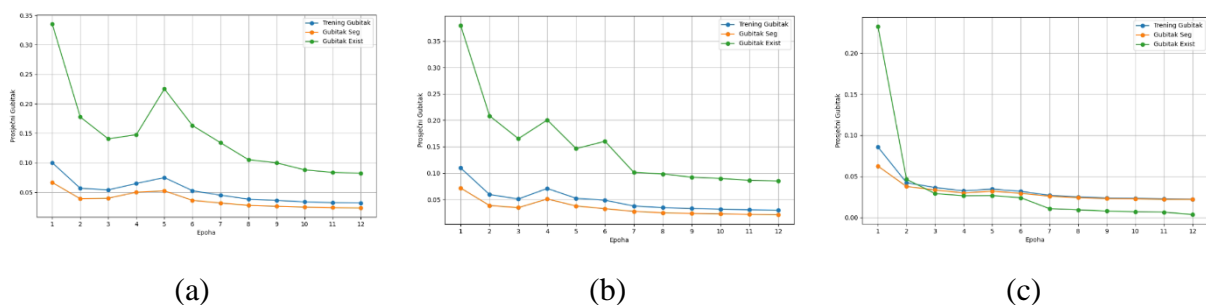
Mreža	Dotrenirana	Mjere	Dan [%]	Dan-kiša [%]	Noć [%]	Noć-kiša [%]	Sumrak [%]	Tunel [%]	Ukupni rezultat [%]
SCNN-VGG16	Ne	Preciznost	57,14	38,00	77,04	60,71	90,24	62,50	64,56
		Odziv	53,05	39,58	67,71	39,08	58,73	30,30	54,43
		F1	55,02	38,76	72,08	47,55	71,15	40,82	59,07
	Da	Preciznost	94,38	100	99,55	100	98,39	96,97	97,49
		Odziv	90,32	95,83	99,10	100	96,83	96,97	95,36
		F1	92,31	97,87	99,33	100	97,6	96,97	96,41
SCNN-ResNet34	Ne	Preciznost	43,88	28,20	68,90	49,03	68,18	48,39	42,88
		Odziv	30,82	22,92	50,67	28,74	23,81	45,45	52,68
		F1	36,21	25,29	58,4	36,23	35,29	46,88	36,15
	Da	Preciznost	95,22	97,78	99,55	100	96,83	100	96,91
		Odziv	92,83	91,67	99,10	100	96,83	100	97,65
		F1	94,01	94,62	99,33	100	96,83	100	96,18
SCNN-ERFNet	Ne	Preciznost	53,92	25,93	77,11	65,67	84,09	48,00	61,40
		Odziv	56,63	29,17	69,51	50,57	58,73	36,36	57,30
		F1	55,24	27,45	73,11	57,14	69,16	41,38	59,28
	Da	Preciznost	95,62	93,62	99,55	100	98,41	100	97,66
		Odziv	93,91	91,67	99,55	100	98,41	100	96,86
		F1	94,76	92,32	99,55	100	98,41	100	97,26

Modeli koji su dotrenirani postižu izrazito visoku preciznost u gotovo svim kategorijama. Na primjer, SCNN-VGG16 postiže 100% preciznost u kategorijama "dan-kiša", "noć-kiša" i "tunel" nakon dotreniravanja. Kod SCNN-ERFNet, sličan rezultat se postiže, s preciznošću od 100% u kategorijama "noć-kiša" i "tunel", što upućuje na visoku sposobnost modela za detekciju linija u složenim uvjetima nakon dodatnog treniranja. Modeli koji su dotrenirani postižu izrazito visoku preciznost u gotovo svim kategorijama. Na primjer, SCNN-VGG16 postiže 100%

preciznost u kategorijama "dan-kiša", "noć-kiša" i "tunel" nakon dotreniranja. Kod SCNN-ERFNet, sličan rezultat se postiže, s preciznošću od 100% u kategorijama "noć-kiša" i "tunel", što upućuje na visoku sposobnost modela za detekciju linija u složenim uvjetima nakon dodatnog treniranja. Najveće varijacije između dotreniranih i ne-dotreniranih modela vidljive su u kategorijama kao što su "dan-kiša" i "tunel". Na primjer, SCNN-ResNet34 bez dotreniranja ima preciznost od samo 28,20% u uvjetima "dan-kiša", dok se s dotreniranjem taj rezultat povećava na 97,78%. Ove razlike sugeriraju da dotreniranje značajno pomaže u prilagodbi modela na specifične uvjete, što je važno za stvarnu primjenu.

Prije dodatnog treniranja mreža na novokreiranom podatkovnom skupu najslabiji rezultat prema F1 mjeri pokazuje SCNN-ResNet34, dok najbolji rezultat ostvaruje SCNN-ERFNet. Nakon dotreniranja mreže daju znatno bolje rezultate, mreže u prosjeku postižu ukupnu F1 mjeru od 96,62%. Najslabiji rezultat pokazuje SCNN-ResNet34, dok najbolji rezultat ostvaruje SCNN-ERFNet koji ostvaruje ukupnu F1 mjeru od 100% u čak 2 kategorije, noć-kiša i tunel.

Treniranje modela odvijalo se kroz 12 epoha, zadržavajući isti broj epoha kao i u izvornom repozitoriju kako bi se održala dosljednost i izbjegle bilo kakve promjene u konfiguraciji mreže. Na slici 4.8 prikazani su grafovi treniranja ovih modela po epohama, koji uključuju praćenje triju metrika – ukupnog gubitka tijekom treniranja, pogreške segmentacije i pogreške u detekciji postojanja linija. Ove metrike, opisane u dijelu 4.3.1, korištene su za procjenu napretka modela tijekom cijelog procesa treniranja.



Slika 4.8 Prosječan gubitak po epohi tijekom dodatnog treniranja (a) VGG16 model (b) ResNet34 model (c) ERFNet model

Graf na slici 4.8 pod (a) pokazuje kako gubitak pada tijekom prvih epoha, što znači da model uči. Međutim, u sredini treniranja (između epoha 4 i 6), gubitak za postojanje linija (Gubitak Exist) oscilira, što može ukazivati na nestabilnost u učenju ili na previše složene podatke. Nakon toga, svi gubici se stabiliziraju, što znači da je model postao stabilniji i bolji u generalizaciji. Pod (b) graf pokazuje da gubitak postojanja linija (zelena linija) ima početne oscilacije između epoha 3 i 6, a zatim se stabilizira, što ukazuje na izazove tijekom učenja. Trening gubitak i gubitak



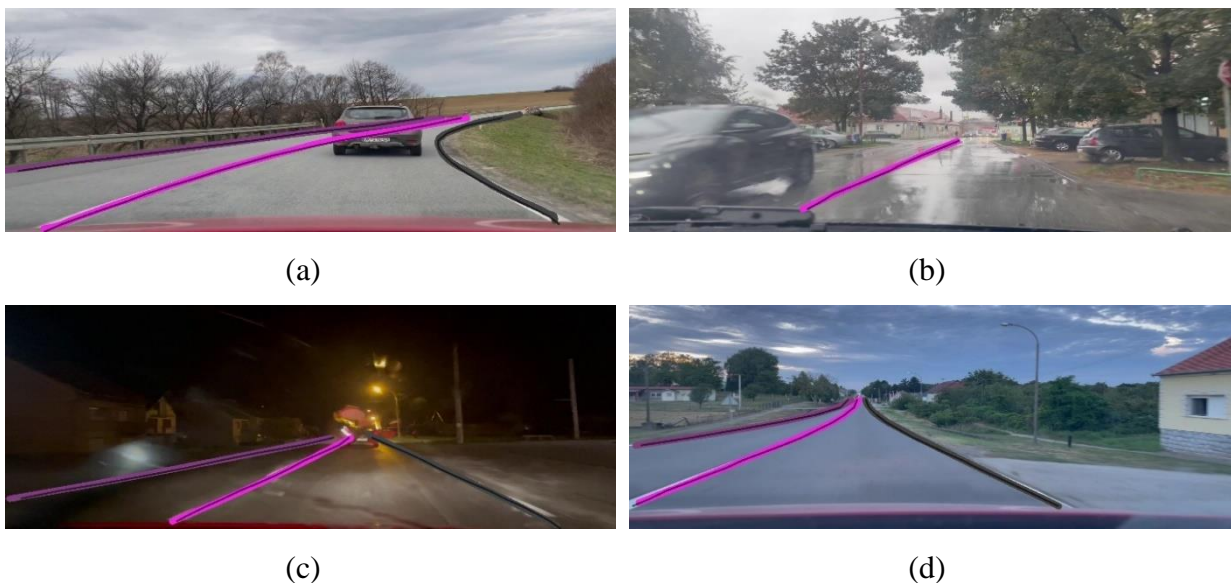
segmentacije (plava i narančasta linija) se stabilno smanjuju, što znači da model postupno i pouzdano uči. U ovom grafu gubici treninga, segmentacije, i postojanja linija svi pokazuju nagli pad na početku, što znači da model brzo uči ključne značajke. Nakon toga, svi gubici postaju vrlo stabilni i nastavljaju se smanjivati do niskih vrijednosti, što pokazuje da model uspješno konvergira bez većih oscilacija. To znači da se model dobro trenira i postaje sve bolji kako epohe napreduju.

Na slici 4.9 prikazana su predviđanja modela SCNN-VGG16 treniranog na CULane skupu podataka u usporedbi s označenim linijama u vlastitom podatkovnom skupu. Na ovoj slici debele linije predstavljaju prethodno kreirane segmentacijske maske, dok tanke crne linije označavaju predviđanja modela. Na slikama se može primijetiti da se pod (a) i (d) prikazuju samo linije vozne trake, a ne cijelo vozno područje, dok pod (c) označava samo središnju liniju.



*Slika 4.9 Prikaz predviđenih linija SCNN-VGG16 modela treniranog na CULane podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Najslabiji rezultat model postiže za kategoriju "dan-kiša," što je vidljivo na slici 4.9 pod (b), gdje model ne detektira niti jednu liniju, što ukazuje na izazove u prepoznavanju pri lošijim vremenskim uvjetima. Rezultati prikazani na slici 4.8 u skladu su s rezultatima prikazanim u tablici 4.3, čime se potvrđuje dosljednost u procjeni modela. Na slici 4.10 na isti način kao i kod 4.9 prikazano je predviđanje sada dotreniranog SCNN-VGG16 modela. Kao što se vidi iz sva 4 okvira, sve linije su označene bez prevelikih odstupanja. Najgore rezultate prema tablici 4.3 daje za kategoriju „dan“, dok su najbolji rezultati za kategoriju „noć-kiša“, gdje je F1 mjera 100%.



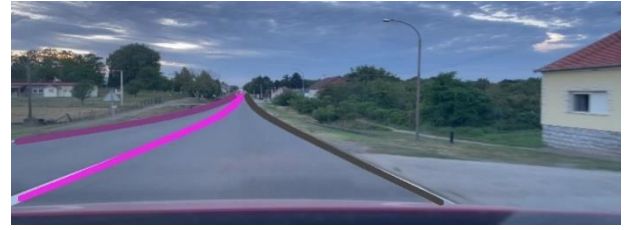
*Slika 4.10 Prikaz predviđenih linija SCNN-VGG16 modela treniranog na CULane podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Model SCNN-ResNet34 pokazao se kao najslabiji među trima mrežama treniranim na CULane podatkovnom skupu, s obzirom na rezultate testiranja na vlastitom skupu podataka. Na slici 4.11 prikazana su predviđanja ovog modela, gdje je jasno vidljivo da u različitim uvjetima uspješnost detekcije linija vozne trake varira. Na slikama označenim kao (b), (c) i (d), model nije u mogućnosti detektirati niti jednu liniju, što ukazuje na značajne poteškoće pri prilagodbi na nove uvjete prikazane u vlastitom skupu podataka. Nasuprot tome, na slici (a) model uspijeva detektirati srednju liniju i dio desne rubne linije, što sugerira da model ipak pokazuje ograničenu sposobnost prepoznavanja u određenim scenarijima. Prema rezultatima prikazanim u tablici 4.3, najveći postotak F1 mjere model postiže za kategoriju "noć". Nakon dotreniravanja SCNN-ResNet34 mreže na vlastitom skupu podataka, rezultati su pokazali značajno poboljšanje. Prema tablici 4.3, najlošiji rezultati zabilježeni su za kategoriju „dan,“ dok su najbolji rezultati ostvareni u kategorijama „noć-kiša“ i „tunel“, gdje je F1 mjera dosegla 100%.





(c)



(d)

*Slika 4.11 Prikaz predviđenih linija SCNN-RESNET34 modela treniranog na CULane podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Na slici 4.12 vidljivo je da su, nakon dotreniravanja modela, sve linije uspješno detektirane. Ovi rezultati sugeriraju da model, nakon dodatnog prilagođavanja, pokazuje visoku razinu točnosti u složenijim ili manje uobičajenim uvjetima.



(a)



(b)



(c)



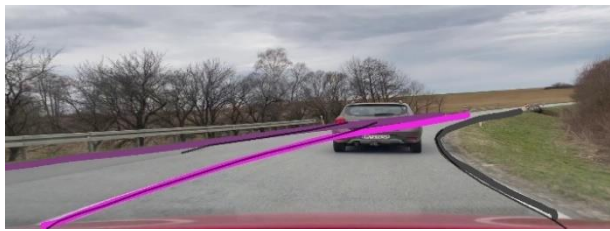
(d)

*Slika 4.12 Prikaz predviđenih linija SCNN-RESNET34 modela treniranog na CULane podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

SCNN-ERFNet pokazuje najbolje rezultate među promatranim modelima, kako prije, tako i nakon dotreniravanja na vlastitom skupu podataka. Prije dotreniravanja, model postiže najlošije rezultate za kategoriju "dan-kiša", dok najbolje rezultate ostvaruje za uvjete "sumrak". Na slici 4.13 pod (a) model uspješno detektira sve tri označene linije, no ne u njihovoj punoj duljini, što upućuje na ograničenu preciznost u kompleksnijim i neviđenim uvjetima. Na slici (b), koja predstavlja kategoriju "dan-kiša", model detektira pravu liniju i desnu rubnu liniju koja u stvarnosti



nije prisutna, što ukazuje na tendenciju modela da pogrešno interpretira detalje u složenim vremenskim uvjetima. Na slici (c) model ne uspijeva detektirati niti jednu liniju, što ističe njegova ograničenja u uvjetima s niskim kontrastom ili slabijom vidljivošću zbog kiše.



(a)



(b)



(c)



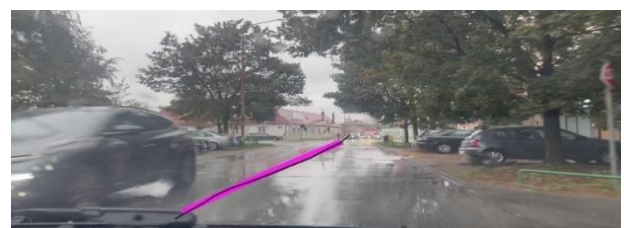
(d)

*Slika 4.13 Prikaz predviđenih linija SCNN-ERFNET modela treniranog na CULane podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Na slici pod (d) model uspješno prepoznaje linije unutar trenutne vozne trake, ali ne obuhvaća cijelo vozno područje, što može ukazivati na potrebu za dodatnom prilagodbom modela za potpuniju detekciju voznih linija u različitim uvjetima i okruženjima. Nakon dotreniravanja modela SCNN-ERFNet, rezultati se značajno poboljšavaju, pri čemu F1 mjera skače s 59,28% na 97,26%. Iako su najlošiji rezultati zabilježeni u kategoriji "dan", najbolji rezultati ostvareni su za uvjete "noć-kiša" i "tunel," gdje je F1 mjera dosegula 100%. Na slici 4.14 jasno je prikazano da su u sva četiri scenarija pronađene sve označene linije, što pokazuje visoku točnost modela nakon prilagodbe vlastitom skupu podataka. Ovi rezultati sugeriraju da je model jako prilagodljiv na neviđene uvijete.



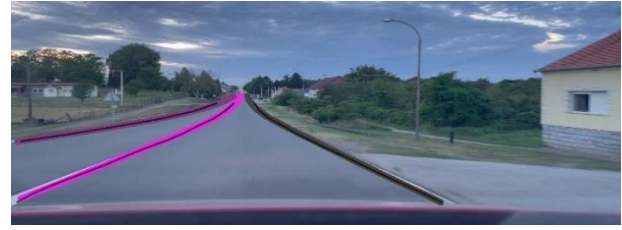
(a)



(b)



(c)



(d)

*Slika 4.14 Prikaz predviđenih linija SCNN-ERFNET modela treniranog na CULane podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

S druge strane, relativno niža F1 mjera za uvjet "dan" može upućivati na poteškoće pri detekciji linija u dnevnim uvjetima, gdje prisutnost jačeg osvjetljenja i reflektivnih elemenata može ometati model. Nakon dotreniravanja, svi promatrani modeli bilježe značajna poboljšanja u F1 mjeri, posebno u složenijim uvjetima poput "noć-kiša" i "tunel". Ovi rezultati pokazuju da modeli u početku slabije reagiraju na neviđene uvjete, ali nakon dotreniravanja postižu izrazito dobre rezultate. Ova poboljšanja sugeriraju da je dotreniravanje prilagodilo modele specifičnim karakteristikama vlastitog skupa podataka, omogućujući im bolje prepoznavanje linija vozne trake čak i u uvjetima smanjene vidljivosti i visokog kontrasta. Poboljšana preciznost nakon dotreniravanja također može biti rezultat optimizacije modela za raznolike uvjete osvjetljenja i refleksije na cesti, koji su izazovni u dnevnim scenarijima, ali olakšavaju detekciju noću. Slabije performanse u kategoriji "dan" mogu biti posljedica prljavog vjetrobranskog stakla, pri čemu sunčev odsjaj dodatno otežava vidljivost linija u uvjetima niske svjetlosti koja se reflektira od ceste.

#### **4.4.2 Analiza rezultata modela treniranih na LLAMAS skupu podataka**

U ovom dijelu analiziraju se rezultati modela treniranih na LLAMAS skupu podataka. Repozitorij [36] omogućava da se LLAMAS podatkovni skup konvertira u format zapisa kompatibilan s CULane skupom podataka, čime se LLAMAS podaci mogu koristiti s originalnim skriptama CULane skupu, uključujući skriptu eval.sh za evaluaciju. Ovaj pristup osigurava dosljednost u procesu treniranja i testiranja, omogućujući primjenu istih metrika i kategorija evaluacije kao kod modela treniranih na CULane skupu. Zahvaljujući ovoj prilagodbi, postalo je moguće dobiti usporedive rezultate po kategorijama (npr. dan, kiša, noć) i za modele trenirane na LLAMAS skupu podataka, omogućavajući evaluaciju njihovih performansi u istim uvjetima. U Tablici 4.4 prikazani su rezultati testiranja mreža treniranih na LLAMAS podatkovnom skupu i testiranih na vlastitom skupu podataka.

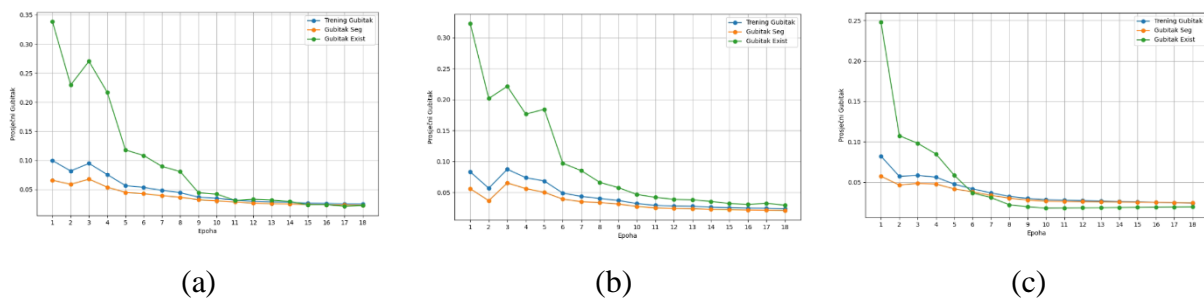
Tablica 4.4 Rezultati evaluacije modela treniranih na podatkovnom skupu LLAMAS i testiranih na testnom skupu vlastitog podatkovnog skupa

Mreža	Dotrenirana	Mjere	Dan [%]	Dan-kiša [%]	Noć [%]	Noć-kiša [%]	Sumrak [%]	Tunel [%]	Ukupni rezultat [%]
SCNN-VGG16	Ne	Preciznost	57,65	39,24	82,40	61,64	93,65	83,87	67,03
		Odziv	70,25	64,58	86,10	51,72	93,65	78,79	74,90
		F1	63,33	48,82	84,21	56,25	93,65	81,13	70,75
	Da	Preciznost	95,99	95,65	100	98,85	96,83	100	97,79
		Odziv	94,27	91,67	99,55	98,85	96,83	100	96,73
		F1	95,12	93,62	99,77	98,81	96,83	100	97,26
SCNN-ResNet34	Ne	Preciznost	57,53	38,03	74,37	61,43	98,41	57,69	64,38
		Odziv	68,46	56,25	79,37	49,43	98,41	45,45	70,26
		F1	62,25	45,38	76,79	54,78	98,41	50,85	67,19
	Da	Preciznost	95,64	95,45	99,55	98,85	98,41	100	97,65
		Odziv	94,27	87,50	99,10	98,85	98,41	100	96,45
		F1	94,95	99,13	99,32	98,85	98,41	100	97,05
SCNN-ERFNet	Ne	Preciznost	54,74	38,71	73,68	64,20	93,55	52,17	63,22
		Odziv	64,16	50,00	81,61	59,77	92,06	36,36	96,17
		F1	59,07	43,64	77,45	61,90	92,80	42,86	66,06
	Da	Preciznost	95,26	86,96	99,10	98,85	98,41	96,97	96,69
		Odziv	93,55	83,33	99,10	98,85	98,41	96,97	95,77
		F1	94,39	85,11	99,10	98,85	98,41	96,97	96,23

Prije dotreniravanja, mreže u prosjeku postižu ukupnu F1 mjeru od 68%, što ukazuje na ograničenu sposobnost modela da precizno prepozna linije vozne trake u različitim uvjetima. Prije dodatnog treniranja na vlastitom skupu podataka, mreže trenirane na LLAMAS podatkovnom skupu postižu znatno bolje rezultate prilikom testiranja na vlastitom testnom skupu u usporedbi s mrežama treniranim na CULane podatkovnom skupu. Razlog za ovakvu razliku leži u činjenici da je LLAMAS podatkovni skup sadržajno sličniji slikama iz vlastitog skupa. Ta sličnost omogućuje modelima treniranim na LLAMAS skupu bolju generalizaciju i prepoznavanje značajki na vlastitim podacima, dok modeli trenirani na CULane skupu, koji ima različite scenarije i uvjete, teže prepoznaju ključne elemente na slici. Rezultati dobiveni testiranjem mreža treniranih na CULane i LLAMAS bazama podataka na vlastitom testnom skupu pokazuju vrlo slične performanse, s razlikama u preciznosti, odzivu i F1 mjeri unutar raspona od  $\pm 1\%$ . VGG16 i ResNet34 modeli trenirani na LLAMAS podatkovnom skupu daju malo bolje rezultate od istih tih modela treniranih na CULane podatkovnom skupu, dok model ERFNet daje malo bolje rezultate kada je treniran na CULane podatkovnom skupu. Takvi slični rezultati ukazuju na to da su oba

skupa podataka, CULane i LLAMAS, sadržajno dovoljno slična, s obzirom na scenarije detekcije linija vozničkih traka i izazove s kojima se modeli suočavaju tijekom treniranja. Ova sličnost omogućuje modelima da razviju gotovo identične sposobnosti prepoznavanja značajki relevantnih za detekciju linija, što rezultira sličnim učinkom kada se testiraju na vlastitom skupu podataka.

Treniranje modela odvijalo se kroz 18 epoha. Na slici 4.15 prikazani su grafovi treniranja ovih modela po epohama, koji uključuju praćenje prethodno objašnjene tri metrike. U grafu pod (a), gubitak postojanja linija (zeleno) u početku oscilira između epoha 2 i 4, a zatim se postupno smanjuje i stabilizira nakon 9. epohe. Trening gubitak (plavo) i gubitak segmentacije (narančasto) također padaju i pokazuju manje oscilacije u početnim epohama, ali postaju stabilni od 6. epohe nadalje. Pod (b) je vidljivo da gubitak postojanja linija (zeleno) pokazuje oscilacije između epoha 2 i 5, no nakon toga značajno pada i stabilizira se od 7. epohe. Trening gubitak (plavo) i gubitak segmentacije (narančasto) također opadaju s povremenim manjim oscilacijama, ali postaju stabilni oko 6. epohe. To sugerira da je model u početku imao poteškoća s učenjem, ali se stabilizirao u kasnijim epohama, što je dobar znak za konvergenciju modela. U grafu pod (c) gubitak postojanja linija (zeleno) pokazuje značajan pad nakon prve epohe, a zatim se nastavlja stabilno smanjivati do 6. epohe, nakon čega ostaje konstantan. Trening gubitak (plavo) i gubitak segmentacije (narančasto) također padaju tijekom prvih nekoliko epoha i stabiliziraju se nakon epohe 6. Graf pokazuje da je model brzo naučio ključne značajke u ranim epohama, a zatim konvergirao i postao stabilan tijekom kasnijih epoha.

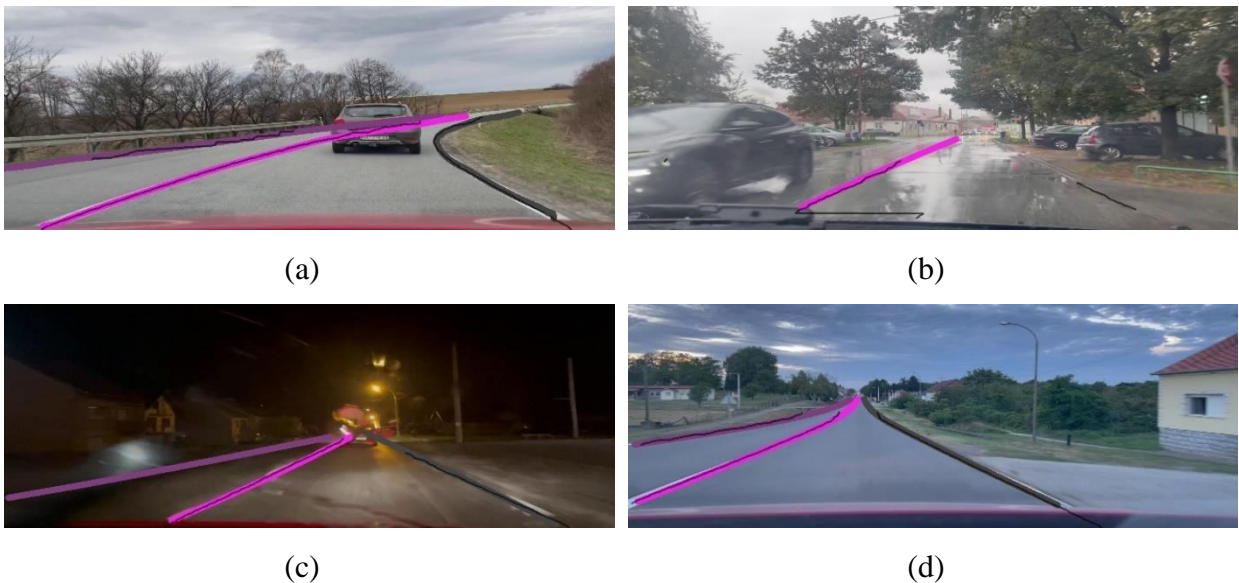


Slika 4.15 Prosječan gubitak po epohi tijekom dodatnog treniranja (a) VGG16 model (B) ResNet34 model (c) ERFNet model

Najslabiji rezultat pokazuje SCNN-ERFNet, dok najbolji rezultat ostvaruje SCNN-VGG16. Za dodatno treniranje modela Nakon dotreniravanja, rezultati su značajno poboljšani, te mreže sada u prosjeku postižu F1 mjeru od 96,85%. Najslabiji rezultat nakon dotreniravanja pokazuje SCNN-ResNet34, dok najbolji rezultat postiže SCNN-VGG16, koji ostvaruje ukupnu F1 mjeru od 100% u kategoriji "tunel". Na slici 4.16 prikazana su predviđanja modela SCNN-VGG16



treniranog na CULane skupu podataka, uspoređena s označenim linijama u vlastitom podatkovnom skupu. Oznake na slikama odgovaraju onima u podpoglavlju 4.4.1. Sa slike 4.16 može se vidjeti da su linije detektirane na svakoj slici, pri čemu su u slučajevima (a) i (d) detektirane sve tri linije, iako ne u njihovoj punoj duljini. Na slici (b) model ispravno detektira označenu liniju, ali prikazuje i desnu liniju vozne trake koja nije označena, dok na slici (c) model detektira samo trenutnu voznu traku, ali ne i cijelo vozno područje. Najslabiji rezultat model ostvaruje za kategoriju "dan-kiša," što je vidljivo na slici 4.9 pod (b), gdje model ne detektira niti jednu liniju, što ukazuje na izazove u prepoznavanju pri lošijim vremenskim uvjetima. Nasuprot tome, najbolji rezultat postiže u kategoriji "sumrak," što je ekvivalentno primjeru sa slike 4.14 (d), gdje su linije detektirane s visokim stupnjem točnosti.



*Slika 4.16 Prikaz predviđenih linija SCNN-VGG16 modela treniranog na LLAMAS podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

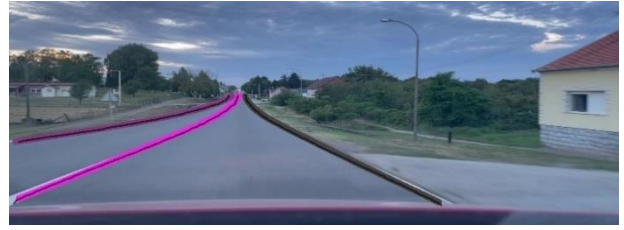
Na slici 4.17 prikazano je predviđanje sada dotreniranog SCNN-VGG16 modela, na isti način kao na slici 4.16. Kao što je vidljivo u sva četiri okvira, sve linije vozne trake su uspješno označene, bez značajnih odstupanja.







(c)



(d)

*Slika 4.17 Prikaz predviđenih linija SCNN-VGG16 modela treniranog na LLAMAS podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Prema podacima iz tablice 4.4, model postiže najbolje rezultate u kategoriji "tunel," gdje je F1 mjera dosegla 100%. Najslabije rezultate model postiže u uvjetima "dan-kiša," što sugerira da vremenski uvjeti poput kiše i promjenjivog osvjetljenja predstavljaju izazov za pouzdanu detekciju linija. Na slici 4.18 prikazana su predviđanja modela SCNN-ResNet34, pri čemu je jasno vidljivo da detekcija linija vozne trake varira ovisno o uvjetima.



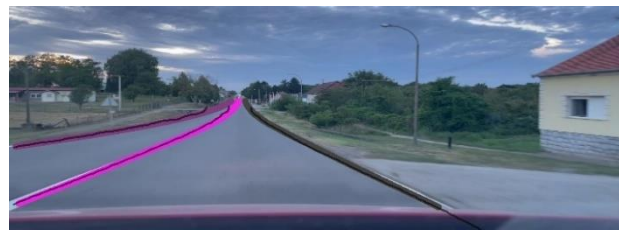
(a)



(b)



(c)

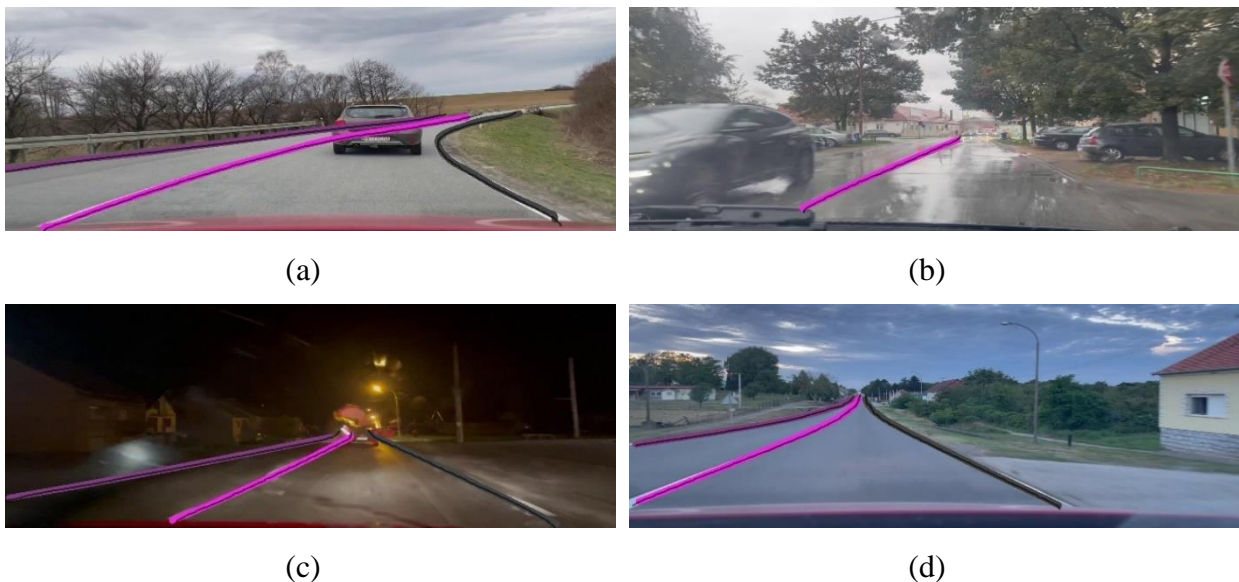


(d)

*Slika 4.18 Prikaz predviđenih linija SCNN-RESNET34 modela treniranog na LLAMAS podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Na slikama (a) i (d) model uspješno detektira sve označene linije vozne trake. Posebno se ističe slika (d), koja prikazuje kategoriju "sumrak," u kojoj model postiže visoku točnost s F1 mjerom od 98,41%. Na slici (b) model djelomično detektira liniju vozne trake i označava desnu rubnu liniju koja zapravo nije označena, što ukazuje na moguću konfuziju modela u složenijim

uvjetima i potrebu za dodatnom prilagodbom u prepoznavanju stvarnih linija. Na slici (c) označene su linije trenutne vozne take ali ne i cijelog voznog područja. Nakon dotreniravanja SCNN-ResNet34 mreže na vlastitom skupu podataka, rezultati su pokazali značajno poboljšanje. Prema podacima iz tablice 4.4, najlošiji rezultati zabilježeni su u kategoriji "dan," dok su najbolji rezultati ostvareni u kategoriji "tunnel," gdje je F1 mjera dosegla 100%, što ukazuje na visoku preciznost modela u uvjetima smanjene osvijetljenosti i stabilnog kontrasta. Na slici 4.19 vidljivo je da su, nakon dotreniravanja, sve linije vozne trake uspješno detektirane bez značajnih odstupanja. Ovi rezultati sugeriraju da model, nakon dodatnog prilagođavanja, pokazuje visoku razinu točnosti u složenijim ili manje uobičajenim uvjetima, kao što su oni s manjim vanjskim utjecajima poput refleksije ili promjenjivog osvjetljenja. Dotrenirani model očito bolje prepoznaje linije u uvjetima sa stabilnim osvjetljenjem, dok izazovi poput jakog dnevnog svjetla ostaju područja gdje bi dodatne prilagodbe mogle poboljšati točnost detekcije.



*Slika 4.19 Prikaz predviđenih linija SCNN-RESNET34 modela treniranog na LLAMAS podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

SCNN-ERFNet pokazuje najslabije rezultate među promatranim modelima prije dotreniravanja na vlastitom skupu podataka. Model postiže najlošije rezultate za kategoriju "tunnel," dok najbolje rezultate ostvaruje u uvjetima "sumrak." Na slici 4.20 pod (a) model uspješno detektira sve tri označene linije, ali ne u njihovoj punoj duljini, što ukazuje na ograničenu preciznost u kompleksnijim i neviđenim uvjetima. Na slici (b), koja predstavlja kategoriju "dan-kiša," model prikazuje vrlo loše rezultate u detekciji linija. U ovoj situaciji, model je uspio detektirati samo početak srednje označene linije, dok je s desne strane vozne trake detektirao

nepostojeću liniju s kratkom duljinom, što upućuje na nesigurnost u složenim vremenskim uvjetima. Na slici (c) model uspijeva označiti linije unutar trenutne vozne trake, ali ne obuhvaća cijelo vozno područje. Na slici pod (d) model također detektira linije unutar trenutne vozne trake, ali ne cijelog voznog područja; osim toga, u daljini prikazuje horizontalnu liniju koja u stvarnosti ne postoji, što ukazuje na moguću konfuziju modela prilikom obrade udaljenih elemenata na slici.



(a)



(b)



(c)



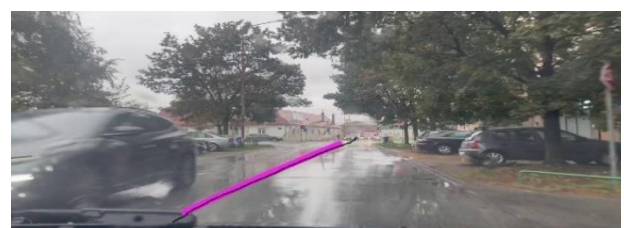
(d)

*Slika 4.20 Prikaz predviđenih linija SCNN-ERFNET modela treniranog na LLAMAS podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Ovi rezultati naglašavaju potrebu za dodatnim dotreniravanjem kako bi model bolje prepoznavao relevantne linije u složenim i dinamičnim uvjetima. Nakon dotreniravanja modela SCNN-ERFNet, rezultati se značajno poboljšavaju. Iako ovaj model i dalje daje najlošije rezultate od sva tri promatrana modela, ako pogledamo tablicu 4.4, F1 mjera mu je sada samo 1,03% niža od performansi najboljeg modela. Najlošiji rezultati zabilježeni su u kategoriji "dan-kiša," dok su najbolji rezultati ostvareni u kategoriji "noć," gdje je F1 mjera dosegla impresivnih 99,10%. Na slici 4.21 jasno je prikazano da su u sva četiri scenarija uspješno pronađene sve označene linije, što ukazuje na visoku točnost modela nakon prilagodbe vlastitom skupu podataka.



(a)

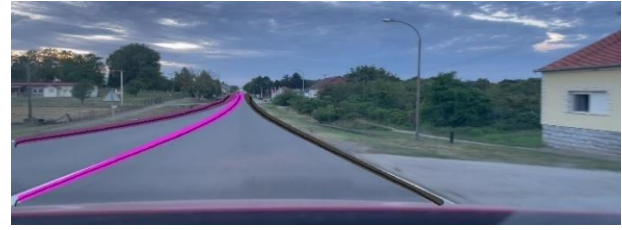


(b)





(c)



(d)

*Slika 4.21 Prikaz predviđenih linija SCNN-ERFNET modela treniranog na LLAMAS podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Prije dotreniravanja, sve mreže pokazuju značajne varijacije u performansama, s najnižim rezultatima zabilježenim u kategorijama "dan-kiša" i "tunel." Najbolji rezultati prije dotreniravanja postignuti su u kategorijama "sumrak" i "noć," dok je ukupna F1 mjera svih mreža ispod 71%. Nakon dotreniravanja, sve mreže postižu znatno bolje rezultate, pri čemu SCNN-VGG16 ostvaruje najvišu ukupnu F1 mjeru od 97,26%, a SCNN-ERFNet najnižu, ali još uvijek visoku vrijednost od 96,23%. U svim kategorijama, mreže bilježe iznimno visoke performanse, osobito u uvjetima poput "tunela," gdje sve mreže dosežu ili vrlo blizu 100%. Kategorija "dan-kiša" ostaje najzahtjevnija, gdje SCNN-ERFNet postiže niže rezultate u usporedbi s ostalim mrežama, ali i dalje bilježi značajan napredak nakon dotreniravanja. Općenito, dotreniravanje značajno poboljšava performanse svih modela u svim uvjetima, što potvrđuje vrijednost dodatnog prilagođavanja modela na vlastitom skupu podataka za postizanje visoke točnosti u detekciji linija vozne trake.

#### **4.4.3 Analiza rezultata modela treniranih na TuSimple skupu podataka**

Prije evaluacije rezultata modela treniranih na TuSimple skupu podataka, potrebno je precizno definirati parametre evaluacije kako bi se osiguralo pravilno čitanje podataka. S obzirom da repozitoriji [36] ne konvertira TuSimple podatkovni skup, kao što je slučaj kod LLAMAS podatkovnog skupa, vlastiti skup se mora prilagoditi TuSimple zapisu podataka. Prilagodba vlastitog skupa podataka TuSimple zapisu omogućena je pomoću prilagođenih konvertera preuzetih iz rada [38], dok su segmentacijske maske generirane prema uputama iz direktorija [36]. S obzirom na ovu razliku prilikom testiranja mreža također će se koristiti i drugačije metrike za evaluaciju modela. U ovom slučaju nije moguće analizirati rezultate po različitim uvjetima.

Za evaluaciju točnosti modela korišten je prilagođeni vlastiti skup kao testni skup podataka. Kako bi se postigla precizna evaluacija, kao glavna metrika koristi se točnost (*engl.* Accuracy),

koja je zasnovana na usporedbi predviđenih točaka linija vozne trake s referentnim (GT) točkama. Kada je udaljenost između predviđene i referentne točke unutar postavljenog praga od 30 piksela, točka se klasificira kao TP. Dodatne metrike uključuju omjer lažno pozitivnih (*engl.* False Positive Rate – FPR) i omjer lažno negativnih primjera (*engl.* False Negative Rate – FNR). FPR prikazuje udio pogrešno predviđenih linija u odnosu na ukupni broj predviđenih linija, dok FNR ukazuje na udio neprepoznatih referentnih linija u odnosu na sve stvarne linije. Prag IoU za evaluaciju postavljen je na 0,5. Izračun točnosti modela zasnovan je na omjeru ispravno predviđenih točaka i ukupnog broja točaka unutar okvira isječka, prema formuli:

$$Točnost = \frac{\sum_{clips} c_{clips}}{\sum_{clips} s_{clips}} \quad (4-4)$$

gdje je  $c_{clips}$  broj točno predviđenih točaka u okviru isječka, a  $s_{clips}$  ukupan broj točaka unutar tog okvira. FPR se izračunava kao:

$$FPR = \frac{F_{pred}}{N_{pred}} \quad (4-5)$$

gdje je  $F_{pred}$  broj pogrešno predviđenih linija, a  $N_{pred}$  označava ukupan broj predviđenih linija. Slično tome, FNR se računa prema:

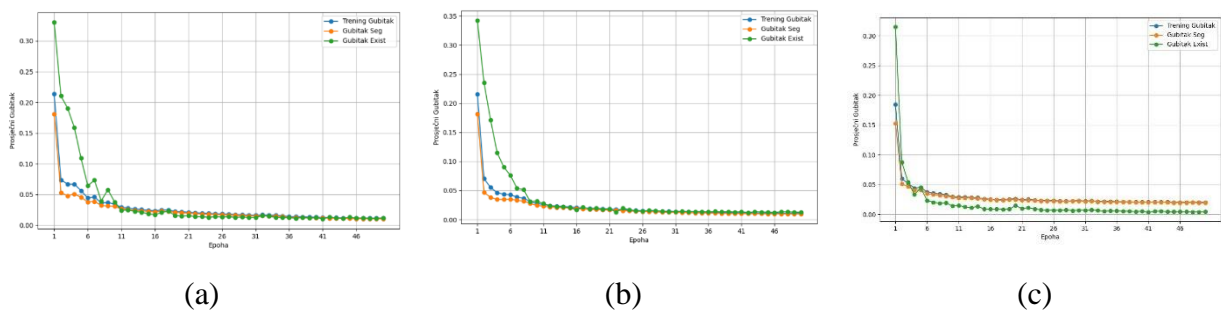
$$FNR = \frac{M_{pred}}{N_{gt}} \quad (4-6)$$

gdje je  $M_{pred}$  označava broj stvarnih linija koje model nije prepoznao, dok  $N_{gt}$  označava ukupan broj stvarnih linija. Rezultati evaluacije modela na vlastitom testnom skupu prikazani su u tablici 4.5. Prije dotreniravanja, mreže u prosjeku postižu točnost od 58,31%, što ukazuje na ograničenu sposobnost modela da precizno prepozna linije vozne trake u različitim uvjetima. Najslabiji rezultat pokazuje SCNN-VGG16, dok najbolji rezultat ostvaruje SCNN-ResNet34.

*Tablica 4.5 Rezultati evaluacije modela treniranih na podatkovnom skupu TuSimple i testiranih na testnom skupu vlastitog podatkovnog skupa*

Mreža	Dotrenirana	FPR	FNR	Točnost [%]
SCNN-VGG16	Ne	0,73	0,90	54,94
	Da	0,72	0,99	35,10
SCNN-RESNET34	Ne	0,70	0,83	62,28
	Da	0,70	0,98	35,33
SCNN-ERFNET	Ne	0,70	0,85	57,70
	Da	0,72	0,99	34,01

Nakon dotreniravanja, rezultati su značajno pogoršali, te mreže sada u prosjeku postižu točnost od 34,81%. Najslabiji rezultat nakon dotreniravanja pokazuje SCNN-ERFNet, dok najbolji rezultat postiže SCNN-ResNet34. Treniranje modela odvijalo se kroz 50 epoha, zadržavajući isti broj epoha kao i u izvornom repozitoriju kako bi se održala dosljednost i izbjegle bilo kakve promjene u konfiguraciji mreže. Na slici 4.22 prikazani su grafovi treniranja ovih modela po epohama. U sva tri grafa su prikazani trening gubitak, gubitak segmentacije i gubitak postojanja linija kroz 50 epoha. U početnim epohama svi gubici bilježe nagli pad koji može biti problematičan. Može ukazivati na previsoku stopu učenja, što dovodi do nestabilnosti, ili na prekomjerno prilagođavanje —gdje model prebrzo nauči specifičnosti trening podataka i ne generalizira dobro na nove primjere.



Slika 4.22 Prosječan gubitak po epohi tijekom dodatnog treniranja (a) VGG16 model (b) ResNet34 model (c) ERFNet model

Na slici 4.23 prikazana su predviđanja modela SCNN-VGG16 treniranog na TuSimple skupu podataka, uspoređena s označenim linijama u vlastitom podatkovnom skupu. Oznake na slikama odgovaraju onima u podpoglavlju 4.4.1. Sa slike 4.23 (a) (c) i (d) može se primijetiti kako mreža detektira linije unutar trenutne vozne trake, a ne i za cijelo vozno područje. Pod (b) mreža ne detektira nikakve linije. Nakon dotreniravanja SCNN-VGG16 mreže na vlastitom skupu podataka točnost opada za 19,4%. Prema tablici 4.5 to je najmanji pad točnosti kod izabranih modela.



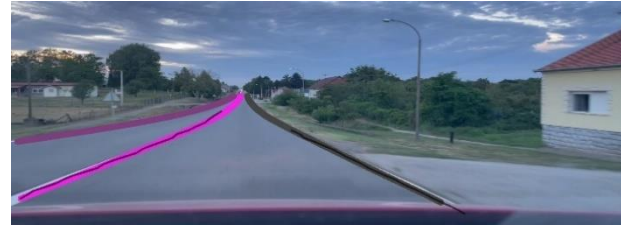
(a)



(b)



(c)



(d)

*Slika 4.23 Prikaz predviđenih linija SCNN-VGG16 modela treniranog na TuSimple podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Usporedbom slike 4.23 koja prikazuje predviđanja mreže prije dotreniravanja i slike 4.24 koja prikazuje predviđanja mreže nakon dotreniravanja vidljivo je da je mreža pronašla iste linije. Unatoč tome što točnost opada nakon dodatnog treniranja VGG16 mreže za 19,84% linije na slikama prikazanih pod 4.23 su grbave i krivudave dok su iste te linije predviđene istim tim modelom nakon dodatnog treniranja na trening skupu novokreiranog skupa podataka puno ravnije.



(a)



(b)



(c)



(d)

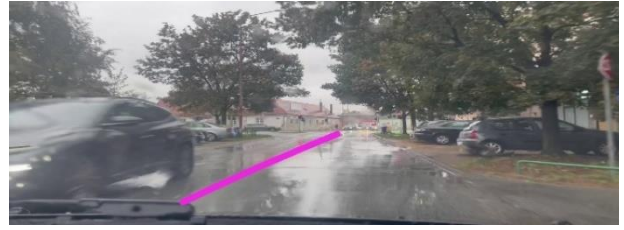
*Slika 4.24 Prikaz predviđenih linija SCNN-VGG16 modela treniranog na TuSimple podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Na slici 4.25 prikazana su predviđanja modela SCNN-ResNet34, pri čemu je jasno vidljivo da detekcija linija vozne trake daje slične rezultate kao i kod SCNN-VGG16 modela. Slika 4.25 je skoro identična slikama 2.23 i 2.24. Kod ovog modela rezultati prije dodatnog treniranja su veći u odnosu na preostala dva modela.





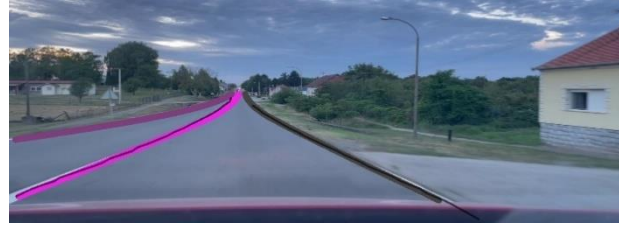
(a)



(b)



(c)



(d)

*Slika 4.25 Prikaz predviđenih linija SCNN-RESNET34 modela treniranog na TuSimple podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Slika 4.26 prikazuje predviđanje ovog modela nakon dotreniravanja gdje vidi da nema promjene u usporedbi s istim tim modelom ali bez dotreniravanja. Slika 4.27 prikazuje rezultate predviđanja modela SCNN-ERFNet. Za razliku od prethodna dva modela okvir (b) detektira srednju označenu liniju i desnu nepostojeću liniju vozne trake.



(a)



(b)



(c)



(d)

*Slika 4.26 Prikaz predviđenih linija SCNN-RESNET modela treniranog na TuSimple podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

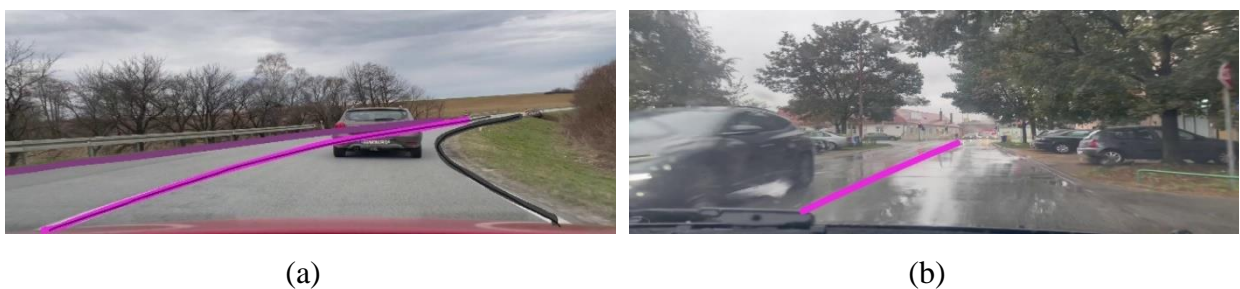


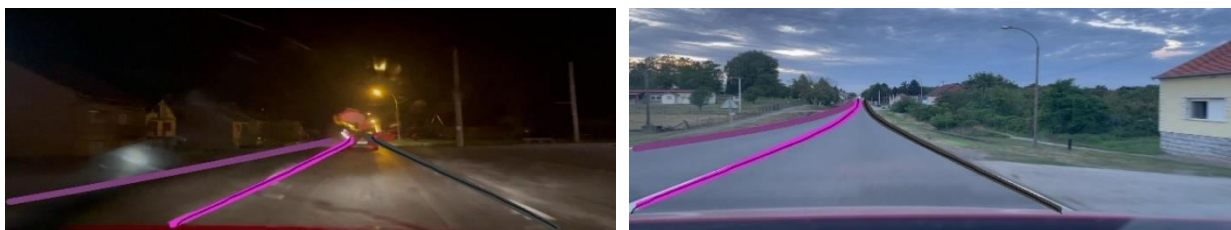
Desna linija nije kratka ili nesigurna kao i u primjerima kada je bila detektirana na modelima koji su trenirani na drugim podatkovnim skupovima. Kao i kod slika 4.25 i 4.23 linije označene na slikama prije nego što je model dodatno treniran su neravnomjerno nacrtane. Nakon dotreniravanja modela na trening skupu vlastitog podatkovnog skupa linije su vidno glađe i ravnomjernije nacrtane što je vidljivo na slici 4.27.



*Slika 4.27 Prikaz predviđenih linija SCNN-ERFNET modela treniranog na TuSimple podatkovnom skupu i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Nakon dotreniravanja modela SCNN-ERFNet također dolazi do pada točnosti. Slika 4.28 prikazuje predviđanja dotreniranog modela. Ako se uspoređi sa slikom 4.27 koja prikazuje predviđanja prije dotreniravanja može se uočiti razlika u okviru (b) gdje detektira srednju liniju na slici 4.24 dok ne detektira ništa na slici 4.25. Pad točnosti nakon dotreniravanja i povećanje FNR može se pripisati razlici između rezolucije slika unutar podatkovnih skupova korištenih za treniranje i testiranje. Mreže su trenirane na TuSimple podatkovnom skupu pa je samim time rezolucija slika bila  $1280 \times 720$  piksela.





(c)

(d)

*Slika 4.28 Prikaz predviđenih linija SCNN-ERFNET modela treniranog na TuSimple podatkovnom skupu, dotreniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

Model je dotreniran na vlastitom skupu podataka gdje je rezolucija slika  $1640 \times 590$  piksela, što je moglo negativno utjecati na sposobnost generalizacije modela. Također, postupak generiranja segmentacijskih oznaka u repozitoriju [35] nije detaljno objašnjen, moguće je nagađati iz komentara drugih korisnika o metodi njihova generiranja pa je moguće da oznake nisu generirane dosljedno. To dodatno otežava preciznu detekciju linija, međutim, konverteri između formata su provjereni i točni, pa prijepis iz formata u format nije uzrok problema.

#### 4.4.3 Treniranje i testiranje mreža na vlastitom skupu podataka

U ovom dijelu uzeta su tri prethodno navedena modela bez da su prethodno trenirana na nekom postojećem skupu podataka i trenirana na vlastitom skupu podataka, a zatim su testirani. Tablica 4.6 prikazuje rezultate dobivene neposredno nakon završetka treniranja mreža na trening skupu vlastitog skupa podataka.

*Tablica 4.6 Rezultati evaluacije modela na trening skupu vlastitog podatkovnog skupa*

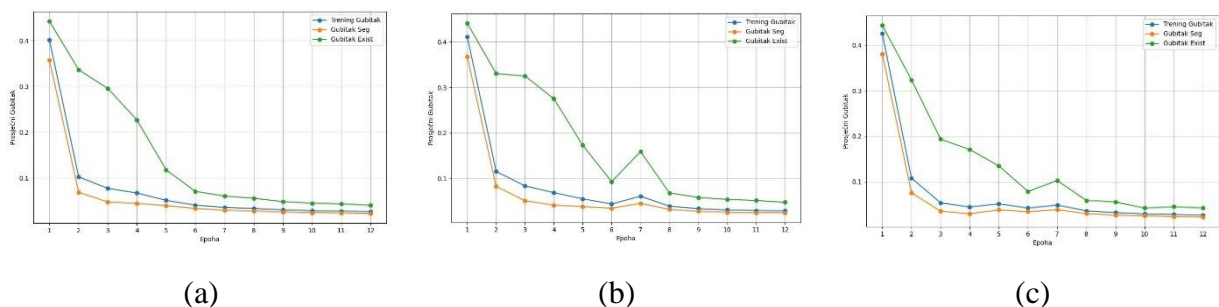
Mreža	Mjere	Dan [%]	Dan-kisa [%]	Noć [%]	Noć-kiša [%]	Sumrak [%]	Tunel [%]	Ukupni rezultati [%]
SCNN-VGG16	Preciznost	93,78	97,78	99,33	99,03	96,71	94,44	98,33
	Odziv	91,68	89,52	99,67	96,26	95,83	89,47	97,47
	F1	92,17	93,48	99,50	97,63	96,27	91,89	97,92
SCNN-RESNET34	Preciznost	97,93	97,94	99,67	99,06	98,59	100	98,59
	Odziv	88,52	89,62	100	100	94,59	92,11	94,83
	F1	92,98	93,57	99,83	99,53	96,55	95,83	96,63
SCNN-ERFNET	Preciznost	90,66	85,71	100	96,18	100	100	94,88
	Odziv	88,52	80,00	99,67	97,14	94,59	97,37	91,83
	F1	89,58	82,72	99,83	98,56	97,22	98,68	93,44

Ovi rezultati odražavaju sposobnost modela da nauči obrasce i prilagodi svoje parametre tijekom procesa treniranja, no ne pružaju potpunu sliku o njegovoj generalizaciji na nepoznate podatke. Vidljivo je da mreža VGG16 daje najbolje rezultate nakon treniranja dok ERFNet daje najlošije. S druge strane u tablici 4.7 prikazani su rezultati dobiveni nakon testiranjem modela na testnom skupu vlastitog skupa podataka. Ovi rezultati služe za procjenu sposobnosti modela da generalizira naučeno znanje i precizno obavlja zadatak na podacima koji nisu korišteni tijekom procesa treniranja, čime se procjenjuje njegova stvarna praktična primjenjivost.

Tablica 4.7 Rezultati evaluacije modela treniranih na trening skupu vlastitog podatkovnog skupa i testiranih na testnom skupu vlastitog podatkovnog skupa

Mreža	Mjere	Dan [%]	Dan-kisa [%]	Noć [%]	Noć-kiša [%]	Sumrak [%]	Tunel [%]	Ukupni rezultati [%]
SCNN-VGG16	Preciznost	93,70	97,78	99,55	97,70	96,83	93,94	96,53
	Odziv	90,68	91,67	99,55	97,70	96,83	93,94	94,95
	F1	92,17	94,62	99,55	97,70	96,83	93,94	95,74
SCNN-RESNET34	Preciznost	92,19	93,48	99,12	98,85	96,83	96,88	95,84
	Odziv	88,89	89,58	99,55	98,85	96,83	93,94	94,27
	F1	90,51	91,49	99,33	98,85	96,83	93,94	95,05
SCNN-ERFNET	Preciznost	91,18	86,36	99,55	97,70	95,24	100	95,01
	Odziv	88,89	79,17	99,55	97,70	95,24	96,97	93,45
	F1	90,02	82,61	99,55	97,70	95,24	98,46	94,22

Za treniranje modela na skupu trening skupu vlastitog podatkovnog skupa preuzete su skripte korištene za treniranje modela na CULane podatkovnom skupu. Upravo zbog toga treniranje se odvijalo kroz 12 epoha. Na slici 4.2 modeli pokazuju nagli pad gubitaka u prvim epohama, što ukazuje na brzo učenje. VGG, pod (a) model ima najstabilnije gubitke, dok ERFNet prikazana na grafu pod (c) i ResNet pod (b) pokazuju oscilacije u gubitku postojanja linija oko 6. i 7. epohe. Nakon 8. epohe, gubici se stabiliziraju za sve modele i ostaju niski do kraja treniranja, što ukazuje na konvergenciju.



Slika 4.29 Prosječan gubitak po epohi tijekom treniranja (a) VGG16 model (B) ResNet34 model (c) ERFNet model

Nakon završetka treniranja, modeli su testirani na testnom skupu vlastitog skupa podataka kako bi se objektivno procijenila njihova učinkovitost i sposobnost generalizacije na novim neviđenim primjerima. Na slici 4.30 prikazana su predviđanja VGG16 modela treniranog i testiranog na vlastitom skupu podataka.

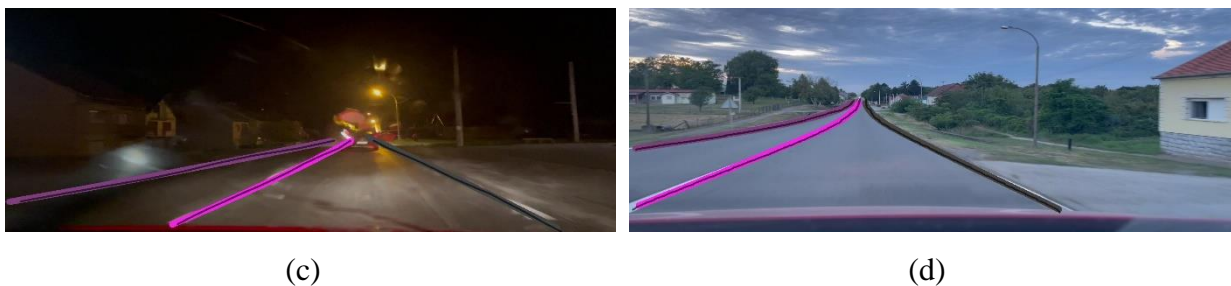


*Slika 4.30 Prikaz predviđenih linija SCNN-VGG16 modela treniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak*

U sva četiri okvira model uspješno detektira sve označene linije na cesti, čime pokazuje visoku preciznost u prepoznavanju linija vozne trake. Bez obzira na različite uvjete prikazane u svakom okviru, model dosljedno prepoznaje i prati linije vozne trake. Na slici 4.31 prikazana su predviđanja SCNN-ResNet34 modela treniranog i testiranog na vlastitom skupu podataka. U sva četiri okvira model uspješno detektira sve označene linije na cesti, čime pokazuje visoku preciznost u prepoznavanju linija vozne trake. Na slici 4.32 prikazana su predviđanja SCNN-ERFNet modela treniranog i testiranog na vlastitom skupu podataka. U sva četiri okvira model uspješno detektira sve označene linije na cesti, čime pokazuje visoku preciznost u prepoznavanju linija vozne trake.

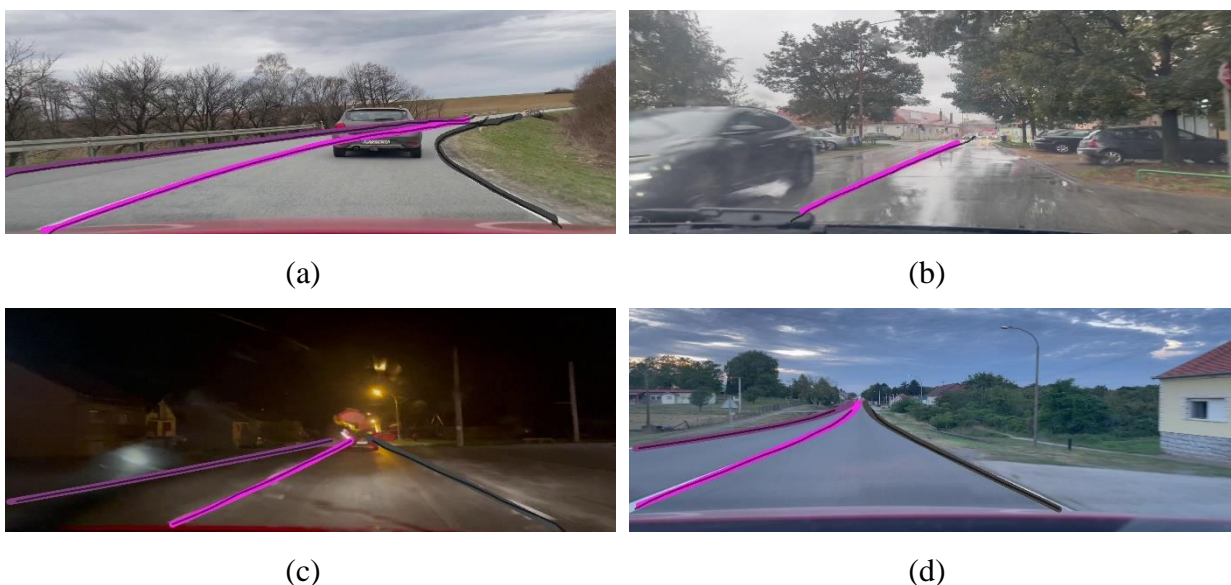






Slika 4.31 Prikaz predviđenih linija SCNN-ResNet34 modela treniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak

Na temelju tablice 4.7 s rezultatima modela treniranih i testiranih na vlastitom skupu podataka, može se zaključiti da SCNN-VGG16 postiže ukupnu F1 mjeru od 95,74%, što je najviši rezultat među promatranim modelima. Najbolji rezultat model postiže u uvjetima "noć" (99,55%) i "noć-kiša" (97,71%), što pokazuje robusnost u uvjetima smanjene vidljivosti.



Slika 4.32 Prikaz predviđenih linija SCNN-ERFNet modela treniranog i testiranog na novokreiranom skupu podataka (a) ruralno područje, krivina, djelomična naoblaka, auto zaklanja liniju (b) loša vidljivost, kišni dan, slaba vidljivost jedine linije (c) noćna vožnja po kiši, slaba vidljivost kroz vjetrobransko staklo, mutna slika (d) ravna cesta bez prometa, jako dobra vidljivost, sumrak

Rezultati su vrlo ujednačeni u svim kategorijama, pri čemu su i uvjeti "dan" i "tunel" zabilježeni s visokim vrijednostima F1 mjere (92,17% i 93,94%). SCNN-ResNet34 postiže ukupnu F1 mjeru od 95,05%, što je nešto niže od SCNN-VGG16, ali i dalje vrlo visoko. Najbolji rezultat bilježi u uvjetima "noć" (99,33%) i "noć-kiša" (98,85%), što također pokazuje solidnu prilagodbu modela u uvjetima s manjom osvijetljenošću. Rezultati u kategorijama "dan" (90,51%) i "dan-

kiša" (91,49%) su nešto niži, što sugerira određene izazove u detekciji u dnevnim uvjetima ili pri promjenjivim vremenskim uvjetima. SCNN-ERFNet postiže najnižu ukupnu F1 mjeru među modelima (94,22%), ali i dalje vrlo visoku. Najbolje rezultate ostvaruje u uvjetima "noć" (99,55%) i "tunel" (98,46%), što pokazuje da model dobro funkcionira pri slabijem svjetlu i u kontroliranim osvjetljenjima. Najslabiji rezultat zabilježen je za kategoriju "dan-kiša" (82,61%), što ukazuje na poteškoće u prepoznavanju linija vozne trake u uvjetima promjenjivog osvjetljenja i refleksija zbog kiše. Sve mreže postižu vrlo visoke ukupne F1 mjere, s najboljim performansama u uvjetima "noć" i "noć-kiša," što pokazuje robusnost u uvjetima s manjom osvjetljenošću. Najlošiji rezultati zabilježeni su za kategoriju "dan-kiša," osobito kod SCNN-ERFNet modela, što može ukazivati na to da modeli općenito teže prilagodbi u uvjetima promjenjivog i reflektivnog osvjetljenja.

#### **4.5 Završni osvrt na rezultate dobivene testiranjem i evaluacijom različitih metoda za detekciju linija vozne trake na različitim skupovima podataka**

Rezultati modela treniranih na različitim skupovima podataka pokazuju važnost dotreniravanja za postizanje bolje točnosti u detekciji linija vozne trake. Dotrenirani modeli općenito bilježe značajna poboljšanja, posebno u složenijim uvjetima poput „dan-kiša“ i „noć-kiša,“ gdje postižu znatno višu preciznost. Performanse modela snažno ovise o karakteristikama skupa podataka. Na primjer, dotrenirani modeli na CULane skupu, uključujući SCNN-VGG16, SCNN-ResNet34 i SCNN-ERFNet, postižu F1 mjere u rasponu od 96,18% do 97,26%, čime se pokazuju vrlo prilagođenima uvjetima u ovom skupu. Slično tome, SCNN-VGG16 postiže najbolju F1 mjeru od 97,26% na LLAMAS skupu, dok ostali modeli bilježe tek nešto niže rezultate. Ovi podaci potvrđuju sposobnost modela da prepoznaju linije u uvjetima specifičnim za ove skupove. Ako usporedimo rezultate dobivene prije dotreniravanja modela treniranih na CULane i LLAMAS podatkovnim skupovima možemo zaključiti kako je novokreirani vlastiti skup podataka sličniji LLAMAS.

Primjena modela na TuSimple podacima, međutim, pokazuje izazove u prilagodbi modela na ovaj format. Točnost dotreniranih modela ovdje opada na oko 35%, a FPR i FNR ostaju relativno visoki, što upućuje na probleme generalizacije. Razlike u rezoluciji i segmentacijskim oznakama u TuSimple podacima mogu pridonijeti ovoj slabijoj prilagodbi.

Na vlastitom skupu podataka, dotrenirani modeli postižu visoke rezultate, s F1 mjerama od 94,22% do 95,74%, što pokazuje njihovu sposobnost da točno detektiraju linije vozne trake u realnim uvjetima. Modeli kada su trenirani na trening skupu vlastite baze podataka daju malo

lošije, ali i dalje jako slične rezultate kao i modeli trenirani na CULane i LLAMAS bazama podataka. Rezultati osciliraju za oko 3%, rezultati F1 mjere dobiveni na mrežama treniranih na CULane su u rasponu od 96,18% do 97,26%, kod mreža treniranih na LLAMAS 96,23%-97,25%, dok za mreže trenirane na vlastitoj bazi podataka rezultati se nalaze u rasponu od 94,22%-95,74%.

Pojedinačno gledano, SCNN-VGG16 postiže najviše rezultate među dotreniranim modelima, osobito u uvjetima s niskom osvjetljenošću. SCNN-ResNet34 također bilježi visoku F1 mjeru u izazovnim uvjetima kao što je „noć-kiša,“ no pokazuje slabije rezultate na TuSimple skupu. SCNN-ERFNet pokazuje dobre performanse u noćnim uvjetima i uvjetima u tunelu, no nešto slabije rezultate ostvaruje u uvjetima „dan-kiša“, što može biti posljedica promjena u osvjetljenju i refleksija. Mogući razlog dobrih rezultata modela treniranih na vlastitom skupu podataka može biti njezina veličina i ograničenost scenarija. Novokreirani skup podataka sastoji se od 1500 slika snimljenih u različitim uvjetima, ali zbog relativno malog opsega i ponavljanja scenarija, nema složenih situacija kao što su kružni tokovi, raskrižja ili prometne gužve. Zbog toga model lakše može prepoznati i naučiti obrasce pozicija linija, što dovodi do boljih rezultata na testiranju unutar ovog ograničenog skupa podataka.

Ukupno gledano, rezultati potvrđuju važnost dotreniravanja i prilagodbe modela na specifične skupove podataka za bolju točnost detekcije linija vozne trake. Modeli trenirani na skupovima poput CULane i LLAMAS postižu visoku točnost, dok prilagodba različitim formatima, poput TuSimple, ostaje izazov zbog razlika u strukturi podataka. Ova analiza pokazuje da je važno odabrati odgovarajući skup podataka i primijeniti dotreniranje kako bi se modeli optimalno prilagodili stvarnim uvjetima vožnje.

## 5. ZAKLJUČAK

Detekcija voznih traka ključna je komponenta u sustavima autonomne vožnje koja omogućava vozilima da precizno prepoznaju i slijede prometne trake. Korištenjem modela zasnovanih na računalnom vidu i strojnom učenju, sustavi postaju sposobni za precizno prepoznavanje traka u raznim prometnim uvjetima. Ovaj diplomski rad obuhvaća razvoj aplikacije za označavanje linija vozne trake, kreiranje vlastitog podatkovnog skupa te detaljnu evaluaciju modela za njihovu detekciju zasnovanih na strojnom učenju. Razvijeni alat omogućava jednostavno i precizno označavanje linija voznih traka na slikama, čime omogućava kreiranje kvalitetnog podatkovnog skupa za treniranje i evaluaciju modela. Ključne funkcionalnosti uključuju kreiranje i upravljanje projektima, intuitivno rukovanje galerijama slika, ručno označavanje linija te spremanje označenih podataka u prikladne formate za daljnju analizu. Uspješnost alata testirana je kroz različite funkcije te je potvrđena njegova učinkovitost, što čini solidnu osnovu za primjenu u sustavima autonomnih vozila. Za potrebe rada, novokreirani skup podataka snimljen je mobilnim uređajem na lokalnim prometnicama, a slike su zatim uokvirene i označene u alatu kako bi se dobio skup podataka prilagođen realnim uvjetima. Evaluacija modela provedena je na mrežama treniranim na različitim podatkovnim skupovima: CULane, TuSimple, i LLAMAS, te testiranim na testnom skupu novokreiranog vlastitog skupa podataka. Nakon toga evaluacija je provedena nad modelima treniranim na trening skupu iz vlastitog podatkovnog skupa. Analizom performansi modela istražen je utjecaj dotreniravanja modela na njihovu učinkovitost u detekciji voznih linija u realnim uvjetima. Dotreniravanje se pokazalo ključnim za prilagodbu modela specifičnim i realnim uvjetima u vlastitom skupu podataka, što je rezultiralo značajnim poboljšanjem performansi u izazovnim uvjetima. Rezultati su pokazali da modeli postižu visoke performanse na skupovima podataka na kojima su trenirani. Modeli kao što su SCNN-VGG16 i SCNN-ResNet34 postigli su zavidne rezultate u scenarijima sa slabijim osvjetljenjem, dok je SCNN-ERFNet pokazao pouzdanost u noćnim uvjetima i tunelima. Međutim, prilagodba modela na različite strukture podataka, kao što je TuSimple, pokazala se izazovnom, što upućuje na potrebu za dodatnim prilagođavanjem i usklađivanjem podataka pri prelasku na drugačije formate.

Zaključno, dotreniranje modela na vlastitom skupu podataka poboljšava njihovu točnost i prilagodljivost u specifičnim uvjetima vožnje. Rezultati također ukazuju na potrebu za razvojem dodatnih europskih podatkovnih skupova s raznovrsnijim prometnim scenarijima, budući da su postojeći veliki skupovi podataka uglavnom snimani u Kini ili SAD-u na širokim i prometnim cestama. Time bi se omogućila veća prilagodba modela na uvjete koji su specifični za europske prometnice i infrastrukturne karakteristike, poput manjih prometnica, zavoja i užih voznih traka.



## LITERATURA

- [1] C. Low, H. Zamzuri, S. A. Mazlan, "Simple robust road lane detection algorithm", Proc. Int. Conf. Intell. Adv. Syst., 2014
- [2] D. Vajak, M. Vranješ, R. Grbić, D. Vranješ, „Recent Advances in Vision-Based Lane Detection, Solutions for Automotive Applications“, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek
- [3] K. Kluge, S. Lakshmanan. „A Deformable-Template Approach to Lane Detection.“ U Intelligent Vehicles '95 Symposium, Sep 1995
- [4] Y. Xing, C. Lv, Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision, IEEE, Journal of Automatica Sinica, vol. 5, no. 3, Svibanj 2018
- [5] A. Mammeri, A. Boukerche, and G. Lu, "Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter," in Proc. ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst., 2014.
- [6] Vajak D, Vranješ M, Grbić R, Tešlić N. „A Rethinking of Real-Time Computer Vision-Based Lane Detection“ In: 2021 IEEE 11th International Conference on Consumer Electronics (ICCE-Berlin); 2021.
- [7] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding.", Kina, 2017. // Dostupno na: <https://arxiv.org/abs/1712.06080>
- [8] J. Liu, L. Lou, D. Huang, Y. Zheng, W. Xia, "Lane Detection Based on Straight Line Model and K-Means Clustering", IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), pp. 527-532, May 2018.
- [9] GeeksforGeeks. K-means Clustering Introduction [Internet]. GeeksforGeeks; [datum pristupa: 4. studenoga 2024]. Dostupno na: <https://www.geeksforgeeks.org/k-means-clustering-introduction/>
- [10] Y. Huang, Y. Li, X. Hu, W. Ci, "Lane Detection Based on Inverse Perspective Transformation and Kalman Filter", KSII Transactions on Internet and Information Systems, vol. 12, iss. 2, pp. 643-661, February 2018.
- [11] M. Aly, "Real time Detection of Lane Markers in Urban Streets," in Proc. of IEEE Intelligent Vehicles Symposium, pp. 7-12, June 4-6, 2008, Dostupno na: <https://ieeexplore.ieee.org/document/4621152>

- [12] Y. Xing, C. Lv, H. Wang, D. Cao and E. Velenis, "Dynamic integration and online evaluation of vision-based lane detection algorithms", IET Intelligent Transport Systems, vol. 13, iss. 1, pp. 55-62, Sijećanj 2019.
- [13] R. Kumar, A. Gupta, i V. Sharma, "Efficient techniques for scaling up machine learning models on big data analytics: A comprehensive review," objavljeno u Journal of Big Data, vol. 8, br. 1, 2021. Dostupno na: <https://link.springer.com/content/pdf/10.1186/s40537-021-00444-8.pdf> (Datum pristupa: 28. lipanj 2024)
- [14] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, 2016.
- [15] K. O'Shea, R. Nash, "An Introduction to Convolutional Neural Networks," 2015. Dostupno na: <https://arxiv.org/pdf/1511.08458> (Datum pristupa 29. lipanj 2024.)
- [16] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards End-to-End Lane Detection: an Instance Segmentation Approach." Belgija, 2018
- [17] TuSimple benchmark, "Tusimple-benchmark/doc/lane\_detection-TuSimple/tusimple benchmark GitHub." [https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane\\_detection](https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection)
- [18] T. Zheng et al., "RESA: Recurrent Feature-Shift Aggregator for Lane Detection." , Kina, 2021.
- [19] Y.Y. Ye, X.L. Hao, H.J. Chen, "Lane detection method based on lane structural analysis and CNNs", IET Intelligent Transport Systems, vol. 12, iss. 6, pp. 513-520, Kolovoz, 2018.
- [20] V. John, Z. Liu, S. Mita, C. Guo, K. Kidono, "Real-time road surface and semantic lane estimation using deep features", Signal, Image and Video Processing, vol. 12, iss. 6, pp. 1133-1140, Listopad, 2018.
- [21] Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In CVPR
- [22] K. Behrendt and R. Soussan, "Unsupervised Labeled Lane Markers Using Maps," objavljeno u 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Južna Koreja, 2019., str. 832–839
- [23] "The KITTI Vision Benchmark Suite." [http://www.cvlibs.net/datasets/kitti/eval\\_road.php](http://www.cvlibs.net/datasets/kitti/eval_road.php), (Datum pristupa: 8.8.2024.)
- [24] F. Yu, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning." , Sjedinjene Američke Države, 2020., (Datum pristupa: 8.8.2024.)

- [25] A. Hanbury, „A survey of methods for image annotation“, J. Vis. Lang. Comput., sv. 19, izd. 5, str. 617–627, lis. 2008, doi: 10.1016/j.jvlc.2008.01.002. (Datum pristupa: 8.8.2024.)
- [26] LabelMe, GitHub repozitoriji, <https://github.com/labelmeai/labelme> (Datum pristupa: 8.8.2024.)
- [27] Visual Geometry Group, Oxford, UK Dostupno na: <https://www.robots.ox.ac.uk/~vgg/software/via/> (Datum pristupa: 8.8.2024.)
- [28] Labelbox, dostupno na: <https://docs.labelbox.com/docs/what-is-a-project> (Datum pristupa: 8.8.2024.)
- [29] Cvat, dostupno na: [https://docs.cvat.ai/docs/getting\\_started/workflow-org/](https://docs.cvat.ai/docs/getting_started/workflow-org/) (Datum pristupa: 8.8.2024.)
- [30] Cvat, GitHub repozitoriji, dostupno na: <https://github.com/cvat-ai/cvat> (Datum pristupa: 8.8.2024.)
- [31] Deepen AI, dostupno na: <https://www.deepen.ai/image-annotation> (Datum pristupa: 8.8.2024.)
- [32] G. Lin, Building Simple Annotation Tools. University of California, San Diego, 2016.
- [33] Qt Group, “Qt Documentation | Home” Dostupno na: <https://doc.qt.io/> (Datum pristupa : 8. kolovoz 2024.)
- [34] Qt Group, “Qt Designer Manual,” Dostupno na: <https://doc.qt.io/qt-6/qtdesigner-manual.html> (Datum pristupa: 8. kolovoz 2024.)
- [35] SciPy, „CubicSpline“ Dostupno na : <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.CubicSpline.html> (Datum pristupa 7.kolovoz 2024)
- [36] PytorchAutoDrive, GitHub repozitorij ” <https://github.com/voldemortX/pytorch-auto-drive> (pristup ostvaren 5.9.2024.)
- [37] SegLabelGenerate, Github repozitoriji [https://github.com/XingangPan/seg\\_label\\_generate](https://github.com/XingangPan/seg_label_generate) (pristup ostvaren: 15.10.2024)
- [38] Jukić G. Analiza i vrednovanje metoda za prepoznavanje linija vozne trake zasnovanih na strojnom učenju na različitim skupovima podataka [Diplomski rad]. Osijek: Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek; 2022 [Datum pristupa: 23.8.2024.] Dostupno na: <https://urn.nsk.hr/urn:nbn:hr:200:781486>

## SAŽETAK

U ovom diplomskom radu izrađen je vlastiti alat za označavanje linija vozne trake, kao i usporedba različitih modela strojnog učenja za detekciju linija vozne trake. Cilj rada je omogućiti precizno i intuitivno označavanje linija voznih traka na slikama u svrhu generiranja kvalitetnog skupa podataka koji se koristi za treniranje i evaluaciju modela zasnovanih na dubokom učenju. Razvijeni alat omogućuje kreiranje i upravljanje projektima, intuitivno rukovanje galerijom slika, ručno označavanje linija i spremanje podataka u formate pogodne za daljnju analizu. Nakon definiranja i testiranja funkcionalnosti aplikacije, fokus je usmjeren na izradu vlastitog skupa podataka koji uključuje slike snimljene u različitim vremenskim i prometnim uvjetima. Na temelju prikupljenog skupa podataka evaluirani su modeli trenirani na tri različita skupova podataka (CULane, TuSimple i LLAMAS), kao i modeli dotrenirani na vlastitom skupu. Analiza je uključivala mjerenje ključnih metrika poput F1 mjere, preciznosti i odziva, što je omogućilo objektivnu usporedbu performansi između modela u različitim scenarijima, uključujući dan, noć, kišu, tunel i sumrak. Rezultati pokazuju značajna poboljšanja modela nakon dotreniranja na novokreiranom skupu podataka, što potvrđuje važnost prilagodbe modela specifičnim uvjetima primjene.

# **LANE MARKING TOOL AND COMPARISON OF MACHINE LEARNING BASED MODELS FOR LANE DETECTION**

## **ABSTRACT**

In this work, a custom tool for lane marking has been developed, along with a comparison of various machine learning models for lane detection. The objective of this work is to enable precise and intuitive annotation of lane lines on images in order to generate a high-quality dataset used for the training and evaluation of deep learning-based models. The developed tool supports project creation and management, intuitive handling of image galleries, manual lane marking, and data saving in formats suitable for further analysis. After defining and testing the application's functionality, the focus was directed towards creating a custom dataset that includes images captured under various weather and traffic conditions. Based on the collected dataset, models trained on three different datasets (CULane, TuSimple, and LLAMAS) were evaluated, as well as models further fine-tuned to the custom dataset. The analysis included measuring key metrics such as F1 score, precision, and recall, allowing an objective comparison of the models' performance across different scenarios, including day, night, rain, tunnel, and twilight. The results demonstrate significant improvements in the models after fine-tuning to the newly created dataset, confirming the importance of adapting models to specific application conditions.

## **ŽIVOTOPIS**

Rea Križanac rođena 14. studenog 1999. godine u Požegi. Završetkom Prirodoslovno-matematičke gimnazije u Požegi 2018. godine, upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer elektrotehnika. 2022. godine uspješno završava preddiplomski studij informacijske i komunikacijske tehnologije te upisuje diplomski sveučilišni studij elektrotehnika, smjer mrežne tehnologije.

## **PRILOZI**

- P.4.1. direktoriji koji sadrži označenu bazu podataka kreiranu u sklopu ovog diplomskog rada (priloženu na DVD-u uz rad)