



UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET



ANDRIJA PERUNIČIĆ

**PROCJENA BRZINE VOZILA IZ VIDEO
SIGNALA KORISTEĆI DETEKTOR OBJEKATA
NA SLICI I REKURENTNE NEURALNE MREŽE**

– MASTER RAD –

PODGORICA, 22. FEBRUAR 2024.

PODACI I INFORMACIJE O MAGISTRANTU

Ime i prezime: **Andrija Peruničić**

Datum i mjesto rođenja: 03.08.1999. godine, Berane

Prethodno završene studije:

Elektrotehnički fakultet, osnovne akademske studije (180 ECTS kredita),
studijski program: Elektronika, telekomunikacije i računari, 2018.

INFORMACIJE O MAGISTARSKOM RADU

Elektrotehnički fakultet

Studijski program: Računari

Naslov rada: **Procjena brzine vozila iz video signala koristeći detektor objekata na slici i rekurentne neuralne mreže**

Mentor: Prof. dr Slobodan Đukanović

UDK, OCJENA I ODBRANA MAGISTARSKOG RADA

Datum prijave magistarskog rada: 03.07.2023.

Datum sjednice Vijeća na kojoj je prihvaćena tema: 22.09.2023.

Komisija za ocjenu/odbranu rada:

1. Prof. dr Vesna Popović - Bugarin, ETF Podgorica, predsjednica,
2. Prof. dr Slobodan Đukanović, ETF Podgorica, mentor
3. Doc. dr Miloš Brajović, ETF Podgorica, član

Datum odbrane: 27.02.2024.

Ime i prezime autora: Andrija Peruničić

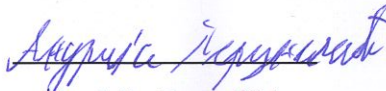
ETIČKA IZJAVA

U skladu sa članom 22 Zakona o akademskom integritetu i članom 18 Pravila studiranja na master studijama, pod krivičnom i materijalnom odgovornošću, izjavljujem da je master rad pod naslovom

"Procjena brzine vozila iz video signala koristeći detektor objekata na slici i rekurentne neuralne mreže"

moje originalno djelo.

Podnosilac izjave:


Andrija Peruničić

U Podgorici, dana godine 3. decembar 2023.

Sadržaj

Sažetak	1
Abstract	2
1 Uvod	3
1.1 Inteligentni transportni sistemi	4
1.2 Prethodna istraživanja	6
2 Rekurentne neuralne mreže	12
2.1 Osnovni principi rada rekurentnih neuralnih mreža	13
2.2 Treniranje rekurentnih neuralnih mreža	15
2.3 Optimizacija pomoću gradijentnog spusta	15
2.4 Gradijentni spust sa inercijom, Adam optimizacija	18
2.5 Regularizacija	20
2.6 Duboke rekurentne neuralne mreže	21
2.7 Long short-term memory (LSTM)	23
2.8 Gated recurrent unit (GRU)	25
3 Detekcija objekata na video snimcima	28
3.1 Poznati skupovi podataka i mjera kvaliteta algoritma	29
3.2 Tradicionalni detektori	29
3.3 Detektori zasnovani na dubokom učenju	31
3.4 Dvofazni detektori	32
3.5 Jednofazni detektori	35

4 Procjena brzine vozila	40
4.1 Predložena metoda	41
4.2 Detekcija vozila i ekstrakcija potrebnih podataka	43
4.3 Vremenska serija i rekurentna neuralna mreža	43
4.4 Skup podataka	44
4.5 Treniranje modela	47
5 Eksperimentalni rezultati	49
5.1 Postavka eksperimenta	50
5.2 Mjera kvaliteta	50
5.3 Postavke treniranja	51
5.4 Odabir mreže	51
5.5 Učenje sa sekvencama različitih dužina	52
5.6 Odabir karakteristika	53
5.7 Odabir sekvenci	56
5.8 Rezultati	58
Zaključak	60

Sažetak

Precizna i pouzdana procjena brzine vozila je važan dio saobraćajnog sistema, posebno kod sprovođenja zakonskih odredbi koje za cilj imaju poboljšanje bezbjednosti u saobraćaju. Visoka preciznost procjene brzine je ključna zbog posljedica koje nosi. Zbog toga se koriste skupi visoko-precizni uređaji za procjenu brzine. U cilju pristupačnije procjene brzine, noviji sistemi uključuju tehnologije kao što je kompjuterska vizija, gdje bi se težnja stavila na kamere kao primarne senzore za nadgledanje saobraćaja.

Trenutno dostupne metode za procjenu brzine vozila zasnovane na kompjuterskoj viziji zahtijevaju složene procese instalacije i kalibracije. Da bi pravilno funkcionisale, ove metode često zahtijevaju fiksne kamere i poznavanje specifičnih parametara kao što su ugao kamere i dimenzije puta ili vozila. Ovo ograničenje utiče na opštu primjenjivost i jednostavnost upotrebe ovih metoda.

U cilju prevazilaženja pomenutih ograničenja, u okviru ovog istraživanja je razvijena metoda za procjenu brzine vozila u potpunosti zasnovana na mašinskom učenju. Primjenom detektora objekata na slici, dobijaju se podaci o položaju i veličini vozila na slici. U cilju obrade sekvence ovih podataka dizajnirana je rekurentna neuralna mreža. Rekurentna mreža, koja posjeduje mogućnost da pamti i prati promjenu pozicije vozila tokom snimka, vrši konačnu procjenu brzine vozila.

Rezultati ovog istraživanja pokazuju da je moguće razviti robustnu i preciznu metodu za vizuelnu procjenu brzine vozila zasnovanu na dubokom učenju, koja će biti opšta i neće zahtijevati prilagođavanje pojedinačnim terenima. Takođe, ovu metodu je moguće primijeniti u realnom vremenu, gdje će se brzina vozila procjenjivati u svakom dijelu videa. Ovakav pristup omogućava efikasno praćenje brzine vozila u različitim situacijama i uslovima, doprinoseći tako unapređenju inteligentnih sistema za nadgledanje saobraćaja i optimizaciji saobraćajnih tokova.

Ključne riječi: procjena brzine vozila, rekurentne neuralne mreže, LSTM, detektor objekata na slici, YOLO

Abstract

Accurate and reliable assessment of vehicle speed is an important part of the traffic system, especially in the implementation of legal provisions aimed at improving traffic safety. High accuracy of speed estimation is crucial because of the consequences it carries. Therefore, expensive high-precision speed estimation devices are used. In order to make speed estimation more accessible, newer systems include technologies such as computer vision, where the focus would be on cameras as the primary sensors for traffic monitoring.

Currently available computer vision-based vehicle speed estimation methods require complex installation and calibration processes. To work properly, these methods often require fixed cameras and knowledge of specific parameters such as camera angle and road or vehicle dimensions. This limitation affects the general applicability and ease of use of these methods.

In order to overcome the mentioned limitations, within this research, a method for estimating vehicle speed based entirely on machine learning was developed. By applying the object detector, data is obtained about the position and size of the vehicle in the image. In order to process the sequence of these data, a recurrent neural network was designed. The recurrent network, which has the ability to remember and track the change in the position of the vehicle during the recording, makes a final estimate of the vehicle's speed.

The results of this research show that it is possible to develop a robust and accurate method for visual estimation of vehicle speed based on deep learning, which will be general and will not require adaptation to individual terrains. Also, this method can be applied in real time, where the speed of the vehicle can be estimated in each part of the video. This approach enables effective monitoring of vehicle speed in various situations and conditions, thus contributing to the improvement of intelligent traffic monitoring systems and optimization of traffic flows.

Keywords: vehicle speed estimation, recurrent neural networks, LSTM, real-time object detection, YOLO

Glava 1

Uvod

U današnjem svijetu brze urbanizacije i rastućih zahtjeva za efikasnim i održivim transportom, inteligentni transportni sistemi (ITS) igraju ključnu ulogu u transformaciji načina na koji se ljudi i roba kreću širom svijeta. Ovi napredni tehnološki sistemi integrišu informacione i komunikacione tehnologije sa infrastrukturom transporta radi poboljšanja bezbjednosti, efikasnosti i udobnosti transportnog procesa. ITS obuhvata senzorske mreže, bežičnu komunikaciju, tehnologije obrade podataka, vještačku inteligenciju i automatizaciju vozila. ITS omogućava prikupljanje, razmjenu i analizu podataka u realnom vremenu, što značajno doprinosi efikasnosti i bezbjednosti sistema.

ITS je kompleksan sistem i, donosi izazove u raznim djelovima njegove strukture. Ovaj rad će biti usmjeren na senzorsku mrežu ITS-a, konkretno na procjenu brzine vozila na osnovu video senzora.

U uvodnom dijelu rada, biće razmotren kontekst i motivacija ovog rada. Brzi urbani rast, sve gušći saobraćaj i sve veći ekološki izazovi postavljaju potrebu za novim pristupima transportu. ITS pružaju inovativna rešenja za optimizaciju saobraćajnih tokova, smanjenje zagušenja, minimizaciju saobraćajnih nesreća i unapređenje korisničkog iskustva. Poboljšanjem senzora i softvera za procjene podataka važnih u ITS-u može se poboljšati njegova efikasnost i jednostavnost primjene. Ovim se smanjuje vrijeme i cijena instalacije što ga čini lako dostupnim.

Sljedeći segment rada će istražiti prednosti koje ITS pružaju. Poboljšanje bezbjednosti saobraćaja, smanjenje emisije štetnih gasova, skraćenje vremena putovanja i bolje upravljanje infrastrukturom samo su neki od pozitivnih efekata koje ovi sistemi donose. Pored toga, biće predstavljena dosadašnja istraživanja procjene brzine vozila iz video snimka. Ovdje će biti pomenute popularnije metode i detaljno objašnjeni načini na koje rješavaju

glavne izazove kao što su kalibracija kamere, detekcija i praćenje vozila i način krajnje procjene.

Nakon toga, fokus će biti usmjeren na rekurentne neuralne mreže. Prvo će biti predstavljena jednostavna rekurentna neuralna mreža na kojoj će biti objašnjen princip rada i problemi koje ove mreže rješavaju. Nakon toga će biti detaljnije objašnjene naprednije rekurentne neuralne , kao što su GRU (eng. *Gated Recurent Unit*) i LSTM (eng. *Long Short-Term Memory*).

U narednom dijelu će biti posvećena pažnja detektorima objekata na video snimku i njihovom praćenju. Biće objašnjen način rada detektora zasnovanih na dubokom učenju. Ovdje će biti objašnjeni najznačajniji algoritmi, a to su YOLO (*You Only Look Once*) i R-CNN (*Region Convolutional Neural Network*).

Sljedeće poglavlje će predstaviti predloženu metodu procjene brzine vozila. Ovdje će biti izložen princip rada svakog dijela metode, i predstavljen način na koji se nadovezuju. Nakon toga će biti više riječi o rezultatima eksperimenata. Biće izvršena analiza uticaja više parametara na preciznost krajnje procjene i predstavljena konfiguracija sa optimalnim rezultatima.

U zaključku, ovaj rad će naglasiti značaj primjene dubokog učenja u inteligentnim transportnim sistemima i istaknuti potencijalne pravce za dalji razvoj. Kroz analizu izazova i mogućnosti, cilj je osvijetliti sve aspekte koje treba uzeti u obzir prilikom implementacije ove metode.

1.1 Inteligentni transportni sistemi

Pojam Inteligentni transportni sistem se pojavio posljednjih godina, označavajući modernizovani način upravljanja saobraćajem gdje se komunikacija između učesnika saobraćaja, kao i saobraćajne infrastrukture, obavlja na način koji se oslanja na razvoj informacionih i komunikacionih tehnologija [1]. ITS obuhvata različite komponente koje se međusobno integrišu kako bi omogućile efikasnije upravljanje saobraćajem, poboljšanje bezbjednosti i olakšavanje putovanja.

Senzori su osnovna komponenta ITS-a, koja omogućava prikupljanje stvarnih podataka o saobraćajnoj situaciji i uslovima na putevima. Ovi uređaji uključuju kamere, lidare, radare, magnetometre, ultrazvučne i termalne senzore.

Kamere su jedan od najčešće korišćenih senzora u ITS-u. Mogu biti postavljene duž puteva, raskrsnica, semafora i pešačkih prelaza. Kamere registruju vizuelne informacije o

vozilima i drugim objektima na putu. Ove informacije se koriste za analizu saobraćajnog toka, praćenje i identifikaciju vozila u saobraćaju. Trenutno se kamere najčešće nalaze u dopunskoj ulozi prilikom detekcije prekršaja i prikupljanja dokaznog materijala.

Lidari (eng. *Light Detection and Ranging*) su senzori koji koriste laserske zrake za mjerenje udaljenosti i stvaranje trodimenzionalnih slika okoline. Oni precizno identifikuju poziciju i brzinu vozila, kao i oblik i prisustvo prepreka na putu. Lidari se često koriste u tehnologijama autonomne vožnje kako bi vozila mogla precizno razumjeti okolinu.

Radari su senzori koji koriste elektromagnetne talase za otkrivanje objekata i mjerenje brzine. Oni su korisni za detekciju brzine vozila, praćenje gustine saobraćaja i upozoravanje vozača na potencijalne sudare ili prepreke. Radari su posebno važni za sisteme upozorenja na sudar (*Collision Warning Systems*) i regulaciju brzine.

Magnetometri detektuju prisustvo metala, poput vozila, na osnovu promjene magnetnog polja. Ovi senzori često se koriste za detekciju vozila na raskrsnicama, semaforima i naplatnim stanicama.

Termalni senzori registruju temperaturne razlike između vozila, asfalta i drugih objekata na putu. Ovi senzori su korisni za detekciju vozila u smanjenoj vidljivosti, npr. noću ili u magli.

Efikasna razmjena informacija između vozila, infrastrukture i centralnih sistema ključna je za uspješno funkcionisanje ITS-a. Bežične komunikacione mreže, poput V2X (vozilo-vozilo i vozilo-infrastruktura), omogućavaju vozilima da komuniciraju međusobno i sa okruženjem [2]. Ove mreže omogućavaju brzo dijeljenje informacija o brzini, pravcu kretanja, prisustvu prepreka i drugim relevantnim faktorima, čime se poboljšava sigurnost i efikasnost.

Prikupljeni podaci od senzora i komunikacionih mreža zahtijevaju obradu kako bi postali korisni informacioni sadržaj. Računarski sistemi za obradu podataka analiziraju i interpretiraju informacije o saobraćaju, prilagođavajući se dinamičkim uslovima na putevima. Ovi sistemi primjenjuju algoritme za predviđanje saobraćajnih zagušenja, identifikaciju rizičnih situacija i optimizaciju signalizacije na putu.

Vještačka inteligencija igra ključnu ulogu u obradi i interpretaciji ogromnih količina podataka koje generišu senzori i komunikacione mreže. AI algoritmi omogućavaju sistemu da "uči" iz podataka i donosi pametne odluke u realnom vremenu. Analitika podataka pomaže u identifikaciji uzoraka, predviđanju saobraćajnih uslova i pružanju informacija za bolje upravljanje saobraćajem.

Automatizacija vozila je sve važniji dio ITS-a. Ova komponenta uključuje tehnologi-

je kao što su adaptivni tempomat, automatsko kočenje, samostalno parkiranje i potpuno autonomna vozila. Ovi sistemi koriste kombinaciju senzora, komunikacionih mreža i AI tehnologije kako bi vozila mogla samostalno da upravljaju vožnjom, uzimajući u obzir okolinu i saobraćajne uslove.

Još jedna komponenta ITS-a su pametni semafori. Pametni semafori koriste informacije iz senzora i komunikacionih mreža kako bi se prilagodili stvarnim saobraćajnim uslovima. Na primjer, semafori mogu automatski produžiti zeleno svjetlo za vozila ako se detektuje zagušenje ili preusmjeriti saobraćaj u slučaju nesreće. Takođe, signalizacija na putu može pružiti informacije o vremenu putovanja, saobraćajnim preprekama i alternativnim rutama.

Implementacija ITS-a donosi i određene izazove. Ovi sistemi zahtijevaju visoku infrastrukturnu investiciju i koordinaciju između različitih sektora. Pitanja privatnosti i sigurnosti podataka takođe su važna, budući da se velika količina informacija razmjenjuje između vozila i infrastrukture.

1.2 Prethodna istraživanja

Među podacima koji senzori prikupljaju u ITS-u, brzina vozila je jedan od važnijih. Ovaj podatak igra ulogu u bezbjednosti saobraćaja, kao i predviđanju uslova u saobraćaju. Osim ovoga veoma je važna uloga u praćenju prekršaja učesnika u saobraćaju. Procjene brzine se mogu koristiti prilikom izdavanja kazni vozačima koji prekorače dozvoljenu brzinu, a one uključuju novčane i čak zatvorske kazne. Zbog ovoga je važna visoka preciznost senzora za procjenu brzine pa se često koriste skuplji uređaji kao što su radari.

Brzina vozila se može procijeniti i pomoću drugih senzora kao što su kamere i audio senzori. Ovo bi smanjilo cijenu i složenost implementacije sistema smanjujući broj potrebnih senzora. Razvoj metoda za procjenu brzine pomoću ovih senzora je umnogome omogućio napredak tehnika kompjuterske vizije.

Osnovni izazov procjene brzine pomoću kamera proizlazi iz inherentne diskretne prirode video senzora, koji transformišu trodimenzionalni svijet u dvodimenzionalnu ravan. Ovo stvara digitalnu reprezentaciju čija preciznost je obrnuto proporcionalna kvadratu udaljenosti od kamere do vozila [3].

Osim ovoga, izazovi koji se javljaju u vizuelnoj procjeni brzine su:

- detekcija vozila na snimku
- praćenje vozila
- kalibracija kamere
- način procjene brzine

Uprkos izazovima, u posljednje vrijeme je primjetan porast broja istraživačkih radova i predloženih metoda za vizuelnu procjenu brzine, što ukazuje na rastući interes i napredak u ovoj oblasti. U nastavku će biti sumirana istraživanja iz odabranih radova u ovoj oblasti.

Detekcija vozila

Detektovanje vozila se vrši na više načina. Obično je video na kojoj se detektuje brzina vozila snimljen pomoću kamera za nadzor saobraćaja. Ovo znači da je u svakom snimku pozadina ista, što se može iskoristiti kao pogodna karakteristika. Pošto su na snimcima promjenljiva jedino vozila, oduzimanjem statične pozadine dobija se slika na kojoj ostaje samo vozilo. Ovaj pristup je primijenjen u radovima [4, 5]. Ova metoda može biti veoma efikasna. Primjenom određenog praga ovako oduzeta slika se može pretvoriti u binarnu. Oblasti u binarnoj slici koje su preko postavljenog praga predstavljaju vozilo. Odavde se efikasno može izvući njegova pozicija i njenom promjenom izračunati brzina.

Metode dubokog mašinskog učenja kao što su YOLO, R-CNN, Fast RCNN, SSD su veoma efikasne kod problema detekcije objekta na video snimku. Kao takve vrlo često su zastupljene u dijelu za detekciju objekata na slici kod metoda procjene brzine. Ovaj pristup je primijenjen u radovima [6, 7, 8, 9]. U ovom pristupu se na slici ne detektuje regija na kojoj se nalazi vozilo nego granični okvir. Granični okvir je pravougaonik koji uokviruje detektovano vozilo na slici. Napredak algoritama, kao što je YOLO, omogućio je primjenu ovih detektora u detekcijama u realnom vremenu, što je veoma važno u primjeni za nadgledanje saobraćaja.

Jedan dio metoda prilazi detekciji vozila na drugi način. Za potrebe procjene brzine nije potrebna detekcija cijelog vozila. Dovoljna je detekcija nekog njegovog dijela. Praćenjem kretanja tog dijela prati se i kretanje samog vozila. Najčešće primjenjivan pristup je detekcija registarskih tablica. Razlog za njihovo praćenje je jasan kontrast na slici kao i standardizovan oblik unutar jedne zemlje. Osim ovoga, prednost ovog načina detekcije je moguća primjena alata za detekciju registarskih tablica, koji su veoma precizni,

brzi i pouzdani. Detekcija tablica se primjenjuje u metodama predloženim u radovima [10, 11, 12, 13]. Alati detekcije registarskih tablica su zasnovani na metodama kompjuterske vizije i detekcije teksta na slikama. Ovakav pristup direktno sakuplja podatke i o registarskim tablicama, što je korisno u nadzoru saobraćaja pri identifikaciji posmatranog vozila.

Praćenje vozila

U realnim situacijama nadzora saobraćaja, najčešće se javlja situacija da na svakom frejmu video snimka postoji veći broj vozila. U ovim slučajevima su neophodne metode praćenja vozila na snimku, da bi se svaka detekcija pripisala odgovarajućem vozilu. U radovima koji uzimaju u obzir ovu situaciju najčešće se koristi Kalmanovov filter. On je primijenjen u radovima [12, 8, 7, 6].

Kalmanov filter je matematički algoritam koji koristi niz podataka posmatranih tokom vremena, koji sadrže šum i druge nepreciznosti, kako bi sa većom tačnošću procenio nepoznate promjenljive [14]. Ima široke primjene u navigaciji, automatici i kompjuterskoj viziji.

Osnovne komponente Kalmanovog filtera u praćenju objekata uključuju: stanje sistema, dinamički model, mjerenja, matrice kovarijansi, predikcija i korekcija.

Stanje objekta koje želimo pratiti opisuje se vektorom stanja x , koji sadrži relevantne varijable i njihove brzine (npr. položaj, brzinu). Ovaj vektor se mijenja tokom vremena prema dinamičkom modelu objekta.

Kalmanov filter koristi model dinamike sistema koji opisuje kako se stanje objekta mijenja tokom vremena. Ovaj model obično koristi linearnu diferencijalnu jednačinu (*brzina = konstanta * vrijeme + inicijalna brzina*).

Senzori prikupljaju mjerenja objekta, ali ta mjerenja često sadrže šum. Mjerenja se opisuju matricom mjerenja H , koja povezuje vektor stanja i mjerenja.

Kalmanov filter koristi matrice kovarijansi kako bi modelovao nesigurnost u stanju sistema i u mjerenjima. Matrica kovarijansi stanja P predstavlja nesigurnost u procjeni stanja, dok matrica kovarijansi mjerenja R predstavlja nesigurnost u mjerenjima.

Kalmanov filter koristi dinamički model kako bi predvidio novo stanje sistema na temelju prethodne procjene stanja.

Nakon što se dobije nova procjena stanja, Kalmanov filter kombinuje ovu procjenu s

mjerenjima kako bi poboljšao procjenu stanja objekta. Ovaj korak koristi matricu korekcije K , koja se izračunava na temelju matrica kovarijansi.

Kalibracija

Kod većine metoda za detekciju brzine vozila veliki uticaj imaju parametri kamere. Parametri kamere moraju biti uzeti u obzir da bi procjena bila precizna. Ovaj proces unošenja podataka o kameri se naziva kalibracija. Kalibracija kamere podrazumijeva unos dvije vrste podataka, spoljašnjih i unutrašnjih. Spoljašni parametri predstavljaju informacije o poziciji kamere, njenoj visini i udaljenosti u odnosu na put kao i uglu posmatranja. Kod stereo kamera takođe su potrebni podaci o međusobnoj udaljenosti kamera. Unutrašnji parametri predstavljaju informacije o broju frejmova na slici i rezoluciji slike, odnosno broju piksela na slici.

Kalibracija je potrebna zbog uspostavljanja znanja o okolini. Znanjem nekih rastojanja u okolini, mogu se mapirati rastojanja na piksele na frejmu snimka. Odavde se može direktno izračunati pređena distanca vozila u stvarnosti pomoću snimka tog vozila. Poznavanjem unutrašnjih parametara, kao što je broj frejmova može se dobiti vrijeme za koje je vozilo prešlo određenu distancu. Ove dvije informacije su dovoljne za procjenu brzine vozila. Kalibracija je obično ručna. Ovo uvodi potrebu prilagođavanja procjene posebno za svaku kameru na kojoj se procjena koristi. U radovima [9, 8, 7, 5, 10] se koristi ovakva kalibracija.

Potreba same kalibracije predstavlja problem u detekciji brzine vozila. Potrebotom unosa okolnih informacija se otežava instalacija detektora, kao i ograničava njihova prilagodljivost na bilo kakve promjene. Neka promjena u postavci zahtijeva novu kalibraciju. Primjena metode je takođe ograničena poznavanjem svih parametara, jer nekada oni nisu poznati.

Problem potrebnih spoljašnjih parametara se djelimično prevazilazi upotrebom stereo kamera. Stereo kamere posjeduju snimke iz više uglova. Poređenjem ovih snimaka i razlike uglova između objekata na različitim snimcima omogućava se izračunavanje stvarnih distanci. Metode koje koriste stereo kamere zahtijevaju samo unutrašnje parametre kao i dodatni parametar o međusobnom položaju kamera. Ovaj pristup je primijenjen u [11]. Ovdje dolazi do novih izazova u cijeni i složenosti postavljanja i rada sa stereo kamerama. Takođe, javljaju se evidentni nedostaci u infrastrukturi gdje su mnogo češće zastupljene monokularne kamere.

Automatska kalibracija

Kalibracija predstavlja veliki problem kod metoda za detekciju brzine vozila, posebno potreba ručne kalibracije. Uzimajući ovo u obzir, razvijale su se automatske metode kalibracije. Kod ovih metoda određuju se takozvane tačke nestajanja (eng. *vanishing points*). Ove tačke određuju pravac na videu duž kojeg prolazi put i duž kojeg se kreću vozila. Određivanje ovog pravca se može odraditi na više načina.

Određivanje pravca pomoću tačaka nestajanja se najjednostavnije može uraditi posmatranjem puta, i to bijelih linija na putu. Pod pretpostavkom da je put prav, detekcijom ovih linija jednostavno je detektovati pravac kretanja. Problem kod ovog pristupa je što nije uvijek primjenljiv, kao što je slučaj tamo gdje bijele linije nisu vidljive. Ovakav pristup se koristi u [6]. Kod ovakvih pristupa kao osnova za određivanje odnosa stvarnih dimenzija i dimenzija na videu se koristi neka standardizovana veličina. Najčešće je to širina puta i traka na putu, kao u što je u [15]. Ovi pristupi imaju problem sa nepreciznostima u praćenju standarda u stvarnosti, kao i različitim standardima na različitim mjestima.

Potpuno automatska kalibracija je predstavljena istraživanjem [16]. Kalibracija predstavljena u ovom radu je upotrijebljena kao osnov različitih metoda detekcije brzine vozila. Ova kalibracija upotrebljava dvije tačke nestajanja za dva pravca i podatke o veličini velikog broja vozila.

Prva tačka nestajanja je slična kao u prethodnom slučaju. Pravac koji ona predstavlja je paralelan pravcu kretanja posmatranih vozila i pravcu prostiranja puta. Druga tačka nestajanja određuje pravac paralelan putu, odnosno ravni na kojoj se put nalazi, a normalan na pravac kretanja vozila. Ova dva pravca služe za orijentaciju puta i određivanja ugla puta u odnosu na kameru. Takođe imaju upotrebu prilikom detekcije vozila, gdje se većina ivica vozila poklapa sa ova dva pravca.

Pomoću ovih pravaca se konstruiše 3d oblik koji određuje vozilo na slici. Iz njega se mogu dobiti dimenzije posmatranog vozila na slici. Ovaj proces određivanja dimenzija vozila se radi za veći broj vozila, zbog čega proces kalibracije traje nekoliko minuta. Raspodjela dimenzija posmatranih vozila se upoređuje sa ranije prikupljenom raspodjelom dimenzija velikog seta vozila. Poklapajući ove raspodjele može se dobiti procjena odnosa stvarnih dimenzija i dimenzija na videu.

Proces automatske kalibracije podrazumijeva postavljanje kamere na neku lokaciju za nadzor saobraćaja. Nakon toga se kamera ostavi da posmatra saobraćaj neko vrijeme, obično nekoliko minuta. Prednost ovoga je da se kalibracija ne vrši ručno, kao i mogućnost brze kalibracije koja se može ponoviti ako se neki uslovi promijene.

Izračunavanje brzine

Dosadašnje metode konačnu procjenu brzine računaju kao vrijeme za koje je vozilo prešlo neku distancu na putu. Distance na putu se određuju pomoću različitih oblika kalibracije. One se predefinišu na putu i ponavljaju se periodično sa nekim određenim razmakom. Kada vozilo pređe preko jedne distance direktno se izračunava brzina jer je poznata i razdaljina i vrijeme za koje je vozilo tu razdaljinu prešlo.

Ovakav pristup je u mnogome zavistan od kvaliteta kalibracije gdje se dobijaju vrijednosti distanci i vremena prolaska vozila.

Glava 2

Rekurentne neuralne mreže

Rekurentne neuralne mreže su se prvi put pojavile 1980-ih godina u jednostavnom obliku. Do veće popularizacije i značaja dobile su 1990-ih godina sa sve većom dostupnošću računarske snage i bogatijim skupovima podataka [17]. U ovom periodu su našle primjenu u rješavanju ozbiljnijih sekvencionalnih problema kao što su finansijsko predviđanje i predviđanje pozicije objekata. Dalja unapređenja su nastala pojavom GRU (eng. *Gated recurrent unit*) i LSTM (eng. *Long short-term memory*) arhitektura koje prevazilaze probleme nestajućih i eksplodirajućih gradijenata. Ove arhitekture su omogućile primjenu rekurentnih mreža na probleme obrade prirodnog jezika, generisanja i prevođenja teksta.

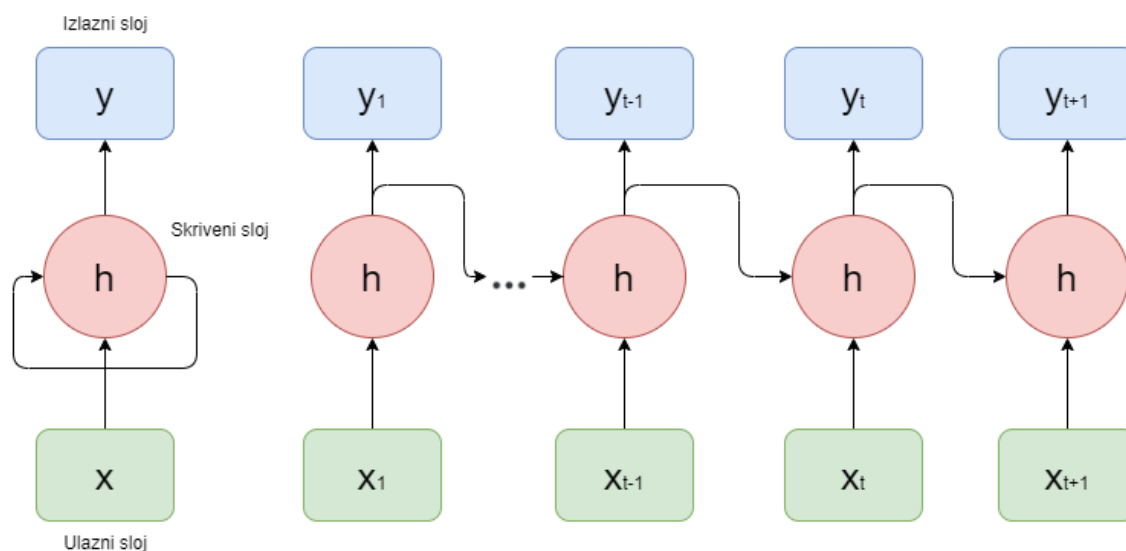
Primarna upotreba rekurentnih mreža je u modelovanju sekvencijalnih podataka. Rekurentna mreža prima i obrađuje podatke iz sekvence jedan po jedan, čuvajući u memoriji podatke o prethodnim elementima sekvence. Ovo u osnovi omogućava obradu sekvenci različitih dužina (pošto se na ulaz mreže u jednom koraku dovodi jedan element sekvence) kao i pamćenje konteksta u kojem se element sekvence nalazi.

Izazovi sa kojim se sve rekurentne neuralne mreže susreću su problemi čuvanja dugačkih zavisnosti koji se pojavljuju u obliku nestajućih i eksplodirajućih gradijenata (eng. *vanishing and exploding gradient*). Prilikom čuvanja relevantnih podataka u setu fiksne veličine radi predviđanja budućeg stanja, potrebno je izvršavati eksponencijalno zaboravljanje podataka daljih u vremenu zbog stabilnosti i robusnosti sistema na šum [18]. Ovo zaboravljanje dovodi do veoma malog uticaja daljih događaja u odnosu na vremenski bliže događaje otežavajući učenje dugačkih zavisnosti.

2.1 Osnovni principi rada rekurentnih neuralnih mreža

Rekurentne neurane mreže pripadaju nadgledanim modelima mašinskog učenja. Ove mreže se razlikuju u odnosu na standardne neuralne mreže po tome što dozvoljavaju veze među neuronima u unutrašnjem, skrivenom sloju sa vremenskim kašnjenjem. Ovo čini svojevrsnu povratnu spregu koja predstavlja osnovni mehanizam pamćenja podataka koji su prethodili nekom koraku, kao i uticaj ove memorije na buduće stanje. Ovo pamćenje se ostvaruje pomoću čuvanja unutrašnjeg stanja. Stanje se izračunava u svakom koraku ulazne sekvence a zbog dozvoljenih veza sa vremenskim kašnjenjem, ovo stanje utiče na sljedeći korak sekvence i sljedeći izlaz.

Ćelija rekurentne neuralne mreže je prikazana na slici 2.1 na lijevoj strani. Na desnoj strani slike je prikazana mreža razvijena na više vremenskih koraka. Prikazano je kako prethodni korak utiče na sljedeće korake.



Slika 2.1: Jednostavna rekurentna neuralna mreža i njena razvijena struktura

Ćeliju rekurentne neuralne mreže sačinjavaju tri sloja, a to su ulazni sloj, skriveni (rekurentni) sloj i izlazni sloj. Na ulazni sloj se dovodi niz vektora koji predstavljaju ulaz u svakom od posmatranih vremenskih koraka. Ulaz će se označavati kao vektor x koji je $x = \{\dots x_{t-1}, x_t, x_{t+1} \dots\}$, gdje je t vremenski trenutak. x_t je vektor koji se dovodi na ulazni sloj u jednom vremenskom trenutku tj. $x_t = \{x_{t1}, x_{t2}, \dots x_{tN}\}$. Ovdje N predstavlja dužinu ulaznog niza i broj neurona ulaznog sloja.

Skriveni sloj će biti označen kao h , gdje je h_t skriveni sloj u vremenskom trenutku t . Skriveni sloj se sastoji od M neurona $h_t = \{h_{t1}, h_{t2}, \dots h_{tM}\}$. Ovaj sloj predstavlja unutrašnje

stanje ili memoriju rekurentne neuralne mreže. Kod potpuno povezane rekurentne mreže svaki od ulaza je povezan sa svakim neuronom u skrivenom sloju. Unutrašnje stanje u nekom vremenskom trenutku h_t je funkcija ulaza u tom trenutku x_t i unutrašnjeg stanja u predhodnom trenutku h_{t-1} :

$$h_t = F(h_{t-1}, x_t)$$

odnosno u specifičnoj parametrizaciji

$$o_t = W_{rec}\sigma(h_{t-1}) + W_{in}x_t + b_H$$

$$h_t = f_H(o_t).$$

Ovdje W_{in} i W_{rec} predstavljaju matrice težine u vezama, između ulaznog i skrivenog sloja W_{in} i u rekurentnoj vezi između neurona skrivenog sloja W_{rec} . σ predstavlja aktivacionu funkciju, koja je u ovom slučaju obično sigmoid ¹, dok f_H predstavlja aktivacionu funkciju skrivenog sloja koja je obično hiperbolični tangens \tanh ². b_H je bias skrivenog sloja.

Izlazni sloj rekurentne neuralne mreže će biti označen kao y , gdje je y_t izlaz u određenom vremenskom trenutku t . Izlazni sloj se sastoji od P neurona $y_t = \{y_1, y_2, \dots, y_P\}$. Vrijednost izlaznog sloja je funkcija skrivenog stanja u istom vremenskom trenutku h_t , odnosno

$$y_t = f_o(W_{out}h_t + b_o).$$

Matrica težine koja predstavlja vezu izlaznog i skrivenog sloja je W_{out} . b_o je bias izlaznog sloja, dok je f_o aktivaciona funkcija izlaznog sloja. Vrijednost predikcije izlaznog sloja se izračunava u svakom vremenskom koraku.

¹Aktivaciona funkcija u neuralnim mrežama je matematička funkcija koja se primenjuje na težinske sume ulaza neuronskog čvora kako bi se odredio izlaz tog čvora. Koristi se da bi se ograničili vrijednosti izlaza iz čvora kao i da bi se uvele nelinearnosti. Uvođenje nelinearnosti omogućava učenje kompleksnijih karakteristika. Sigmoid je jedna od najčešće korištenih aktivacionih funkcija. Sigmoid funkcija je predstavljena jednačinom:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

²Hiperbolični tangens je još jedna važna aktivaciona funkcija koja se često primjenjuje u mašinskom učenju. Ona je predstavljena jednačinom:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

Inicijalizacija težinskih matrica je takođe važna kod rekurentnih neuralnih mreža. Obično se težinske matrice inicijalizuju na slučajne male vrijednosti, Gausovom raspodjelom, standardne devijacije između 0.001 i 0.01. Bias je obično postavljen na nulu, međutim u nekim slučajevima može biti postavljen na neku malu vrijednost, obično kod izlaznog biasa (b_o) [19].

2.2 Treniranje rekurentnih neuralnih mreža

Performanse svake neuralne mreže, pa tako i rekurentne neuralne se određuju pomoću funkcije cijene (eng. *Loss function*). Ova funkcija predstavlja razliku između izlaza iz mreže (y_t) i označenih vrijednosti (z_t)

$$J(y, z) = \sum_{t=1}^T J(y_t, z_t).$$

Kao razlika između dobijenog rezultata i stvarne vrijednosti obično se uzima Euklidovo ili Hammingovo rastojanje kada su u pitanju regresioni problemi sa realnim vrijednostima. Kada su u pitanju problemi klasifikacije koristi se kros-entropija vjerovatnoća izlaza [20].

Prilikom treniranja neuralne mreže osnovni cilj je minimizacija funkcije cijene. Ovo predstavlja značajan problem kod rekurentnih neuralnih mreža. Veze u skrivenom sloju sa vremenskim kašnjenjem predstavljaju dodatnu dimenziju koja utiče na vrijednost funkcije cijene.

2.3 Optimizacija pomoću gradijentnog spusta

Gradijentni spust je najčešći optimizacioni algoritam. Zasniva se na prilagođavanju težina u težinskoj matrici pomoću parcijalnih izvoda funkcije cijene po svakoj od težina u matrici. Da bi izračunavanje parcijalnih izvoda bilo moguće potrebno je da nelinearne aktivacione funkcije budu diferencijabilne

$$W_{t+1} = W_t - \frac{\lambda}{U} \sum_{k=1}^U \frac{\partial J_k}{\partial W}.$$

Ovdje λ predstavlja korak učenja, a U predstavlja veličinu ukupnog seta za treniranje. Gradijentni spust se u osnovnom obliku odnosi na *skupni gradijentni spust* (eng. *Batch gradient decent*). U ovom obliku u svakoj iteraciji optimizacije težinskih matrica računa se gradijent nad čitavim skupom podataka. Zbog ovoga može biti spor prilikom treniranja i ne može se koristiti za setove podataka koji se ne mogu smjestiti u radnoj memoriji [21]. Takođe ovo znači da je za treniranje potrebno posjedovanje čitavog seta podataka, pa se ne može trenirati u hodu tj. kako podaci pristižu. Kod ovog algoritma konvergencija ka globalnom minimumu je zagantovana za konveksne funkcije cijene i ka lokalnom minimumu za funkcije cijene koje nisu konveksne.

Sohastički gradijentni spust (eng. *Stochastic gradient decent SGD*) je drugi oblik gradijentnog spusta koji, suprotno skupnom gradijentnom spustu, izračunavanje gradijenata i ažuriranje matrica težine radi za svaki podatak iz skupa za treniranje. Često ažuriranje matrica težine dovodi do varijacija u funkciji cijene. Ovo omogućava SGD algoritmu da šire istražuje površinu funkcije cijene, što može rezultovati u pronalaženju boljeg lokalnog minimuma kod funkcija koje nisu konveksne. Ova osobina takođe sprečava SGD algoritam da konvergira tačno ka minimumu, navodeći ga da preskoči tačan minimum. Ovaj problem se otklanja uvođenjem adaptivnog koraka učenja koji se smanjuje prilikom konvergencije ka minimumu ili uvođenjem momenta učenja o kojem će biti više riječi.

SGD je brži algoritam od skupnog gradijentnog spusta. Iako vrši izračunavanje gradijenata na svakom podatku iz skupa za treniranje, ovo izračunavanje je brže jer je u pitanju samo jedan ulazni vektor. SGD takođe brže konvergira ka minimumu, i može se koristiti za učenje u hodu.

Kombinacijom ova dva pristupa dobija se novi metod *Mini-skupni gradijentni spust* (eng. *Mini-batch gradient decent*). On dijeli skup podataka za treniranje na male podskupove i izračunava gradijent na njima. Ovo dovodi do stabilnijeg spusta jer su rjeđa ažuriranja težinskih matrica. U drugu ruku, omogućava efikasnije treniranje i brže izračunavanje potrebnih gradijenata. Ova metoda se najčešće primjenjuje kod treniranja neuralnih mreža. Najčešće se koriste setovi veličine između 50 i 256, međutim ovo može da varira u zavisnosti od primjene [21].

Prilikom treniranja neuralnih mreža pomoću gradijentnog spusta, potrebno je izračunavanje gradijenata težinskih matrica svakog sloja. U ovu svrhu se koristi algoritam *Propagacije unazad* (eng. *Backpropagation*). On omogućava propagaciju greške od posljednjeg, izlaznog, sloja do ulaznog sloja. Ovo omogućuje određivanje uticaja svakog sloja u konačnoj grešci na izlaznom sloju i pomoću toga precizno ažuriranje težinskih

matrica u svakom sloju.

Kod rekurentnih neuralnih mreža se javlja uticaj prethodnih vremenskih koraka, pa konačna greška osim kroz nadovezane slojeve, mora da propagira i kroz vrijeme do prvog vremenskog koraka. Ovakva propagacija unazad se naziva propagacija unazad kroz vrijeme (eng. *Backpropagation through time* BPTT). BPTT se ostvaruje razvijanjem vremenskih koraka u prostornu dimenziju. Mreža se kopira za svaki vremenski korak i nadovezuje kao višeslojna neuralna mreža. Ovako razvijena struktura je prikazana na slici 2.1 lijevo. Nakon ovog razvijanja se primjenjuje algoritam propagacije unazad kao za višeslojnu neuralnu mrežu. Izračunati gradijenti za svaku kopiju se na kraju sumira. Rezultujući gradijenti se koristi za ažuriranje skrivenog sloja. BPTT algoritam ima efikasnu računsku složenost koja se dobija na uštrb velike memorijske složenosti. Memorijska složenost proizilazi iz potrebe čuvanja unutrašnjeg stanja za svaki vremenski korak [22].

Uzimajući u obzir parametre mreže $W = \{W_{rec}, W_{in}, W_{out}\}$ i predstavljajući skriveni sloj mreže u vremenskom trenutku sa h_t gradijenti se mogu predstaviti kao:

$$\frac{\partial J}{\partial W} = \sum_{t=1}^T \frac{\partial J_t}{\partial W},$$

$$\frac{\partial J_t}{\partial W} = \sum_{k=1}^t \left(\frac{\partial J_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k^+}{\partial W} \right).$$

Ovdje izraz $\frac{\partial h_k^+}{\partial W}$ predstavlja uticaj prethodnih slojeva na grešku u trenutnom vremenskom koraku [23].

Drugi način za izračunavanje gradijenata je *Rekurentno učenje u realnom vremenu* (eng. *Real time recurrent learning* RTRL). Algoritam RTRL se razlikuje po tome što odmah računa gradijent skrivenog sloja u datom vremenskom koraku, čim se u propagaciji unaprijed izračuna skriveno stanje h_t . Ovaj pristup, poznat kao diferencijacija u naprednoj propagaciji, eliminiše potrebu za čuvanjem svih prethodnih ulaza i skrivenih stanja. Kao rezultat toga, RTRL ima jedinstvenu prednost a to je da zahtjevi za skladištenje ostaju konstantni, nezavisno od dužine sekvence. Ova izuzetna osobina omogućava algoritmu RTRL da efikasno obrađuje značajno duže sekvence u poređenju sa BPTT, omogućavajući modelu da uhvati i uzme u obzir duže zavisnosti u podacima. Međutim, važno je napomenuti da ova efikasnost u upotrebi memorije dolazi uz povećanu računarsku složenost kao trošak.

2.4 Gradijentni spust sa inercijom, Adam optimizacija

SGD algoritam, iako brz i efikasan, ima problem prilikom spusta zbog varijacija u izračunatom gradijentu. Ove varijacije dovode do kretanja koje većim dijelom nije usmjereno ka stvarnom minimumu funkcije cijene. Problem neusmjerenog spusta se može umanjiti primjenom metoda inercije. Metod inercije pomaže usmjeravanju spusta SGD algoritma kao i umanjivanju oscilacija. On funkcionira tako što uključuje uticaj gradijenta izračunatog u prethodnom koraku. Primjenjujući ovu metodu ažuriranje težinske matrice se vrši na sljedeći način:

$$u_t = \gamma u_{t-1} + \lambda \frac{\partial J_t(W_t)}{\partial W_t},$$

$$W_{t+1} = W_t - u_t.$$

Ovdje γ predstavlja parametar koji određuje intezitet uticaja prethodnog gradijenta. Ovaj parametar se obično postavlja na vrijednost 0.9. Ideja ovoga pristupa je pojačavanje uticaja pravca koji se ponavlja u gradijentima i smanjivanje uticaja pravca koji variraju, što čini algoritam stabilnijim. Na ovaj način se spust ubrzava u stvarnom pravcu sve dok ne dođe do minimuma.

SGD algoritam sa inercijom se može dalje ubrzati, dodavanjem algoritmu procjenu budućeg stanja težinske matrice. Ovo ubrzanje donosi *Nesterov ubrzani gradijent* (eng. *Nesterov accelerated gradient* NAG). Na sljedeće stanje težinske matrice će većim dijelom uticati inercija prethodnog gradijenta, pa se izračunavanjem

$$v_t = W_t - \gamma u_t$$

dobija dobra aproksimacija sledećeg stanja. Izračunavanjem gradijenta ove procjene i koristeći ga za sledeće ažuriranje, dalje se ubrzava spust sa inercijom

$$u_t = \gamma u_{t-1} + \lambda \frac{\partial J_t(v_t)}{\partial v_t}.$$

Do sada predstavljeni algoritmi imaju konstantan korak učenja, jednak za sve parametre u težinskim matricama. Uvrštavajući adaptivni korak učenja koji se prilagođava prilikom gradijentnog spusta eliminiše se potreba za ručnim prilagođavanjem koraka učenja. Takođe sa mogućnošću posebnog koraka učenja za svaki parametar dozvoljava se upotreba manjih koraka kod češćih parametara, a istovremeno i većih koraka kod parametara koji su rjeđi [21].

Ažuriranje kod SGD algoritama sa adaptivnim korakom učenja ažuriranje težinske matrice se vrši na sljedeći način:

$$g_t = \frac{\partial J_t(W_t)}{\partial W_t},$$

$$W_t = W_{t-1} - \frac{\mu}{\sqrt{G_t + \varepsilon}} * g_t.$$

Ovdje G_t predstavlja dijagonalnu matricu u kojoj se nalazi suma kvadrata dosadašnjih gradijenata. Element ε služi za izbjegavanje situacija u kojim bi moglo doći do dijeljenja sa nulom. On se postavlja na malu vrijednost, obično 10^{-8} . Operacija između trenutnog gradijenta i adaptivnog koraka učenja je množenje matrice po elementima. Ovaj pristup ima problem rasta matrice G_t . Pošto se tokom treniranja sumiraju kvadrati gradijenat, ova matrica postaje sve veća. Njen rast uzrokuje smanjivanje koraka učenja dok ne postane toliko mali da dalje učenje nije moguće.

Problem sa matricom kvadrata gradijenata se prevazilazi pomoću uvođenja opadajuće srednje vrijednost kvadrata gradijenata ($E[g^2]$). Ova vrijednost se ažurira na sledeći način:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1 - \gamma)g_t^2.$$

Opadajuća srednja vrijednost se kontroliše konstantom $\gamma < 1$, pa se ne može desiti da neograničeno raste. Vrijednost ove konstante se obično postavlja na 0.9.

Algoritam *Adaptivne estimacije inercije* (eng. *Adaptive moment estimation Adam*) kombinuje algoritme inercije i adaptivni korak učenja. Adam algoritam čuva srednju vrijednost prethodnih kvadrata gradijenata za izračunavanje adaptivnog koraka učenja. Osim ovoga Adam čuva opadajuću srednju vrijednost prethodnih gradijenata, koja predstavlja gradijentni spust sa inercijom. Ova dva parametra, opadajuća srednja vrijednost prethodnih gradijenata m_t i opadajuća srednja vrijednost prethodnih kvadrata gradijenata v_t , se ažuriraju na sledeći način:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t,$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2.$$

Parametri v_t i m_t se inicijalizuju na vektor nula. Ova inicijalizacija dovodi do sklonosti algoritma da teži nuli, pogotovu u početnim koracima. Zbog ovoga je potrebno izvršiti korekciju parametara na sljedeći način:

$$\tilde{m}_t = \frac{m_t}{1 - \beta_1},$$

$$\tilde{v}_t = \frac{v_t}{1 - \beta_2}.$$

Sa korigovanim parametrima se nastavlja konačno ažuriranje težinske matrice:

$$W_t = W_{t-1} - \frac{\lambda}{\sqrt{v_t} + \epsilon} * m_t.$$

Vrijednosti konstanti se rijetko ručno prilagođavaju. One su obično postavljene na vrijednosti, $\beta_1 = 0.9$, $\beta_2 = 0.999$ i $\epsilon = 10^{-8}$.

2.5 Regularizacija

Kod neuralnih mreža, prilikom treniranja moguća je pojava pretreniranja (eng. *Overfitting*). U ovoj situaciji model je previše prilagođen podacima u setu za treniranje, pa vrši slabiju generalizaciju i daje slabije rezultate na novim podacima. U ovom slučaju se dodaje regularizacija u modelu, da bi se učenje bolje generalizovalo. Postoji više metoda regularizacije a najčešće korištene su L_1 , L_2 i *Dropout* regularizacija.

Kod L_1 i L_2 regularizacije se mijenja funkcija cijene. U funkciji se dodaje element koji predstavlja L_1 ili L_2 normu težina težinske matrice.

$$J(W) = J(W) + \gamma \|W\|_p^p$$

$$\|W\|_p = \left(\sum_{j=0}^{|W|} |W|^p \right)^{\frac{1}{p}}$$

gdje je $p \in \{1, 2\}$ u zavisnosti od tipa regularizacije, a γ predstavlja parametar koji određuje jačinu regularizacije. Dodavanjem ovog člana funkcija cijene prilikom minimizacije teži i smanjivanju vrijednosti parametara u težinskoj matrici, favorizujući one koje koji imaju vrijednost blisku nuli. Ovim se vrši selekcija parametara, čime se poništavaju oni koji dovode do pretreniranja.

Dropout regularizacija je oblik regularizacije koja prilikom treninga prekine slučajno odabrane konekcije između slojeva. Ona onemogućava modelu da se osloni previše na mali broj težina u težinskoj matrici, time omogućavajući generalnije učenje. Primjena ovakvog dropout-a na rekurentnim neuralnim mrežama nije pogodna jer bi dolazilo do gubitaka memorije. Zbog ovoga je predložena metoda *RNNDrop*. Ovdje se na početku svake sekvence nasumično odredi jedna maska koja će se koristiti za dropout. Ova maska se primjenjuje tokom čitavog trajanja na jednoj sekvenci [24].

2.6 Duboke rekurentne neuralne mreže

Duboko učenje predstavlja vrstu mašinskog učenja koja koristi višeslojne, hijerarhijske modele zbog njihovih mogućnosti efikasnijeg i preciznijeg predstavljanja složenih funkcija. Upotreba višeslojne neuralne mreže povećava reprezentativnu moć modela, pa tako smanjuje ukupan broj čvorova potrebnih za reprezentaciju funkcije čineći ga efikasnim.

Rekurentne neuralne mreže su po svom obliku već višeslojne. Ova višeslojna struktura se može vidjeti na slici 2.1 gdje je prikazana razvijena rekurentna mreža. Na slici se vidi raniji ulaz prolazi kroz više skrivenih slojeva dok ne dođe do nekog kasnijeg izlaza. Iako je rekurentna mreža duboka u vremenu, nije duboka u prostoru. Posmatrajući individualno samo jedan vremenski korak, sve konekcije u tom koraku su plitke tj. imaju samo jedan sloj neurona. Pošto rekurentna neuralna mreža ima tri vrste konekcija u jednom vremenskom trenutku, dubina se može dodati na više načina.

Duboka ulazni sloj - skriveni sloj konekcija predstavlja produbljivanje rekurentne neuralne mreže postavljajući više slojeva između ulaznog i skrivenog sloja. Ovo omogućava učenje bolje reprezentacije ulaza u jednom vremenskom koraku. Prosleđivanje ove apstraktne reprezentacije ulaznog koraka skrivenim slojevima neuralne mreže omogućava lakše učenje vremenskih karakteristika.

$$h_t = f_H(W_{rec}\sigma(h_{t-1}) + W_{in}f_L(W_L(\dots f_1(W_1x_t))))$$

Duboka skriveni sloj - izlazni sloj konekcija se ostvaruje postavljanjem dodatnih slojeva neurona između skrivenog i izlaznog sloja rekurentne neuralne mreže. Ova izmjena omogućava bolju reprezentaciju skrivenog stanja prilikom izračunavanja izlaza. Bolja reprezentacija toleriše veće varijacije u skrivenom sloju, a takođe mu daje mogućnost efikasnijeg čuvanja memorije što može rezultovati kompaktnijim modelom.

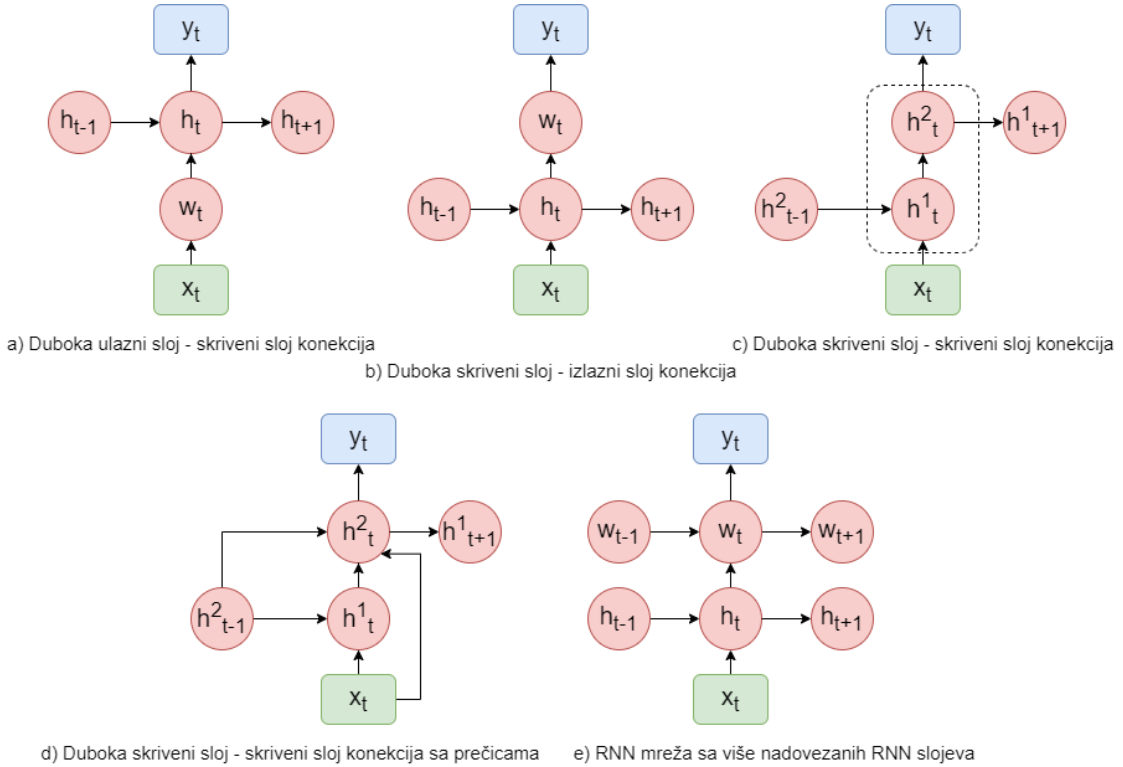
$$y_t = f_o(W_Lf_L(W_{L-1}f_{L-1}(\dots W_1h_t)))$$

Duboka skriveni sloj - skriveni sloj konekcija predstavlja mrežu dobijenu proširivanjem skrivenog sloja rekurentne neuralne mreže na više slojeva. Ovim povećavamo reprezentativnu moć unutrašnjeg skrivenog sloja čime se omogućava bolje učenje i bolja memorija. Dodavanjem novih slojeva unutar skrivenog sloja povećava se dužina puta ulaza u prethodnim vremenskim trenucima. Pošto se povećanjem puta smanjuje broj vremenskih koraka koji mogu da utiču na trenutni izlaz, ovo produbljivanje smanjuje dužinu

dugoročnog pamćenja.

$$h_t = f_h(W_L f_L(W_{L-1} f_{L-1}(\dots W_1 f_1(W_{rec} h_t + W_{in} x_t + b_H)))$$

Ovaj problem se može umanjiti uvođenjem konekcija koje zaobilaze dodate unutrašnje slojeve. Ove konekcije obezbjeđuju kraći put duž kojeg propagacija unazad može proći sa manjim umanjjenjem gradijenata.



Slika 2.2: Oblici duboke rekurentne neuralne mreže

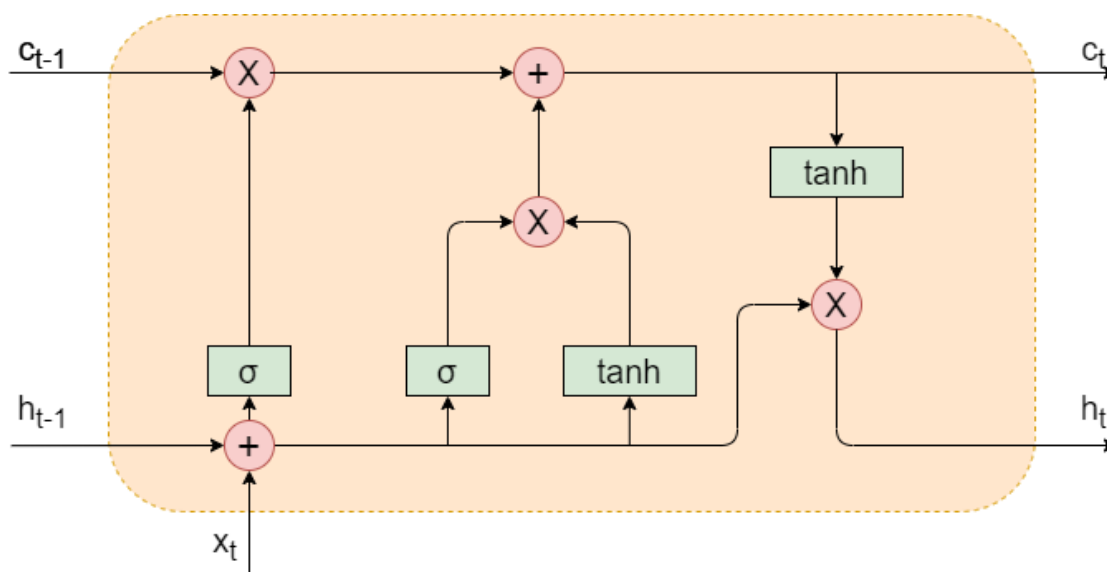
Produblјivanje Rekurentne neuralne mreže se takođe može ostvariti naslaganjem više rekurentnih mreža zaredom. Ovo proširivanje se obavlja sa ciljem ohrabrivanja različitih nivoa rekurentne mreže da rade u različitim vremenskim okvirima [25]. Svaki sloj ove mreže je i dalje jednoslojan, tako da ovakav način produblјivanja ne povećava mogućnost predstavljanja kompleksnih relacija. Kombinacijom ovog i nekog od ranije pomenutih pristupa možemo i povećati reprezentativnu moć i omogućiti rad različitih nivoa neuralne mreže u različitim vremenskim okvirima. Kombinovanje se ostvaruje naslaganjem neuralnih mreža čije su jedinice produblјene sa više slojeva.

$$h_t = f_h(W_{rec}^L h_{t-1}^L + W_{in}^L h_t^{L-1} + b_H^L)$$

$$h_t^0 = x_t$$

2.7 Long short-term memory (LSTM)

Problem nestajućeg i eksplodirajućeg gradijenta kod rekurentnih neuralnih mreža je veoma značajan. On ograničava sposobnost mreže da uči na dužim sekvencama i da pamti dugoročne karakteristike. Da bi se ovo prevazišlo, razvijene su različite arhitekture rekurentne neuralne mreže koje izbjegavaju ovaj problem. Najznačajnije su LSTM (*Long short-term memory*), koji je tema ove sekcije, i GRU (*Gated recurrent unit*), koji će biti predstavljen u narednoj sekciji.



Slika 2.3: Arhitektura ćelije LSTM mreže

LSTM je moćna rekurentna neuralna mreža koja se pojavljuje u više oblika. Svaki oblik ima prednosti u nekoj vrsti problema za koju je predviđen. Ovdje će biti prikazana osnovna verzija LSTM-a.

Centralni dio LSTM-a je *constant error carousel* - CEC. CEC predstavlja način održavanja konstantnog protoka signala greške kroz svaku ćeliju mreže. Kod obične rekurentne neuralne mreže na izlaz, koji se prosljeđuje daljnim koracima, u jednom vremenskom koraku se primjenjuju različite aktivacione funkcije. Prilikom propagacije unazad i izračunavanja gradijenata, aktivaciona funkcija utiče na smanjivanje intenziteta gradijenta. CEC predstavlja postojanje izlaza koji se prosljeđuje kroz svaku ćeliju mreže a na koji ne utiču aktivacione funkcije koje bi smanjile gradijent. CEC je u stvari i sam rekurentna neuralna mreža. Unutar LSTM-a, CEC je proširen dodatnim karakteristikama koje se nazivaju kapije (eng. *gates*). Ovako proširen CEC se naziva *memorijska ćelija* (eng. *memory cell*). Kapije omogućavaju da se memorijska ćelija prilagođava na osnovu ulaznih

podataka a da se na nju ne dodaju direktno aktivacione funkcije.

LSTM ćelija, za razliku od ćelije obične rekurentne mreže, prosljeđuje ćeliji u sljedećem koraku dva parametra: vrijednost memorijske ćelije c_t , i vrijednost izlaza h_t .

Uvođenje kapija u LSTM arhitekturu takođe služi zaobilazanju problema konflikata u ažuriranju težinskih matrica. Ovaj problem se pojavljuje kod rekurentnih neuralnih mreža zbog upotrebe jedne težinske matrice za više zadataka. Ovo se može ilustrovati na sledećem primjeru. Posmatrajući jednu vremensku seriju u kojoj postoji jedan duži period gdje je jedna ulazna karakteristika ključna za smanjenje ukupne greške. U ovim trenucima, izračunati gradijent će se primjenjivati na težinsku matricu tako da se pojača uticaj ovog ulaza. Recimo da se nakon ovog perioda, pojavi period gdje je uticaj ovog ulaza potrebno smanjiti da bi se smanjila ukupna greška. Tada će izračunati gradijenti težiti da smanje isti parametar u matrici težine. Ovo bi predstavljalo konflikt u ažuriranju težinske matrice. Kapije zaobilaze ovaj problem jer se sa dodavanjem kapija dodaje više težinskih matrica. Sukobljena ažuriranja mogu predstaviti na različitim kapija, pa se tako neće poništiti.

LSTM ćelija se sastoji od ulazne kapije (eng. *input gate*), izlazne kapije (eng. *output gate*) i zaboravne kapije (eng. *forget gate*). Arhitektura LSTM-a i kapije su prikazane na slici 2.3.

Ulazna kapija. Ulazna kapija se dobija u dva koraka. Prvi korak se odnosi na kombinaciju trenutnog ulaznog vektora x_t i izlaznog stanja u prethodnom koraku h_{t-1} . Ovaj korak se izračunava na sledeći način:

$$z_t = \tanh(W_z x_t + R_z h_{t-1} + b_z)$$

Ovdje su W_z i R_z težinske matrice vezane za ovaj korak, dok je b_z bias. \tanh je aktivaciona funkcija.

Drugi korak predstavlja kombinaciju ulaznog vektora x_t , izlaznog stanja u prethodnom koraku h_{t-1} kao i memorijske ćelije u prethodnom koraku c_{t-1} . Sljedeći korak se izračunava:

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + p_i * c_{t-1} + b_i)$$

W_i , R_i i p_i su težinske matrice, dok je $*$ operacija množenja element po element.

Ulazna kapija ima ulogu da odredi koji se podaci mogu dodati u memorijsku ćeliju a koji ne. z_t je predlog podataka koji se dodaju, dok je i_t aktivaciona vrijednost ulazne kapije.

Zaboravna kapija. Zaboravna kapija predstavlja kombinaciju ulaznog vektora x_t , izlaznog stanja u prethodnom koraku h_{t-1} i memorijske ćelije u prethodnom koraku c_{t-1}

na sledeći način:

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + p_f * c_{t-1} + b_f)$$

Zaboravna kapija predstavlja određivanje vrijednosti koja bi trebalo da se ukloni iz prethodnog stanja memorijske ćelije.

Memorijska ćelija. Nakon izračunavanja ulazne kapije i zaboravne kapije vrši se ažuriranje memorijske ćelije. Ažuriranje se vrši na sljedeći način:

$$c_t = z_t * i_t + c_{t-1} * f_t$$

Izlazna kapija. Ova kapija se takođe izračunava u dva koraka. Prvo se izračunava kombinacija ulaznog vektora x_t , izlaznog stanja u prethodnom koraku h_{t-1} i novog stanja memorijske ćelije c_t .

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + p_o * c_t + b_o)$$

Sljedeći korak predstavlja kombinaciju prethodnog koraka i novog stanja memorijske ćelije, i predstavljen je na sljedeći način:

$$y_t = \tanh(c_t) * o_t$$

2.8 Gated recurrent unit (GRU)

Osim LSTM postoje različite mreže koje rješavaju problem nestajućih gradijenata. Ove mreže imaju sličnu strukturu koja posjeduje kapije i memorijsku ćeliju. GRU je jedna od ovih varijacija, međutim ona posjeduje samo strukturu sa kapijama, dok izbacuje potrebu za memorijskom ćelijom.

Za razliku od LSTM koja glavnu memoriju čuva u odvojenoj memorijskoj ćeliji, GRU posjeduje samo klasično skriveno stanje predstavljeno kod standardnih rekurentnih neuralnih mreža. Ovim GRU mreža u svakom koraku kao izlaz pruža čitavo skriveno stanje pa nema mehanizam kojim bi ograničila mjeru u kojem je skriveno stanje predstavljeno. Dok LSTM ima za ovo ima izlaznu kapiju, GRU je nema.

GRU svoje skriveno stanje ažurira preko *kapija za ažuriranje* (eng. *Update gate*) i *kapije za resetovanje* (eng. *Reset gate*). Pomoću ovoga mreža ima mogućnost da nauči da potrebne karakteristike zadrži duže u memoriji, kao i da izbriše nepotrebne karakteristike. Pošto se unutrašnje stanje samo ažurira a ne prepisuje u potpunosti, kao kod običnih neuralnih mreža, ove mreže imaju sposobnost učenja dugih vremenskih zavisnosti.

nad mjerom u kojoj će novo stanje biti dodato a staro zaboravljeno.

$$h_t = z_t h_{t-1} + (1 - z_t) \tilde{h}_t$$

LSTM i GRU mreže iako različite ostvaruju slične performanse na zadatku predviđanja vremenskih serija [26].

Rekurentne neuralne mreže su veoma snažan alat za rad sa vremenskim podacima. Međutim imaju inherentne nedostatke. Pošto su prirodno sekvencionalni sistemi nije ih moguće paralelizovati. Ovo označava da je njihovo treniranje i upotreba značajno sporija. Takođe se može zaključiti da prilikom predviđanja vremenske serije, nakon analize čitave ulazne sekvence, predviđanje se izvršava na zadnjem koraku. Ovo može pokazati usko grlo, pošto se svi dotadašnji koraci moraju reflektovati u zadnjem koraku.

Glava 3

Detekcija objekata na video snimcima

Zadatak kompjuterske vizije, koji predstavlja mogućnost računara da percipira i razumije vizualne podatke, oslanja se na tehnologiju detekcije objekata na slici. Mogućnost detekcije objekata je osnov većine zadataka kompjuterske vizije kao što su: praćenje objekata na video snimku, opisivanje slike, prepoznavanje akcija na slici. Detekcija objekata predstavlja mehanizam za određivanje položaja objekata na slici i tipa objekta. Detekcija objekata se može podijeliti na više specifičnijih pod-zadataka koji podrazumijevaju detekciju lica, detekciju pješaka i detekciju skeletona [27].

Prva istraživanja u oblasti detekcije objekata na slici pojavila su se devedesetih godina prošlog vijeka. Ova istraživanja su se većinski zasnivala na tradicionalnim metodama, gdje su se detektovale ručno izrađene karakteristike. Značajni napredak u ovoj oblasti pojavio se 2010-ih godina primjenom dubokog mašinskog učenja. Od tada je nastao značajan porast istraživanja u ovoj oblasti.

Detekcija objekata se koristi u različitim domenima, pa se na osnovu toga može izvoditi na različite načine. Kod problema kao što su detekcija generičkih objekata koristi se detekcija pomoću graničnih okvira (eng. *bounding box*) koji predstavljaju pravougaonike koji uokviruju detektovani objekat. U detekciju generičkih objekata spadaju detekcija lica i detekcija pješaka. Kod problema detekcije dominantnih objekata upotrebljava se segmentacija slike na nivou piksela. Ovdje se istaknuti objekat prikazuje kao regija čije ivice prate ivice objekta.

3.1 Poznati skupovi podataka i mjera kvaliteta algoritma

Važan zahtjev detekcije objekata na slici je skup podataka. U ovu svrhu razvijeno je dosta skupova podataka sa različitim karakteristikama. Skup podataka korišten u ove svrhe bi trebao da posjeduje dosta podataka sa manje pristrasnosti. Poznati skupovi podataka koji su *Pascal-VOC* [28], *The ImageNet Large Scale Visual Recognition Challenge* [29], *MS-COCO* [30] i *Open images* [31].

Mjera kvaliteta koja se obično koristi je prosječna preciznost (eng. *Average precision AP*). Ona se izračunava za sve klase objekata posebno. Kao ukupna mjera uzima se srednja prosječna preciznost (eng. *Mean Average precision mAP*) koja predstavlja srednju vrijednost AP na svim klasama objekata. AP mjeri preciznost lokalizovanja objekata određivanjem presjeka nad unijom (eng. *Intersection over union IoU*). IoU predstavlja odnos površine koja se nalazi u presjeku detektovanog objekta i stvarnog objekta na slici i površine unije graničnih okvira ova dva objekta. IoU može uzeti vrijednost između 0 i 1. Objekat se računa kao detektovan kada je IoU iznad zadanog praga. Ovaj prag je obično 0.5, dok može ići i do 0.95.

Osim preciznosti algoritama detekcije, veoma važna mjera njihovog kvaliteta je brzina rada. Da bi se detekcija objekata primijenjivala u realnom vremenu i na video snimcima, gdje može biti i po 60 frejmova u sekundi koje treba obraditi, potreban je brz algoritam. Nekada da bi se postigla veća brzina, dolazi do pada u preciznosti.

3.2 Tradicionalni detektori

Prije pojave detektora zasnovanih na konvolucionim mrežama, tradicionalne metode su koristile ručno dizajnirane module. Takođe, ograničenja u računarskoj snazi dovodila su do potrebe različitih metoda ubrzanja.

Jedan od važnih detektora je *Viola Jones* detektor, koji obavlja detekciju lica u realnom vremenu [32]. Ovaj detektor koristi pristup pomjerajućih prozora. Uzimajući prozore različitih veličina i pomjerajući ih kroz sliku, prolazi se kroz sve moguće lokacije objekta na slici. Postigao je velika ubrzanja u odnosu na tadašnje metode, a u nekom obliku se koristi i danas za detekciju lica na uređajima sa ograničenim hardverom, kao što su mobilni telefoni i integrisani sistemi.

Unapređenja u brzini *Viola Jones* detektor postiže unapređenjima kao što su integraciona slika i kaskadna detekcija. Integraciona slika je numerička tehnika ubrzanja. Ona

se bazira na mogućnošću podjele integracije i diferenciranja prilikom konvolucije. Pošto se integracioni dio, odnosno integraciona slika može dobiti veoma brzo (algoritmom kumulativne sume, pošto je piksel integracione slike zbir piksela lijevo i gore od njega u originalnoj slici), ubrzava se ukupno vrijeme konvolucije. Kaskadna detekcija predstavlja metodu dijeljenja detekcije na više etapa, gdje se na svakoj etapi eliminiše veliki broj mogućih prozora brzim metodama detekcije, najčešće klasifikatori Haar-sličnih osobina.

Karakteristike na slikama se mogu predstaviti na različite i robustnije načine. Jedan od načina često korišćenih u tradicionalnim detektorima je *Histogram orijentiranih gradijenata* (eng. *Histogram of oriented gradients* HOG) [33]. HOG detektor dijeli sliku na uniformno raspoređene ćelije, gdje se za svaku računa histogram. Da bi detektor bio robustniji na promjene osvjetljenja i kontrasta, vrši se normalizacija blokova ćelija. Pokretni prozor, koji je predstavljen histogramima ćelija koje obuhvata, je dalje prosljeđen klasifikatoru (najčešće SVM¹). Primjenom praga na izlaz iz klasifikatora određuje se da li je u prozoru detektovan objekat. Da bi imao mogućnost detekcije objekata različite veličine, HOG detektor se primjenjuje na sliku skaliranu više puta.

Jedan od najpreciznijih detektora je *Deformable Part-based Model* DPM [34]. DPM detektor se oslanja na podjelu problema detekcije nekog objekta na detektovanje njegovih djelova. DPM koristi mašinsko učenje za učenje načina detekcije svakog od potrebnih djelova posebno. Nakon toga se primjenjuje pokretni prozor različitih veličina. Za svaki prozor se vrši detekcija svakog od djelova koje čine objekat. Za prozore koje sadrže sve djelove koristi se model koji vodi računa o pozicijama svakog od djelova ispitivajući mogućnost da oni čine objekat.

DPM detektori su prilično robustni i pokazuju kvalitetne rezultate na širokom opsegu objekata. Kvalitetni su u detektovanju kompleksnih objekata na slici. Nedostatak DPM detektora je njegova složenost, koja ga čini sporijim od ostalih detektora, naročito kod detekcije objekata sa dosta djelova.

Detektori koji koriste duboko učenje i konvolucione mreže imaju daleko bolje performanse od tradicionalnih detektora. Međutim oni predstavljaju osnovu za različite tehnike koje imaju uticaja na današnje detektore.

¹Support Vector Machine

3.3 Detektori zasnovani na dubokom učenju

Detektori zasnovani na dubokom učenju, predstavljaju sljedeći korak u razvoju ove tehnologije. Oni predstavljaju rješenja za probleme koji su prisutni u tradicionalnim detektorima. Problemi kao što su potreba ručnog dizajniranja karakteristika koje se koriste prilikom prepoznavanja, slaba generalizacija u različitim scenarijima, problem prepoznavanja objekata različitih veličina i orijentacija, kao i nemogućnost prepoznavanja kompleksnih objekata, su neki od problema koje rješavaju detektori zasnovani na dubokom učenju.

Osnova modela dubokog učenja u detekciji objekata su konvolucione neuralne mreže (eng. *Convolutional neural networks* CNN). Konvolucione mreže su oblik neuralnih mreža kod kojih je neuron u jednom sloju povezan sa grupom susjednih neurona u prethodnom sloju. Ovo predstavlja razliku u odnosu na standardne, potpuno povezane neuralne mreže, gdje je svaki neuron iz jednog sloja povezan sa svakim neuronom u narednom sloju. Konvolucione neuralne mreže uče vrijednosti seta težinskih matrica koje predstavljaju filtere. Ovi filteri vrše konvoluciju sa regionima susjednih neurona prethodnog sloja da bi dobili vrijednosti izlaza iz sloja. Prije konačnog izlaza iz sloja, nad rezultatom konvolucije se primjenjuje nelinearna aktivaciona funkcija.

Konvolucione neuralne mreže kod dubokog učenja se često kombinuju sa agregacionim slojevima (eng. *Pooling layer*). Agregacioni slojevi sumiraju izlaze konvolucionih slojeva. Jedne od najčešćih metoda koje se koriste su: maksimalna agregacija, gdje se kao rezultat uzima maksimalna vrijednost svakog neurona i prosječna agregacija, gdje se uzima srednja vrijednost regiona. Agregacioni slojevi omogućavaju robustnost konvolucionih modela na varijacije, kao i ubrzanje smanjivanjem veličine narednih slojeva.

Konvolucione neuralne mreže primjenju princip hierarhijske reprezentacije karakteristika. Ova reprezentacija se ogleda u tumačenju ulaznih slika u više levela. Leveli hierarhijske reprezentacije idu od najnižih reprezentacija koje su reprezentacije u pikselima i veze između grupa piksela, do viših semantičkih karakteristika kao što su oblici i objekti. Ova reprezentacija se ostvaruje i uči automatski konvolucionalnim neuralnim mrežama, gdje se svaka reprezentacija predstavlja konvolucionim slojevima mreže. Konvolucione mreže imaju veliku moć učenja kao i velike mogućnosti optimizacije.

Nadovezivanjem na konvolucione slojeve, nekoliko potpuno povezanih slojeva moguće je prilagođavanje konvolucionog modela različitim problemima. Optimizacija modela vrši se SGD algoritmom. Prilikom optimizacije moraju se uzeti u obzir dvije funkcije

cijene: funkcija cijene lokalizacije, i funkcija cijene klasifikacije. Funkcija cijene klasifikacije obično se računa pomoću razdaljine koordinata ivica graničnih okvira predviđenih modelom i stvarnih podataka. Funkcija cijene klasifikacije se računa kao kros entropija, koja je standardna kod višeklasne klasifikacije.

Za veću generalizaciju i bolju mogućnost detekcije objekata različitih dimenzija i orijentacija, upotrebljava se augmentacija podataka. Augmentacijom se povećava set podataka za treniranje tako što se slike koje su u njemu mijenjaju, primjenjujući različite rotacije i skaliranje pa dodaju u set podataka za treniranje.

Generička detekcija objekata se može podijeliti na dva glavna pristupa. Prvi pristup su dvofazni detektori, koji prate više tradicionalan proces. Kod ovih detektora se prvo pronalaze predloženi regioni, koji se kasnije klasifikuju. Drugi pristup su jednofazni detektori koji probleme lokalizacije i klasifikacije posmatraju kao jedan klasifikacioni problem.

3.4 Dvofazni detektori

Dvofazni detektori (eng. *two stage detector*) dijele čitav proces detekcije objekata na dvije faze. U prvoj fazi se vrši izdvajanje predloga regiona. Ovi predlozi dijele sliku na regione sa većom vjerovatnoćom da će se u njima naći objekat. Ovo se radi pomoću jednostavnijih algoritama, koji u obzir moraju da uzmu mogućnost različitih dimenzija objekata. Algoritmi koji izvlače predloge regiona moraju biti razumno efikasni, da ne bi pravili zastoj ukupnog procesa. Predlaganje regiona pomaže detektoru da se na predloženim regionima bolje fokusira na detekciju i klasifikaciju objekta.

Na drugoj fazi detekcije se svaki od predloženih regiona obrađuje posebno. Prvo se vrši izvlačenje karakteristika pomoću CNN. Nakon toga se koristi sljedeća mreža koja iz niza izdvojenih karakteristika uči klasu objekta. Osim klase regiona vrši se i prilagođavanje graničnog okvira. Ovo radi mreža koja uči kako da prilagodi granice predloženog regiona da bi se povećao kvalitet lokalizacije objekta.

Dvofazni detektori objekata postali su temelj u kompjuterskoj viziji, pružajući vrhunske performanse u zadacima detekcije objekata. Njihova efikasnost i preciznost učinila ih je nezamjenljivi u primenama kao što su autonomna vozila i nadzorne kamere. Ovaj pristup kod detektora objekata uveden je pojavom detektora R-CNN (eng. *Region-based convolutional neural network*) [35]. R-CNN će biti razmatran u prvom dijelu ove sekcije.

Prva faza ili predlaganje regiona se može podijeliti na dvije grupe algoritama: predlaganje grupisanjem (eng. *Grouping proposal*) i ocjenjivanje prozora (eng. *Window scoring*

proposal). Predlaganje regija grupisanjem počinje izdvajanjem dosta malih regija koji se nazivaju superpikseli. Superpikseli predstavljaju regije sa uniformnom teksturom. Nakon toga vrši se grupisanje svakog od superpiksela sa susjednim superpikselima tako da se grupisu oni superpikseli koji su najsličniji. Ovo grupisanje se nastavlja između dobijenih regiona sve dok se ne dobije region koji pokriva čitavu sliku. Ovo hijerarhijsko grupisanje omogućava detekciju objekata svih veličina. Sličnost regija se određuje karakteristikama čitavog regiona tako da se spajanje ne usložnjava kako se regioni povećavaju. Popularan algoritam koji se koristi u R-CNN i Fast R-CNN je *Selective search* algoritam [36].

Predlaganje regiona grupisanjem prozora počinje sa velikim setom prozora različitih veličina. Nakon toga se svaki prozor ocjenjuje proporcionalno vjerovatnoći da se nalazi objekat u njemu. Prozori sa većom vjerovatnoćom nastavljaju dalje. Lokalizacija ovoga algoritma je manja zbog predefinisanih prozora, ali može biti brži ako se ne zada preveliki broj prozora. *Region proposal network* RPN je drugi algoritam koji koristi konvolucione neuralne mreže za određivanje vjerovatnoće postojanja objekta na nekom regionu. RPN je jedno od glavnih poboljšanja koje je uveo Faster R-CNN detektor [37].

Svaki od izdvojenih regiona se prosleđuje na dalju klasifikaciju i precizniju lokalizaciju. Kod R-CNN detektora na ulaz konvolucione mreže dovodi se ulaz fiksne veličine. Ovo označava potrebu da se regioni različitih dimenzija mijenjaju, odsijecaju ako je region veći ili dopunjavaju ako je manji. Ove operacije mogu dovesti do odsijecanja djelova objekata što utiče na detekciju. Odgovor na ovaj problem je predstavljen u okviru SPP-net detektora koji dodaje *Spatial Pyramid Pooling* SPP sloj [38]. Izlaz konvolucione mreže u oba slučaja je set fiksne veličina koji predstavlja izvučenje karakteristike regiona. R-CNN sadrži dosta redundantnih operacija. Kako se većina predloženih regiona poklapa u nekom dijelu, izračunavanje karakteristika iz preklapljenih djelova predloženih regionima postaje redundantno.

Karakteristike regiona se na kraju upotrebljavaju za klasifikaciju regiona. Klasifikacija se obavlja pomoću SVM-a za klasifikaciju u više klasa. Klase koje detektuje su svi traženi objekti, koji predstavljaju pozitivne klase i pozadina, koja predstavlja negativnu klasu. Nakon klasifikacije vrši se regresija graničnih okvira da bi došlo do bolje lokalizacije.

R-CNN uključuje i korak postprocesiranja. Može se desiti da postoji veći broj detekcija istog objekta koje se u najvećoj mjeri preklapaju. Ovdje se uzima detekcija sa najvećom vjerovatnoćom a ostale se uklanjaju. Ovaj postupak se naziva *Non maximum suppresing*.

Već pomenuti detektor SPP-net osim SPP sloja donosi još jedno poboljšanje. Ovdje se u većoj mjeri uspijeva ukloniti redundantnost izračunavanja karakteristika regiona ko-

ji se preklapaju. Umjesto izračunavanja karakteristika za svaki od predloženih regiona, konvoluciona neuralna mreža se primjenjuje na čitavu sliku. Ovim se dobija mapa karakteristika čiji se dio upotrebljava za svaki od predloženih regiona. SPP-net je 20 puta brži od R-CNN, dok ne smanjuje preciznost sa $mAP = 59.2\%$ na VOC07 skupu podataka [39].

Treniranje R-CNN i SPP-net detektora uključuje više koraka, gdje se nezavisno treniraju konvolucioni sloj, pa klasifikacija i na kraju regresija graničnih okvira. *Fast R-CNN* se bavi ovim problemom, predstavljajući mogućnost istovremenog treniranja klasifikacije objekata i regresije graničnih okvira. Ovo se postiže određivanjem funkcije cijene zajedničke za oba procesa. Funkcija cijene je

$$J = J_{cls} + \lambda [p^t \geq 1] J_{loc}.$$

Ovdje je J_{cls} funkcija cijene klasifikacije i predstavlja se kao

$$J_{cls} = -\frac{1}{N} \sum_{i=1}^N \sum_C p_i^t \ln p_{i,c}.$$

N predstavlja broj predloženih regija. C predstavlja broj klasa, dok su p_i^t i $p_{i,c}$ stvarna klasa i -te predložene regije i predviđena vjerovatnoća klase za i -tu regiju.

J_{loc} je funkcija cijene regresije pozicije i veličine graničnog okvira. Ona se izračunava korišćenjem glatke $L1$ norme. Ova norma se primjenjuje da bi se smanjila osjetljivost na eksplozivne gradijente.

$$J_{loc} = \sum SmoothL1(v_i^t - v_i),$$

$$SmoothL1 = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & |x| \geq 1 \end{cases}. \quad (3.1)$$

Operacija $[p^t \geq 1]$ služi za izostavljanje grešaka u regionima gdje je stvarna klasa pozadina. λ je parametar kojim se kontroliše odnos izmjeđđu grešaka klasifikacije i regresije graničnog okvira.

Fast R-CNN ostvaruje 10 puta bržu detekciju u odnosu na SPP-net, a i popravlja kvalitet detekcije sa $mAP = 70\%$ na VOC07 setu podataka.

Primjenom konvolucione neuralne mreže za prvu fazu detekcije dobija se ranije pomenuti RPN. On je prvi put predstavljen u okviru Faster R-CNN detektora. RPN se oslanja na slojeve konvolucione neuralne mreže pomoću kojih se dolazi do mape karakteristika. Nadogradnjom daljih konvolucionih slojeva na mapu karakteristika dobija se RPN. RPN koristi predefinisane, usidrene okvire (eng. *anchor box*), da bi prešao preko različitih pozicija na čitavoj slici. Za svaki od okvira RPN mreža predviđa dvije vrijednosti: objektnost tj. da li je u okviru objekat i pomjeraj graničnog okvira u odnosu na predefinisani okvir.

Čitava mreža, predlozi regiona i konačna detekcija i klasifikacija, se trenira zajedno kod Faster R-CNN detektora. Ovaj detektor je prvi iz ove klase koji ostvaruje detekciju blizu realnog vremena sa 17 fps². Takođe ima visoku preciznost postžući rezultate od $mAP = 73.2\%$ na VOC07 setu podataka.

Prethodno pomenuti detektori koriste mape karakteristika sa samo jednog nivoa, posljednjeg nivoa. Konvoluciona neuralne mreže prirodno, propagacijom unaprijed pravi piramidu mapi karakteristika različitih nivoa. *Feature pyramid network* FPN koristi vrijednosti sa više slojeva konvolucione neuralne mreže da bi ostvario bolju detekciju [40]. Pošto nivoi odgovaraju skalama i veličinama na slici, korištenjem više nivoa karakteristika omogućava se bolje detektovanje objekata različitih veličina. Pošto su na početnim slojevima predstavljene karakteristike bliske levelu piksela, početni slojevi se ne koriste u piramidi. Obično se koristi od srednjeg sloja, pa naviše.

3.5 Jednofazni detektori

Uporedo sa razvojem dvofaznih detektora odvijao se i razvoj jednofaznih detektora. Ovi detektori imaju potpuno drugačiji pristup detekciji objekata. Jednofazni detektori razvijaju jedinstvenu mrežu koja istovremeno predviđa i granične okvire i klase objekata. Jednofazni detektori su znatno brži od dvofaznih jer nemaju potrebu za više faza detekcije, koje ih usporavaju. Jednofazni detektori kao što su *You only look once* YOLO [41] i *Single shot detector* SSD [42] postižu detekciju u realnom vremenu, sa padom u preciznosti detekcije. Obično imaju problem sa detekcijom malih objekata kao i sa detekcijama objekata u grupi.

²Frejмова по секунди (eng. *frames per second*)

Najznačajniji detektor iz ove klase detektora je YOLO detektor. Vremenom je ovaj detektor unaprjeđivan tako da trenutno postoji više verzija. U trenutku pisanja rada postoji osam verzija YOLO-a od v1 do v8, dok svaka verzija ima više modela sa različitim odnosom preciznosti i brzine.

YOLO koristi konvolucionu neuralnu mrežu nad cijelom slikom. Svaki granični okvir koristi sve izvučene karakteristike sa slike za predikciju i klase i graničnog okvira. Predikcija svih graničnih okvira se odvija istovremeno. YOLO dijeli sliku na mrežu ćelija $S \times S$. Za svaku od ovih ćelija se predviđa B graničnih okvira. Granični okviri su na izlazu predstavljeni sa pet parametara. Parametri predstavljaju koordinate centra graničnog okvira (x, y) u odnosu na ćeliju, njegovu visinu i širinu (h, w) u odnosu na cijelu sliku i pouzdanost. Pouzdanost predstavlja mjeru sigurnosti modela da li se na odgovarajućoj ćeliji nalazi objekat. Ova pouzdanost se može predstaviti sa mjerom IoU u odnosu na stvarni objekat.

Za svaku ćeliju YOLO detektora se takođe određuje klasa kojoj pripada. Ovo je predstavljeno sa nizom od C vrijednosti, gdje svaka vrijednost predstavlja vjerovatnoću za svaku od mogućih klasa objekata. Izlaz iz YOLO detektora je tenzor sa dimenzijama $S \times S \times (5B + C)$.

Slično kao i kod dvofaznih detektora, YOLO kao i ostali jednofazni detektori koriste korak postprocesiranja. Zbog mogućnosti višestruke detekcije istog objekta, vrši se odbacivanje svih osim najsigurnije predikcija, predikcije sa najvećom pouzdanošću.

Originalni YOLO v1 predstavlja dva detektora, YOLO i Fast YOLO. Arhitektura YOLO detektora se sastoji od 24 konvoluciona sloja i dva potpuno povezana sloja. Ovaj detektor radi na 45 fps i postiže tačnost od $mAP = 63.4\%$ na VOC07 skupu podataka. Fast YOLO funkcioniše na nešto jednostavnijoj arhitekturi, sa 9 konvolucionih slojeva umjesto 24. Ovaj detektor ostvaruje brzinu detekcije sa 155 fps sa rezultatom od $mAP = 52.7\%$ na VOC07 setu podataka.

Funkcija cijene koja se optimizuje kod YOLO detektora je:

$$\lambda_{cord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} ((x_i - x_i^t)^2 + (y_i - y_i^t)^2) +$$

$$\lambda_{cord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} ((\sqrt{w_i} - \sqrt{w_i^t})^2 + (\sqrt{y_i} - \sqrt{y_i^t})^2) +$$

$$\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - C_i^t) +$$

$$\lambda_{cord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - C_i^t) + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in \text{classes}} (p_i(c) - p_i^t(c))^2.$$

Ovdje 1_i^{obj} označava da li se objekat detektovao u i -toj ćeliji, a 1_{ij}^{obj} da li se j -ti granični okvir odnosi na tu detekciju. Grške koje se smanjuju se samo odnose za slučajeve kada se objekat nalazi u posmatranoj ćeliji. Takođe se posmatraju samo greške za predikciju koja ima najveći IoU za određeni objekat.

Single shot multiBox detector SSD je drugi važan jednofazni detektor. Ovaj detektor koristi različite metode kombinovane sa arhitekturom sličnom YOLO detektoru.

Umjesto mreže koju YOLO primjenjuje, SSD detektor primjenjuje metodu sa usidrenim okvirima. Ova metoda predefiniše set ćelija koje se koriste, ali su one različitih oblika i dimenzija. Ovo omogućava SSD detektoru bolju detekciju objekata različitih veličina.

Osim ovoga SSD primjenjuje i FPN (*Feature pyramid network*) gdje se prilikom detekcije i klasifikacije koriste karakteristike iz više slojeva konvolucione mreže.

SSD ima slične nedostatke kao YOLO, prilikom detekcije sitnih objekata. Omogućava detekciju u realnom vremenu sa brzinom rada od 59 fps. Postoji nekoliko varijanti i poboljšanja SSD detektora, uključujući SSDLite za mobilne aplikacije i različite arhitekture, na primer MobileNet-SSD, radi postizanja ravnoteže između brzine i preciznosti.

SSD se široko koristi u raznim aplikacijama za detekciju objekata, uključujući detekciju pješaka, prepoznavanje saobraćajnih znakova, prepoznavanje lica i mnoge druge.

Jednofazni detektori su veoma efikasni i brzi, međutim kaskaju u preciznosti detekcije. Ova preciznost ima veze sa balansom klasa. Naime većina ćelija koje su predložene u jednofaznom detektoru će biti pozadina, odnosno klase će imati vrijednost nula. Većina ovih regiona su jednostavni za klasifikaciju pa ne nose novu informaciju, a mogu da utiču na učenje modela.

Način efikasnijeg učenja je predložen u okviru RetinaNet detektora uvođenjem nove funkcije cijene pod nazivom *focal loss* [43]. Ova funkcija cijene smanjuje uticaj izračunavanja jednostavnih klasa, gdje su pouzdanosti velike, a stavlja fokus na težim klasifikacijama. Dodavanjem faktora $(1 - p_t)^\gamma$ na krosentropiju dobija se *focal loss*

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log p_t.$$

Ovdje je γ parametar fokusiranja i mora zadovoljavati $\gamma \geq 0$. Ovo dovodi do smanjenja funkcije cijene u slučajevima kada je klasifikacija dobra.

Razvoj različitih djelova mreže jednofaznih detektora, doveo je do prepoznavanja tri dijela mreže: oslonac, vrat i glava. Oslonac se odnosi na konvolucionu neuralnu mrežu koja služi za izdvajanje karakteristika iz slike. Vrat služi za povezivanje oslonca i glave. Odnosi se metode prerade karakteristika kao što su FPN. Glava se odnosi na mreže koje iz karakteristika uče predikciju klasifikacije i pozicije graničnih okvira.

Kao što je napomenuto, YOLO detektor je nadograđivan više puta pa postoji više verzija. Poboljšanja kao što su upotreba usidrenih okvira i FPN mreža je primijenjena u YOLOv2. Ovo je pomoglo prilikom detekcije malih objekata i objekata različitih veličina.

YOLO v3 donosi novu arhitekturu oslonca mreže. Produbljujući konvolucionalnu neuralnu mrežu na 53 sloja dobija se bogatija reprezentacija karakteristika sa slike. Dodat je i SPP sloj. Ova verzija takođe vrši detekciju na tri nivoa, uzimajući izlaze tri različita sloja konvolucione neuralne mreže. Za razliku od prethodne verzije, YOLO v3 odvaja različite glave za različite nivoe.

YOLO v4 donosi nove izmjene u osloncu. Zadržava se ista arhitektura konvolucione mreže sa dodatkom međufaznih konekcija. Takođe se zamjenjuje dosadašnja ReLU aktivaciona funkcija sa Mish aktivacionom funkcijom³. YOLO v4 takođe uvodi novu funkciju cijene CIoU (*Complete intersection over union*). Ona pomaže pri preciznosti određivanja pozicije graničnog okvira i predstavljena je u sljedećem obliku

$$CIoU = IoU - \frac{d_c^2}{c^2} - \frac{a}{\alpha}(1 - IoU).$$

Ovdje je uvrštena distanca između centara predviđenog i stvarnog objekta d_c . Osim ovoga upotrijebljene su druge tehnike poboljšanja efikasnosti detektora.

Kasnije verzije YOLO-a kao što je v8, prelaze na model bez usidrenih okvira. Ovdje se oslobađa potrebe pretpostavki regiona. Detekcija se vrši direktnim predviđanjem centralne tačke objekta, pa regresijom u određivanju tačnih dimenzija.

Primjenom novih modela dubokog učenja koji su se razvili u obradi prirodnog jezika, transformera, razvila se i nova metoda detekcije *Data-efficient Transformer DETR* [44]. Transformeri su modeli sa inovativnom arhitekturom baziranom na mehanizmu samo-pažnje (eng. *self attention*) koja omogućava modelima da efikasno modeliraju zavisnosti

³Mish aktivaciona funkcija je predstavljena sa:

$$Mish(x) = x \tanh(\ln 1 + e^x)$$

između elemenata sekvence, čineći ih izuzetno sposobnim za različite zadatke obrade sekvenci, uključujući prevod teksta, generisanje teksta kao i u ovom slučaju, prepoznavanje objekata u slikama.

DETR nakon konvolucione mreže, umjesto potpuno povezanih mreža ili drugih metoda koristi transformer. DETR pristupa detekciji objekata kao problemu predviđanja skupa. On direktno predviđa broj i pozicije objekata u slici, što ga čini izuzetno prilagodljivim i sposobnim da se nosi sa različitim veličinama i količinama objekata. DETR koristi mehanizam samo-pažnje u transformatorima kako bi uhvatio zavisnosti na velikim udaljenostima i kontekstualne informacije, što može biti ključno za prepoznavanje objekata u složenim scenama.

Glava 4

Procjena brzine vozila

Duboko učenje sve više igra ključnu ulogu u razvoju različitih tehnologija, uključujući i procjenu brzine vozila. Procjena brzine vozila na video snimku pripada oblasti kompjuterske vizije. Kompjuterska vizija se u mnogome oslanja na duboko učenje. Dosadašnja istraživanja u oblasti procjene brzine vozila pomoću video snimka takođe primjenjuju ove metode u nekoj mjeri. Najčešće je to prilikom detekcije vozila na slici. Ovim istraživanjem je razvijena metoda koja je u potpunosti zasnovana na modelima dubokog učenja.

Primjena dubokog učenja za procjenu brzine vozila donosi brojne prednosti, uključujući veću preciznost, pouzdanost, brzu obradu podataka, skalabilnost, fleksibilnost te mogućnost integracije s drugim tehnologijama. Ove prednosti čine duboko učenje atraktivnom opcijom za razvoj metoda za procjenu brzine vozila.

Duboko učenje omogućava razvoj modela koji su otporni na različite izazove u realnim okolnostima, kao što su loši vremenski uslovi ili razne refleksije svjetla. Ovi uslovi mogu uticati na kamere i smanjiti preciznost tradicionalnih metoda koje mogu biti osjetljive na takve faktore. Duboko učenje može naučiti kako se nositi sa različitim okolnostima i pružiti pouzdanu procjenu brzine vozila. Treniranje modela na setu podataka sa širokim spektrom uslova, kao i njegova sposobnost učenja složenih karakteristika, može mu omogućiti pouzdanu procjenu u različitim uslovima.

Duboko učenje može se ubrzati korištenjem grafičkih procesora (GPU-ova) i posebnih čipova za obradu podataka. Ovo omogućava brzu obradu video materijala sa kamera za nadzor saobraćaja, čime se omogućava trenutna procjena brzine vozila. Ova brza obrada ključna je za primjene poput nadzora saobraćaja i sigurnosti na putu.

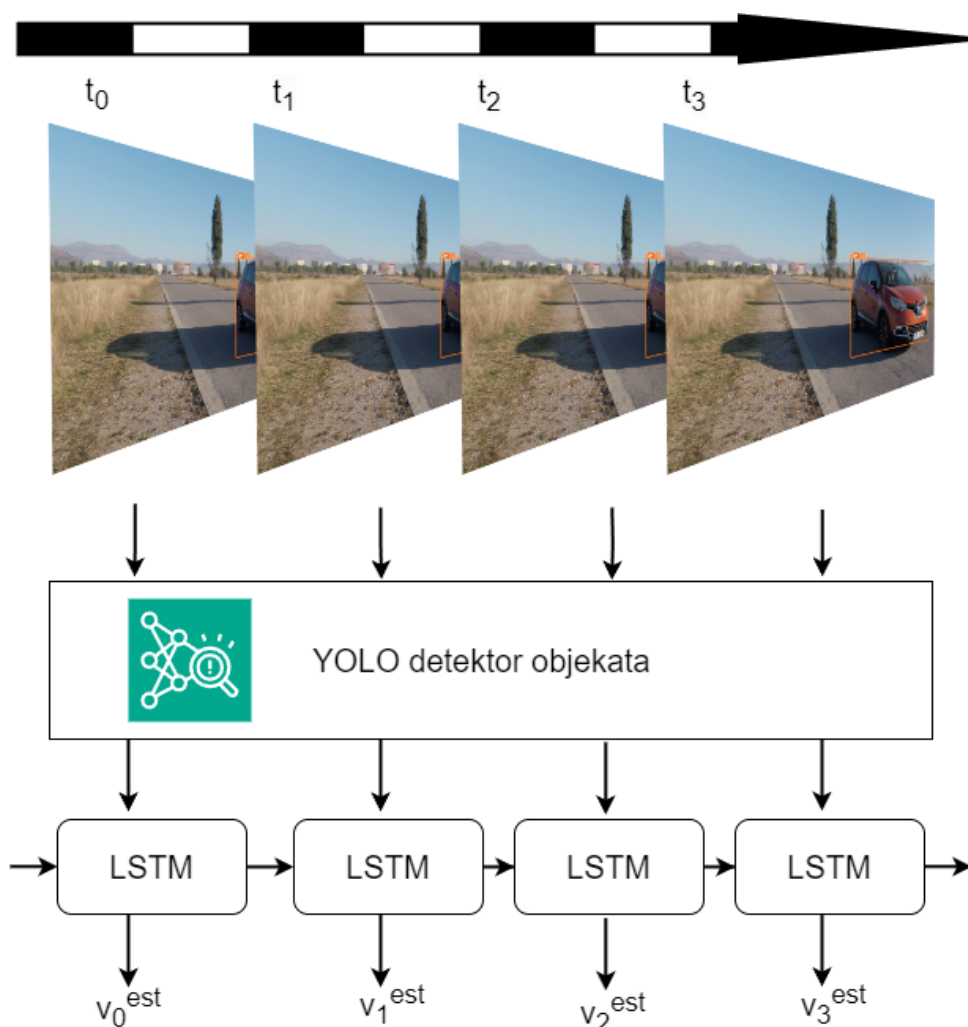
4.1 Predložena metoda

Cilj istraživanja ovog rada je razvoj metode procjene brzine vozila na video snimku koja će biti u potpunosti zasnovana na dubokom učenju. Ovaj pristup procjeni brzine vozila treba da omogući metodu koja koristi jednostavne i široko zastupljene u saobraćaju, monokularne kamere, bez potrebe za procesima kalibracije i specijalnim prilagođavanjima kamere različitim okolinama. Isključivanje svih spoljašnjih karakteristika od kojih zavisi procjena, omogućava razvoj metode koja radi u opštim uslovima. U ovom radu razvijena je ovakva metoda. U cilju razvoja precizne procjene, posmatrana je situacija procjene brzine jednog vozila na snimku.

Predstavljena metoda procjene brzine sastoji se iz više faza. Prva faza predstavlja detekciju vozila na video snimku. U ovoj fazi se koriste detektori objekata na slici. Detektor na kojem je zasnovana ova metoda je YOLO detektor, koji je detaljno pojašnjen u prethodnom poglavlju. Ovaj detektor ima visoku preciznost detekcije, kao i brzinu detekcije. Već istrenirani YOLO detektor će biti dotreniran na setu podataka za detekciju vozila. Set podataka koji je korišten za dotreniranje detektora je [45]. Izlaz ovog detektora je pozicija graničnog okvira u kojem se vozilo nalazi. Pozicija okvira predstavljena je koordinatama njegovog centra i podacima o širini i visini. Izlaz koji govori o klasifikaciji objekta ovdje nije bitan, kako je od interesa samo jedna klasa, vozila.

Sljedeća faza metode predstavlja procjenu brzine vozila iz podataka dobijenih pomoću YOLO detektora. Za ovu fazu dizajnirana je i trenirana neuralna mreža. Procjena brzine vozila je problem koji se ne može izračunati u jednom trenutku. Kako je brzina definisana kao distanca koju vozilo pređe u nekom vremenskom intervalu, jednostavno je primijetiti da je za procjenu brzine vozila sa video snimka potrebna obrada više njegovih frejmova. U ovu svrhu se u metodi primjenjuje rekurentna neuralna mreža, koja obrađuje vremensku seriju podataka koje detektuje YOLO detektor, primijenjen na svakom frejmu videa. Mogućnost rekurentne mreže da radi na vremenskim podacima i pamti vremenske zavisnosti je ključna u ovoj fazi metode.

Kako je za procjenu brzine potreban veći broj vremenskih trenutaka ili frejmova, ona u principu ne može biti dovoljno precizna u previše kratkim intervalima. Zbog ovoga se radi grupisanje više frejmova. Procjena se radi tek nakon prolaska određene grupe frejmova.



Slika 4.1: Princip rada predložene metode procjene brzine

Na slici 4.1 je prikazan čitav proces detekcije brzine vozila. U prvom dijelu je prikazana prva faza metode. Ovdje su izdvojeni nekoliko frejmova iz korištenog skupa podataka. Na frejmovima je prikazan granični okvir koji je detektovao YOLO detektor. Detektovani podaci predstavljaju karakteristike koje su ulaz druge faze metode tj. rekurentne mreže. Na slikama su prikazane značajne karakteristike i njihova promjena tokom video snimka. Karakteristike su x i y koordinate kao i površina graničnog okvira. O odabiru ovih karakteristika biće više riječi u narednom poglavlju.

4.2 Detekcija vozila i ekstrakcija potrebnih podataka

Jedna od osnovnih zahtjeva metode procjene brzine da bi ona bila primjenljiva je brzina procjene. Brzina procjene mora biti visoka kako bi ona bila primjenljiva u realnom vremenu. Ovo postavlja zahtjev za brzom metodom detekcije objekata na videu, da ona ne bi bila zagušenje u sistemu.

Faza za detekciju objekata igra ulogu u izvlačenju karakteristika pomoću kojih se procjenjuje brzina u sljedećoj fazi. Ove karakteristike su promjena koordinata vozila na slici kao i promjena dimenzija vozila na slici. Preciznost detekcije umnogome utiče i na preciznost same procjene brzine.

Zbog ovih zahtjeva je kao detektor objekata odabran YOLO detektor. Verzija koja je korištena u ovom istraživanju je YOLOv5.

4.3 Vremenska serija i rekurentna neuralna mreža

Izlaze prve faze je potrebno obraditi u drugoj fazi metode, procjena brzine. Pošto je brzina problem promjene u više trenutaka, prethodne izlaze prve faze treba pamtit i porediti sa novim izlazima. Ovu ulogu imaju rekurentne neuralne mreže.

Predložena metoda u drugoj fazi koristi duboke rekurentne neuralne mreže za predikciju brzine. Predikcija se može dobiti nakon svakog vremenskog koraka, ili svakog frejma.

Rekurentne mreže su naročito efikasne u učenju vremenskih zavisnosti što je veoma važno prilikom procjene brzine. Rekurentne mreže se mogu nadovezati sa nekoliko slojeva što čini mrežu dubokom i povećava moć učenja mreže.

4.4 Skup podataka

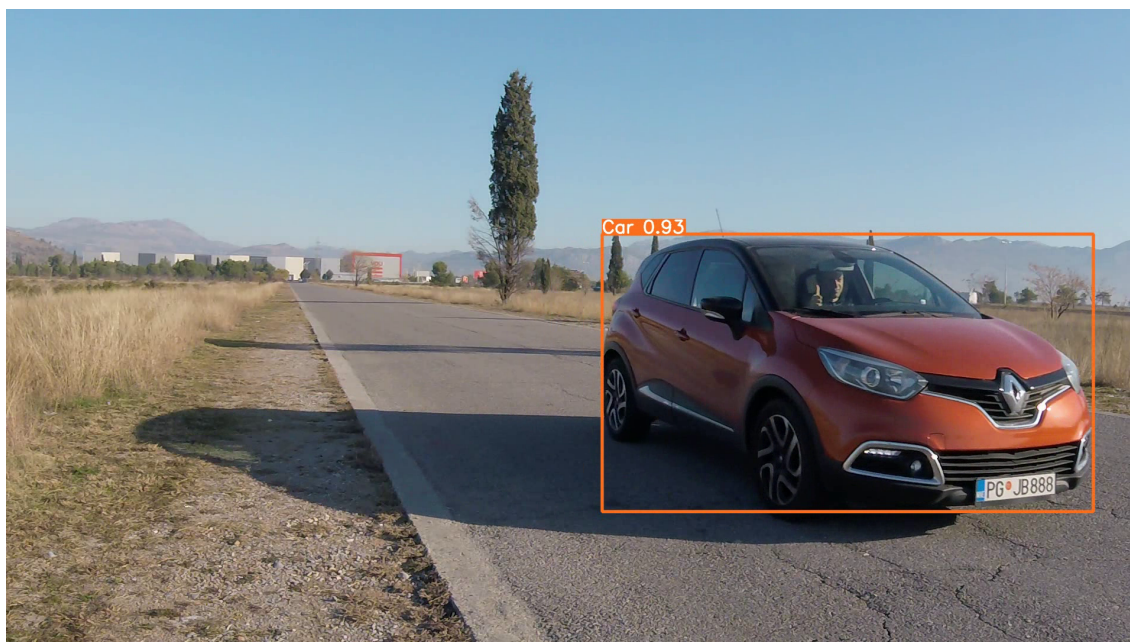
Skup podataka na kome se vrši treniranje i ocjena kvaliteta metode procjene brzine jedan je od ključnih aspekata razvoja metoda procjene. Različiti setovi podataka se koriste u razvoju ovih metoda kako bi se obezbjedila precizna i pouzdana detekcija. Različiti setovi podataka su sakupljeni u ovu svrhu.

Jedni od najznačajnijih skupova podataka su AI City Challenge i BrnoCompSpeed skupovi. AI city Challenge je dio godišnjeg izazova koji je prvobitno pokrenula kompanija NVIDIA 2017. godine. Cilj ovog izazova je unaprjeđenje istraživanja i razvoja u oblasti inteligentne analize video materijala za različite potrebe u pametnim gradovima. Svake godine, dostupan broj video materijala i uzoraka varira. Izazov u 2018. godini fokusirao se na probleme koji uključujući procjenu brzine vozila. Za potrebe ovog izazova, dostavljeno je 142 videa različitih rezolucija i dužina, ali nije bilo dostupnih tačnih vrijednosti brzine vozila.

BrunoCompSpeed je skup podataka koji sadrži 21 video snimak u punoj HD rezoluciji, svaki sa trajanjem od otprilike 1 sat, snimljenih na šest različitih lokacija. Vozila u ovim video zapisima detaljno su označena sa preciznim mjerenjima brzine, prikupljenim pomoću LiDAR tehnologije, a ova mjerenja su potvrđena kroz upoređivanje sa više referentnih GPS mjerenja.

Set podataka primijenjen u ovom radu je VS13 skup [46]. Ovaj skup podataka je namijenjen za treniranje modela neuralnih mreža u svrhu detekcije brzine vozila. VS13 sadrži 400 snimaka dužine 10 sekundi u kojima je uhvaćen prolazak jednog vozila. Kako je orjentisan na preciznosti detekcije brzine vozila, na svakom snimku se nalazi samo jedno vozilo.

U VS13 setu podataka se nalaze snimci 13 različitih vozila. Vozila su različitih karakteristika. Razlikuju se po proizvođaču, vrsti motora, vrsti goriva, vrsti mjenjača i godini proizvodnje. Na svakom automobilu se nalazi tempomat koji brzinu drži konstantnom na svakom od video snimaka. Snimci su snimljeni u urbanom okruženju, na putu bez vozila u pozadini. Brzine vozila u skupu podataka se kreću od 30 km/h do 105 km/h.



Slika 4.2: Slika sa snimka iz VS13 skupa podataka

U tabeli 4.1 su podaci o svim vozilima u skupu podataka. U lijevoj koloni se može vidjeti tip svakoga od vozila koji se nalaze u skupu podataka. U desnoj koloni se nalaze brzine vozila na kojima su snimljeni njihovi prolasci.

Snimci u skupu podataka prikupljeni su korištenjem GoPro Hero5 Session kamere. Kamera je postavljena na udaljenosti od otprilike 0.5 metara iznad tla i na visini od 1.2 metra. Ovaj skup podataka snimljen je duž lokalnog puta ukupne dužine od 622 metra, udaljenog 90 metara od glavnog puta koji povezuje dva crnogorska grada: Podgoricu i Petrovac.

Odabrani put ima nekoliko ključnih prednosti. Njegova značajna dužina omogućava vozačima postizanje stabilnih brzina pri prolasku pored kamere za mjerenje brzine. Takođe, put je dovoljno izolovan od drugih saobraćajnica i izvora smetnji, što rezultira pouzdanim mjerenjima brzine bez većih ometanja. Ova karakteristika omogućava preciznu analizu i evaluaciju snimljenih prolazaka vozila.

Tabela 4.1: VS13 vozila i brzine na snimcima u setu podataka

Vozilo	Brzine (km/h)
Citroen C4 Picasso 1.6 HDI (CitroenC4Picasso)	35, 38, 41, 44, 48, 51, 54, 57, 59, 63, 65, 68, 72, 74, 78, 80, 83, 85, 87, 92, 94, 96, 101
Kia Sportage 1.6 GDI (KiaSportage)	31, 33, 35, 38, 41, 44, 46, 48, 51, 53, 55, 58, 61, 63, 65, 68, 69, 72, 74, 77, 78, 80, 83, 85, 86, 89, 91, 93, 96, 98, 100, 103, 105
Mazda 3 Skyactive (Mazda3)	30, 33, 35, 38, 40, 43, 45, 47, 50, 52, 55, 57, 60, 62, 64, 67, 70, 72, 75, 79, 81, 84, 86, 88, 90, 92, 94, 96, 99, 101, 103, 105
Mercedes AMG 550 (MercedesAMG550)	30, 33, 35, 38, 40, 42, 45, 47, 50, 52, 55, 58, 60, 62, 65, 67, 70, 73, 75, 78, 80, 82, 85, 87, 90, 93, 95, 98, 100, 105
Mercedes GLA 200D (MercedesGLA)	30, 33, 36, 39, 41, 42, 45, 47, 48, 49, 52, 54, 55, 59, 61, 63, 65, 68, 70, 72, 75, 78, 81, 83, 85, 88, 90, 92, 93, 96, 100, 101, 103, 104
Nissan Qashqai 1.5 DCI (NissanQashqai)	35, 38, 40, 42, 45, 48, 50, 53, 55, 58, 60, 61, 64, 65, 68, 70, 73, 75, 78, 80, 82, 85, 88, 90, 93, 94, 96, 98, 102
Opel Insignia 2.0 CDTI (OpelInsignia)	31, 35, 38, 41, 44, 47, 50, 53, 55, 58, 61, 64, 66, 68, 70, 72, 73, 76, 78, 80, 83, 86, 89, 91, 94, 97, 100
Peugeot 208 1.4 HDI (Peugeot208)	30, 32, 34, 37, 40, 43, 45, 47, 50, 51, 54, 57, 60, 62, 64, 67, 68, 71, 73, 76, 77, 79, 82, 84, 87, 90, 92, 95, 96
Peugeot 3008 1.6 HDI (Peugeot3008)	40, 43, 45, 47, 50, 52, 54, 55, 56, 58, 60, 61, 63, 65, 67, 68, 70, 72, 74, 75, 78, 80, 83, 85, 87, 89, 90, 92, 95, 97, 100
Peugeot 307 2.0 HDI (Peugeot307)	30, 33, 35, 38, 40, 43, 45, 47, 48, 50, 53, 56, 59, 60, 63, 66, 69, 72, 73, 76, 79, 82, 85, 88, 91, 94, 97, 101, 103
Renault Captur 1.5 DCI (RenaultCaptur)	30, 33, 36, 38, 40, 41, 44, 46, 47, 48, 50, 52, 56, 58, 60, 63, 66, 68, 70, 72, 76, 78, 80, 83, 86, 88, 90, 92, 94, 97, 98, 100, 102
Renault Scenic 1.9 DCI (RenaultScenic)	30, 35, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 57, 60, 62, 64, 66, 68, 70, 71, 72, 74, 75, 77, 80, 82, 84, 86, 87, 90, 91, 94, 95, 98, 101
VW Passat B7 1.6 TDI (VWPassat)	30, 35, 39, 40, 42, 45, 47, 49, 50, 52, 54, 55, 57, 60, 61, 64, 65, 67, 70, 71, 72, 73, 75, 78, 80, 81, 82, 85, 88, 90, 91, 94, 96, 98, 100

4.5 Treniranje modela

Skup podataka VS13 je dizajniran specifično za detekciju brzine vozila čime se bavi ovaj rad. Međutim ima i nedostatke. Glavni nedostatak je veličina skupa podataka. Za potrebe dubokog učenja VS13 je prilično mali skup podataka. Treniranje na malim skupovima posjeduje izazove koji se javljaju kod generalizacije modela i pretreniranja.

Postoje različite metode kojim se kvalitet treniranja na malom skupu podataka može povećati. Veoma je važno obezbijediti kvalitet podataka na kojima se trenira. Zbog ovoga se primjenjuje čišćenje podataka od izuzetaka i nedostajućih vrijednosti. U ovom istraživanju ovo je primijenjeno na podacima dobijenim nakon YOLO detektora.

Izlazi YOLO detektora se upotrebljavaju kao podaci na kojima se trenira rekurentna neuralna mreža. U ovim izlazima nekada je moguće da se dese nepravilnosti, obično da izostane detekcija vozila na nekom frejmu. Ovo dovodi do prekida u vremenskoj seriji što bi dovelo do otežanog učenja. Ovo je riješeno primjenom linearne interpolacije na ovim tačkama. Ovako je nedostajuća vrijednost popunjena na osnovu prethodnih i sljedećih podataka uzimajući njihovu srednju vrijednost. U okviru pripreme podataka izvršena je i normalizacija podataka, gdje su vrijednosti na ulazu skalirane tako da se nalaze u rasponu od 0 do 1.

Osim ovoga, na malim setovima podataka, koriste se i tehnike augmentacije podataka koje omogućuju generiranje dodatnih primjera iz postojećih podataka. Na primjer, za slikovne podatke to može uključivati rotaciju, promjenu osvjetljenja i kontrasta. Odabir modela je važan za mali skup podataka. Složeni modeli s velikim brojem parametara mogu brzo dovesti do pretreniranja. Česta je i upotreba prethodno treniranih modela (*transfer learning*) gdje se oni koriste kao polazna tačka nakon čega se vrši fino podešavanje (*fine-tuning*) ili dotreniranje modela na novim podacima. U ovom istraživanju ovo je urađeno primjenom YOLO detektora.

Kros-validacija je važna tehnika u mašinskom učenju koja pomaže u procjeni performansi modela. Prilikom razvoja modela koji će trenirati na malom setu podataka, važno je imati pouzdane metode za mjerenje njihove uspješnosti. Kros-validacija omogućuje da se objektivno procijeni kako će model raditi na novim podacima.

Osnovna ideja kros-validacije je podjela dostupnih podataka na skupove za trening i testiranje kako bi se bolje razumjela generalizacija modela na novim podacima. Postupak se obično sastoji od nekoliko koraka.

Prvo se skup podataka dijeli na skup za treniranje i skup za testiranje ili validaciju.

Ovaj skup za validaciju ne predstavlja konačnu evaluaciju modela nego korak u treniranju. Nakon ovoga model se trenira na skupu za trening kako bi naučio odnose između ulaza i ciljne varijable. Nakon što je model treniran, testira se na skupu za validaciju kako bi se procijenile njegove performanse. Postupak se ponavlja više puta s različitim podjelama podataka kako bi se dobila stabilna procjena performansi modela.

Postoji više načina na koje je moguće uraditi. Metod koji se primjenjuje u ovom istraživanju je K-struka kros-validacija. K-struka kros-validacija (*K-fold Cross-Validation*) dijeli skup podataka na K podskupova od kojih se jedan koristi kao testni skup, dok se ostali koriste za trening. Postupak se ponavlja K puta, svaki put s drugačijim testnim skupom. Na kraju se uzima prosječna vrijednost svih iteracija.

Drugi način je metoda *Holdout validation*. Ovdje se skup podataka jednom podijeli na skup za trening i skup za testiranje, pri čemu se obično veći dio podataka koristi za trening. Ova metoda je brža od K-struke kros-validacije, ali može biti manje pouzdana s manjim skupovima podataka.

Leave-One-Out Kros-validacija je metod gdje se svaki podatak koristi kao testni skup jednom, dok se ostali koriste za trening. Ovo je ekstremni oblik K-struke kros-validacije s K jednakim broju primjera u skupu podataka.

Grupna Kros-validacija (*Stratified Cross-Validation*) se koristi kada su podaci neuravnoteženi, tj. kada postoje različite klase koje nisu ravnomjerno zastupljene u skupu podataka. Pomaže u osiguravanju da svaki skup za trening i testiranje zadržava proporcionalno zastupljene klase.

Kros-validacija omogućava maksimalno iskorištavanje svakog dostupnog podataka u obuci i testiranju modela. Ovo je posebno važno kada je ograničen broj primjera. S malim skupovima podataka, jedna podjela na trening i test može imati veliki uticaj na rezultate. Kros-validacija ponavlja taj postupak više puta s različitim podjelama, što daje stabilniju procjenu performansi.

Glava 5

Eksperimentalni rezultati

U ovom poglavlju će biti predstaviti rezultati eksperimenata predložene metode za procjenu brzine. Kroz niz eksperimenata će biti prikazan razvoj metode i odabrani parametri iste. Poređenjem različitih varijanti, biće prikazane karakteristike metode i njeno ponašanje u različitim okolnostima.

Ovde će biti pojašnjeno kako su postavljeni parametri metode. U prvom delu će biti predstavljena postavka eksperimenta, uz objašnjenje načina ocenjivanja kvaliteta metode.

U sledećem delu će biti predstavljeni eksperimenti sa različitim vrstama mreža. Izaćemo rekurentnu mrežu, konkretno LSTM ili GRU, i prikazaćemo eksperimente sa arhitekturom mreže, određivanjem složenosti (broj slojeva, regularizacija, itd.).

Nakon toga, usmerićemo pažnju na karakteristike dobijene YOLO detektorom. Analiziraćemo uticaj svih karakteristika, uključujući i brzinu procene. Definisaćemo dužine sekvenci na kojima se vrši obuka, kao i optimalne uslove za procenu. U poslednjem delu biće predstavljeni rezultati konačne metode.

5.1 Postavka eksperimenta

Svaka od varijacija modela za procjenu brzine vozila je testirana na sljedećem eksperimentu. Prvo se vrši podjela skupa podataka na dio za treniranje i za testiranje. Dio za treniranje se sadrži od svih snimaka, na različitim brzinama, jednog vozila. Ostatak vozila u skupu podataka se koristi za treniranje. Skup za treniranje se dalje dijeli na skup za validaciju i za trening. Ova podjela se vrši nakon nasumičnog miješanja podataka. Podjela je u odnosu 80% za treniranje i 20% za validaciju.

Ovaj postupak se ponavlja tako da je svako od vozila u skupu podataka u jednom trenutku u skupu za testiranje. Svako od ovih ponavljanja se vrši više puta kako bi se dobila stabilnija procjena performansi metode. U ovom istraživanju se vrši po 5 ponavljanja za svako vozilo kao testni skup podataka.

Ukupna greška se računa nakon procjena na svim sekvencama, na svim ponavljanjima. Greška se računa na ponavljanjima za svako vozilo posebno. Takođe se računa i ukupna greška na svim procjenama na svim vozila kao testnim podacima.

5.2 Mjera kvaliteta

Pouzdana mjera kvaliteta je vrlo važna prilikom razvoja modela. Mjerom kvaliteta se dobija ocjena performansi različitih varijacija modela. Poređenjem ovih ocjena može se vidjeti kako se varijacije modela ponašaju u različitim okolnostima, kao i odrediti optimalni oblik modela.

Kao mjera kvaliteta u ovom istraživanju koristi se korijen srednje kvadratne greške (eng. *root mean squer error RMSE*). RMSE se predstavlja u obliku:

$$RMSE = \sqrt{\frac{1}{L} \sum_{l=1}^L (v_l^{est} - v_l^{true})^2}$$

Ovdje L predstavlja broj sekvenci na kojima je izvršena procjena, a v_l^{est} procijenjenu brzinu, dok v_l^{true} stvarna vrijednost u l -toj procjeni. Korijen srednje kvadratne greške je česta mjera kvaliteta modela mašinskog učenja. Osim ovoga ima čestu primjenu u statistici i ekonomiji.

5.3 Postavke treniranja

Rekurentna neuralna mreža na kojoj se treniraju vremenski podaci sa snimaka je duboka rekurentna mreža. Rekurentna mreža je duboka u dva smjera. Produbljena je naslaganjem više rekurentnih slojeva jedan za drugim. Izlaz u svakom terenutku na jednom sloju, predstavlja ulaz u sledeći sloj rekurentne mreže. Mreža je produbljena i na izlazu primjenom potpuno konektovane mreže na izlazu.

Treniranje mreže je postavljeno na sledeći način. Broj epoha treniranja je postavljen na 100. Za teniranje se koristi mini-serijski gradijentni spust. Veličina podskupa ili *batch-a* je postavljena na 32. Za optimizacionu funkciju, koristi se *Adam*.

Kao aktivaciona funkcija kod rekurentnih slojeva mreže koristi se *relu* funkcija. Relu je linearna aktivaciona funkcija zbog čega se često koristi kod dubokih neuralnih mreža, olakšavajući učenje smanjivanjem nelinearnosti u mreži.

Funkcija cijene koja se koristi u treniranju je srednja kvadratna greška (*mean square error mse*). Kao prevencija pretreniranja postavljena je i regularizacija. Regularizacija koja je upotrijebljena je *dropout* za rekurentne mreže. Ovi parametri su određeni eksperimentalnim putem.

5.4 Odabir mreže

Arhitektura rekurentne neuralne mreže je određena eksperimentalnim putem. Za rekurentne slojeve mreže uzete su u obzir prosta rekurntna mreža, LSTM i GRU rekurentna mreža. Prosta rekurentna mreža daje najslabije rezultate procjene brzine. LSTM i GRU mreže treniranjem dolaze do rezultata procjene brzine sličnog kvaliteta. U ovom radu fokus je bio na LSTM mreži zbog stabilnijeg učenja i manje neočekivanih rezultata.

Kako bi se model uspješno prilagodio podacima i dao preciznu procjenu bilo je potrebno povećati njegovu reprezentativnu moć. Ovo se postiže produbljivanjem modela. Produbljivanje je postignuto na dva načina, naslaganjem više rekurentnih slojeva jedan nakon drugog i dodavanjem potpuno povezanih slojeva na izlazu iz mreže. Povećavanjem broja rekurentnih slojeva povećava se i preciznost modela. Osim preciznosti povećava se i njegova veličina i složenost što utiče na njegovu brzinu. Dodavanjem više od tri rekurentna sloja pokazalo se nepotrebno, zbog neznatnog povećanja preciznosti procjene u odnosu na veliku složenost modela. Veća složenost označava takođe i veću podložnost modela na pretreniranje, pa je odabrana arhitektura sa tri rekurentna sloja.

Izlaz rekurentne mreže je njegovo stanje, koje je dimenzija njegovog unutrašnjeg sloja. Ovo stanje je potrebno protumačiti kao procjenu brzine vozila. Za ovo služe slojevi potpuno povezane mreže koje se nalaze na izlazu rekurentne mreže. U ovom istraživanju su se dva potpuno povezana sloja i jedan izlazni sloj sa jednim neuronom pokazali kao dovoljni.

Osim broja slojeva bitna je i veličina skrivenog stanja slojeva rekurentne mreže. Broj neurona u skrivenom stanju mreže utiče na reprezentativnu moć mreže kao i na njihovu složenost. Kod odabira broja neurona u skrivenom stanju obično se uzima broj veći od dimenzija ulaza u mrežu a manji od dvostruke dimenzije ulaza. Kako su ispitivani različiti oblici metoda procjene, za ovo su korištene različite dimenzije.

5.5 Učenje sa sekvencama različitih dužina

Prednost upotrebe rekurentnih neuralnih mreža je mogućnost rada sa sekvencama različitih dužina. Kao ulaz mreže dužina sekvence nije određena, kao kod standardnih neuralnih mreža. Kako podaci iz svakog vremenskog koraka nezavisno dolaze na ulaz mreže, broj koraka se može nesmetano mijenjati. Međutim, postoji ograničenje koje se pojavljuje prilikom treniranja. Prilikom treniranja, kod upotrebe *mini-batch* metode, propagacija unazad se izvršava tek nakon više propagacija unaprijed. Kako kod rekurentnih neuralnih mreža, vremenska dužina sekvence se translira u njenu dubinu, potrebno je da su unutar *batch*-a sekvence istih dužina.

Zbog ovoga ograničenja je potrebno na neki način izvršiti izjednačenje dužina sekvenci koje dolaze na ulaz mreže tokom treniranja mreže. Ovo ograničenje ne postoji kod testiranja i upotrebe modela. Propagacija unaprijed se i dalje može izvršiti sa različitim dužinama sekvence.

Drugi cilj upotrebe rekurentnih mreža je prilagođavanje modela različitim djelovima sekvence. Mreža bi trebala dati precizne rezultate i u situacijama kada nije dostupan video čitavog prolaza vozila nego samo njegov dio. Takođe je bitna mogućnost praćenja brzine tokom prolaza. Korišćenjem ovih mreža moguće je dobiti procjenu brzine u svakom vremenskom trenutku, omogućavajući praćenje brzine u realnom vremenu.

Ove ciljeve je moguće ispuniti na više načina. Prvi pristup predstavlja izjednačavanje dužina sekvenci sa svakog snimka dodavajući nule na kraju i početku sekvence, tako da se sve sekvence izjednače sa najdužom. Ovakve sekvence se daju mreži prilikom treniranja. Rekurentna mreža se podešava na način da se nakon svakih n vremenskih koraka

daje predikcija brzine. Testiranje se dalje obavlja na sekvencama različitih dužina i sa različitih djelova snimka da bi se provjerila njegova generalizacija.

Drugi pristup predstavlja dijeljenje sekvenci na segmente jednake dužine. Segmenti se mogu preklapati u nekom dijelu. Predikcija se vrši na kraju svakog segmenta, šta daje isti efekat kao prethodni pristup, da se predikcija dobija na svkih n vremenskih koraka. Eksperimentisano je sa različitim dužinama sekvence i različitim dužinama preklapanja sekvenci. Kod ovog pristupa mreža se izolovano trenira na svakom segmentu bez znanja sta je bilo prije i posle njega. Ovo je slučaj i kod testiranja gdje se duže sekvence prvo podijele na segmente predefinisane dužine. Postoji mogućnost da se precizniji rezultati dobiju kombinacijom predikcija predhodnih segmenata sa novim segmentima, pa je i ta mogućnost razmatrana.

5.6 Odabir karakteristika

Rekurentna neuralna mreža koja vrši predikciju brzine vozila, to radi nad karakteristikama koje dobija iz detektora objekata na slici. Ove karakteristike su podaci o položaju graničnog okvira, predstavljeni koordinatama njene centralne tačke, kao i podaci o veličini graničnog okvira, predstavljeni visinom i širinom graničnog okvira. Karakteristika koja se iz ovih podataka može izračunati je površina graničnog okvira koja se pokazuje kao važna karakteristika.

Na slici 5.1 je prikazana vremenska promjena važnih karakteristika u ovom istraživanju. Na prvom grafiku se nalazi promjena površine graničnog okvira sa vremenom. Ova karakteristika polako raste dok vozilo prilazi poziciji kamere. Kada dostigne maksimum naglo opada kako počne da izlazi iz kadra. Na drugom grafiku je prikazana promjena vrijednosti x koordinate tokom vremena. U slučaju na slici ova vrijednost raste zbog pozicije kamere u odnosu na vozilo takve da vozilo prolazi sa desne strane kamere. Na trećoj slici se nalazi promjena vrijednosti y koordinate tokom vremena. Ova vrijednost opada zbog ugla kamere na put, takvog da je na početku vozilo blizu vrha snimka, a kako prilazi kameri se spušta.

Motivacija uvođenja karakteristike površine graničnog okvira je uklanjanje uticaja koji predstavlja ugao kamere i njena pozicija u odnosu na put u ulaznim podacima. Ukoliko se promijeni ugao kamere može doći do drugačije promjene karakteristika koje su zasnovane na koordinatama vozila (x i y koordinate centra). Sa druge strane, promjena površine vozila se ne mijenja promjenom ovih parametara.

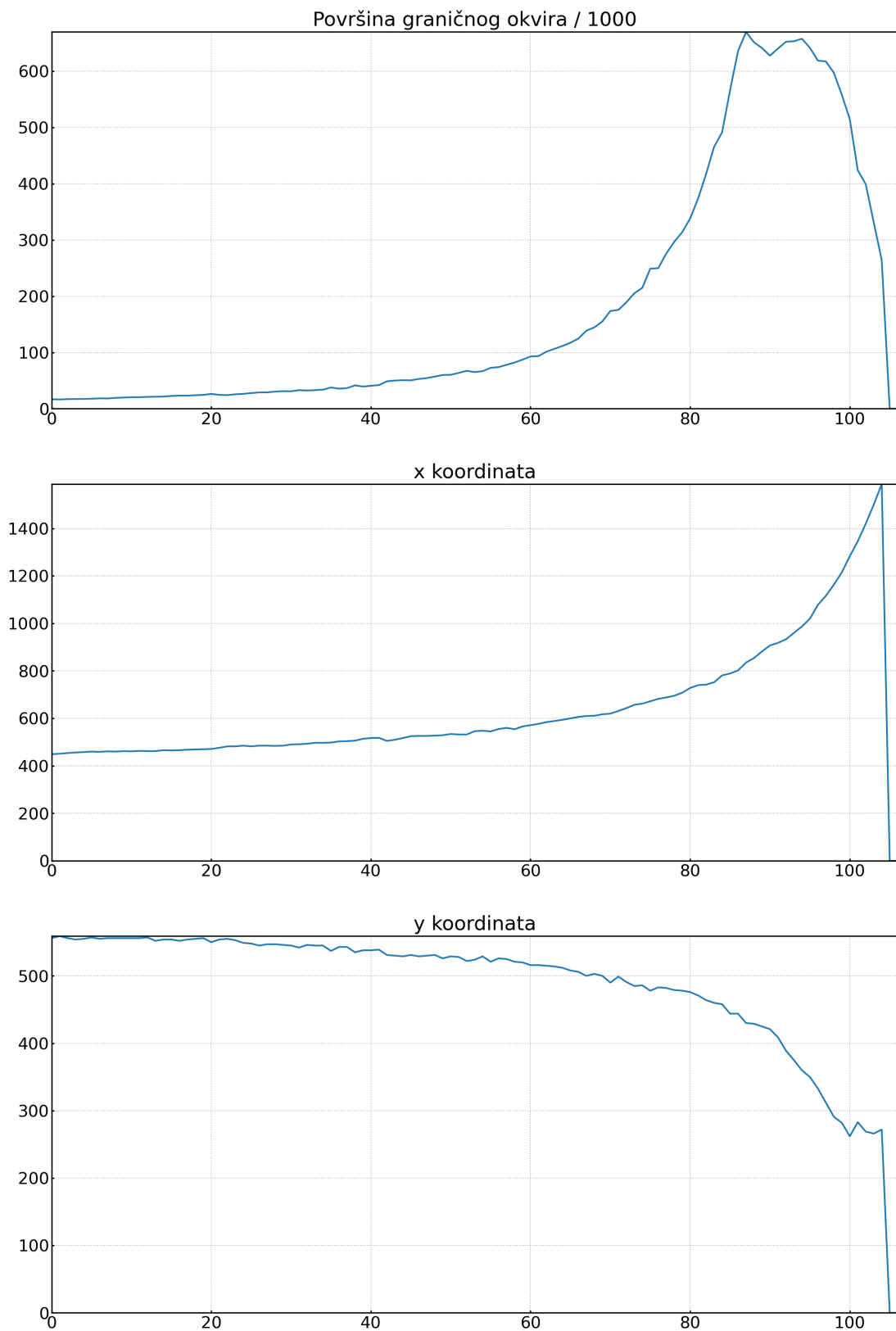
U ovom dijelu je predstavljeno poređenje modela koji kao ulazne podatke koriste površinu i poziciju graničnog okvira vozila, i metode koja koristi samo površinu graničnog okvira.

Rezultati sprovedenog eksperimenta na ova dva pristupa su prikazani u tabeli 5.1. U srednjoj koloni su rezultati kada su mreži dati podaci o površini i poziciji graničnog okvira, dok su u poslednjoj koloni rezultati kada je proslijeđena samo površina.

Tabela 5.1: RMSE [km/h] procijenjene brzine

Vozilo	RMSE _{površina&pozicija}	RMSE _{površina}
Citroen C4 Picasso	5.04	2.38
Kia Sportage	3.25	4.90
Mazda 3 Skyactive	5.10	5.10
Mercedes AMG 550	4.58	4.51
Mercedes GLA	4.06	3.65
Nissan Qashqai	2.93	3.14
Opel Insignia	9.40	3.77
Peugeot 208	3.99	3.28
Peugeot 307	3.27	3.15
Peugeot 3008	5.35	4.99
Renault Captur	4.60	4.90
Renault Scenic	3.84	3.92
VW Passat B7	3.27	3.77
Srednja vrijednost	4.70	4.08

U tabeli se može vidjeti da je ukupna greška eksperimenata manja kod pristupa sa samo površinom kao ulaznom karakteristikom. Analizom ovih rezultata za svako vozilo pojedinačno može se vidjeti da su rezultati uglavnom približni. Razlika koja se dešava u ukupnoj grešci je rezultat greške od 9.4 kod procjene brzine Opel Insignije. Analizom podataka o poziciji za to vozilo, može se vidjeti da se dešava greška prilikom detekcije tog vozila. Ova greška najviše utiče na poziciju graničnog okvira, a veoma malo na njegovu površinu.



Slika 5.1: Vremenska promjena ulaznih karakteristika rekurentne mreže

Ovim možemo zaključiti da upotrebom karakteristike površine graničnog okvira, se dobija robustnija metoda procjene brzine vozila. Greške prilikom detekcije ovim imaju manjeg uticaja na konačnu predikciju brzine. Osim ovoga uklanja se i uticaj ugla i pozicije kamere u odnosu na put.

5.7 Odabir sekvenci

U ovom segmentu je prikazana analiza pristupa sa dijeljenjem svake sekvence na podsekvence jednakih dužina. Odabir dužine podsekvenci i koraka na kojem se uzimaju ima važan uticaj na rezultate metode. Dužina sekvence određuje na koliko vremenskih koraka se vrši procjena brzine vozila. Na malom broju podataka preciznost metode je veoma mala. Što je dužina podsegmenata veća preciznost metode je veća. Međutim povećanjem dužine se smanjuje mogućnost praćenja brzine u svakom trenutku kao i mogućnost generalizacije i primjene na metode na kraćim sekvencama.

U tabeli 5.2 prikazano je poređenje rezultata eksperimenta sa ulazom koji je čitava sekvenca i rezultata eksperimenta sa sekvencama podijeljenim na podsekvence dužine 30 vremenskih koraka.

Može se vidjeti da se preciznost procjene povećava kako je dužina sekvence veća. Ukupna greška ovog eksperimenta dostiže 3.67, što je primjetno poboljšanje u odnosu na eksperiment sa sekvencama od 30 vremenskih koraka. Sa druge strane, kao što je prikazano u tabeli ovo poboljšanje ne važi za svako vozilo u setu podataka. Osim ovoga, duže sekvence otežavaju procjenu u realnom vremenu, pošto je za jednu procjenu potreban duži vremenski period. Takođe, ostvaruju se znatno lošiji rezultati ukoliko ovako treniranoj mreži se dostave sekvence različite dužine od one na kojoj je trenirana. Ovo označava slabiju generalizaciju modela.

Mreža trenirana na podsekvencama dužine 30 vremenskih koraka, takođe pokazuje pad preciznosti kada se primjenjuje na sekvencama dosta različite dužine od one na kojoj je trenirana. Prednost ovog pristupa je u tome što su ove sekvence dovoljno kratke za praćenje brzine u vremenu. Osim toga ima bolju primjenljivost na sekvencama različitih dužina jer se duža sekvenca uvijek može podijeliti na sekvence.

Tabela 5.2: RMSE [km/h] procijenjene brzine

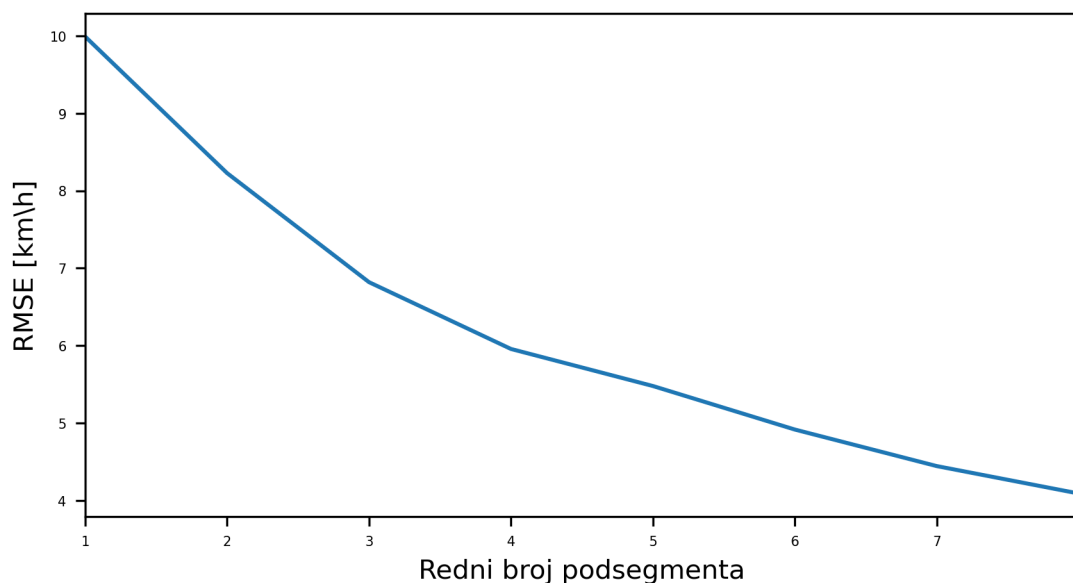
Vozilo	RMSE _{čitava sekvenca}	RMSE _{podsekvence}
Citroen C4 Picasso	4.01	2.38
Kia Sportage	4.07	4.90
Mazda 3 Skyactive	2.72	5.10
Mercedes AMG 550	3.59	4.51
Mercedes GLA	2.04	3.65
Nissan Qashqai	2.26	3.14
Opel Insignia	3.51	3.77
Peugeot 208	4.73	3.28
Peugeot 307	2.23	3.15
Peugeot 3008	5.29	4.99
Renault Captur	3.10	4.90
Renault Scenic	2.25	3.92
VW Passat B7	5.51	3.77
Srednja vrijednost	3.67	4.08

Kako se procjene dobijaju na više djelova svakog smimka, može se analizirati kvalitet procjene na svakom segmentu. Na slici 5.2 prikazana je greška procjene brzine razvijene metode na pozicijama svakog od podsegmenata.

Može se vidjeti da je na početku snimka preciznost procjene brzine najmanja, dok se kako se vozilo primiče kameri preciznost povećava. Ovakvo ponašanje se može objasniti time što je granični okvir vozila, koje je detektovano daleko od kamere, male površine. Ovo znači da su promjene njegove površine zbog njegovog kretanja takođe veoma male pa je teško precizno procijeniti brzinu.

Detektor objekata takođe ima uticaja na ovakvu procjenu. Kako YOLO detektor ima problema sa detekcijom malih objekata na slici, detekcija može biti manje precizna. Ovo se dalje negativno odražava na detekciju brzine.

Nepreciznost detekcije brzine udaljenih vozila, kao i nepreciznost njihove same detekcije proizilazi iz diskretne prirode video snimka. Preciznost digitalne reprezentacije opada srazmjerno kvadratu udaljenosti od kamere.



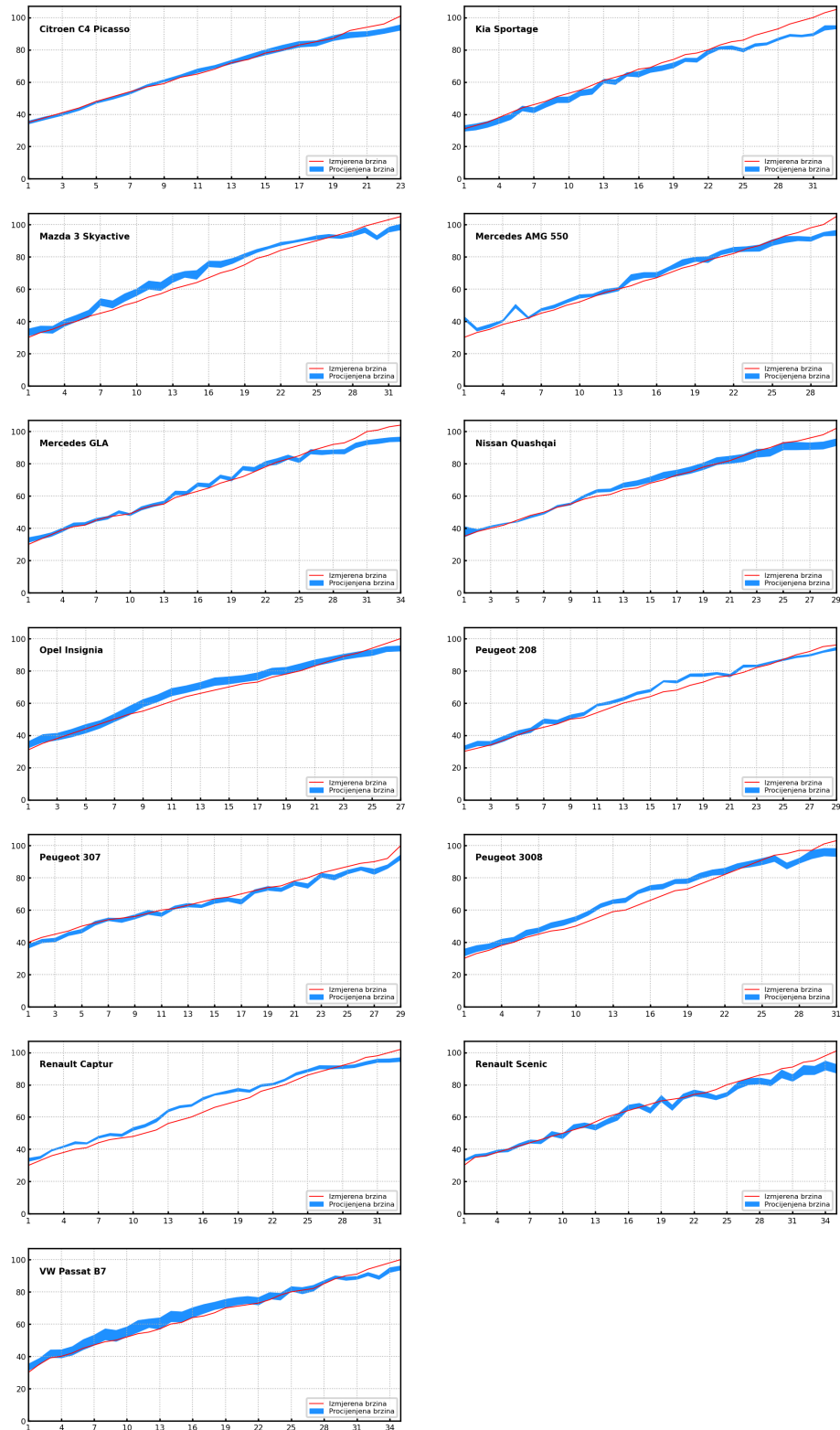
Slika 5.2: Promjena grške u zavisnosti od pozicije podsekvence u ukupnoj sekvenci

5.8 Rezultati

Na slici 5.3 su prikazani rezultati eksperimenata u grafičkom obliku. Crvena linija predstavlja stvarne vrijednosti brzine vozila na snimcima dok plava oblast predstavlja interval 95% pouzdanosti ¹. Svaka od slika predstavlja rezultate eksperimenata za jedno od vozila u setu za testiranje.

Na slikama se može uočiti da je preciznost metode procjene prilično dobra i da procijenjena brzina prilično blisko prati stvarnu brzinu vozila na snimcima. Takođe, na graficima je uočljivo da se najmanja preciznost dobija pri velikim brzinama. U ovim okolnostima predložena metoda daje veću grešku sa težnjom da podcijeni brzinu.

¹Interval pouzdanosti (eng. confidence interval) koristi se u statistici kako bi se procijenio raspon vrijednosti u kojem se sa određenom vjerovatnoćom nalazi stvarna vrijednost posmatranog parametra.



Slika 5.3: Interval 95% pouzdanosti pikazan na svakog vozila iz skupa podataka

Zaključak

Detekcija brzine vozila pomoću vizualnih senzora, kakve su kamere, predstavlja rastuću oblast istraživanja. Ovim istraživanjem je predstavljen inovativni dizajn detektora brzine vozila pomoću video snimka koji primjenjuje algoritme kompjuterske vizije. Ova metoda procjene brzine koristi specijalno dizajniranu rekurentnu neuralnu mrežu za obradu sekvencijalnih podataka dobijenih iz video snimka korišćenjem detektora objekata na slici. Predložena rekurentna mreža pokazuje da je moguća precizna detekcija brzine ostvarujući grešku od 4.08 km/h na VS13 setu podataka.

Bitno ograničenje koje je bilo prisutno u ovom istraživanju je veličina skupa podataka za treniranje. Korišteni skup, VS13 koji sastoji od 400 snimaka od deset sekundi, je relativno mali u kontekstu dubokog učenja. Ovaj problem može dovesti do manje pouzdanih rezultata. Međutim, ovo ograničenje može se djelimično ublažiti primjenom tehnika kros-validacije i pažljivim dizajnom rekurentne neuralne mreže.

Upotreba rekurentne neuralne mreže prilikom detekcije brzine omogućava uklanjanje spoljašnjih kalibracija kamere, koje su potrebne u drugim metodama procjene. Ovo je ostvareno uvođenjem ograničenja u pravcu kretanja vozila, koje na snimku mora prilaziti ili se udaljavati od kamere. Ova metoda se najbolje primjenjuje u situacijama kada je moguće jasno detektovati vozilo i pratiti njegove karakteristike tokom vremena.

U cilju prevazilaženja prikazanih ograničenja, buduća istraživanja će biti usmjerena na prikupljanje većeg i raznovrsnijeg skupa podataka za treniranje. On bi omogućio bolju generalizaciju i pouzdanost modela. Skup podataka koji bi uključivao različite uslove vidljivosti uticao bi na robusnost metode i njenu primjenljivost u različitim uslovima vidljivosti, kakvi su sumrak, kiša ili druge vremenske prilike. Ispitivanje primjene drugih detektora vozila kao što je DETR, koji trenutno pokazuje najbolje rezultate u detekciji objekata na slici, bi se moglo odraziti na kvalitetu procjene brzine.

Uprkos ovim ograničenjima, predložena metoda pruža prilično precizne rezultate procjene brzine. Praktična primjena ove metode može doprinijeti unapređenju sistema kontrole saobraćaja, poboljšanju sigurnosti na putevima i razvoju inteligentnih transportnih sistema.

Literatura

- [1] G. J. Dimitrakopoulos, L. Uden, and I. Varlamis, *The future of intelligent transport systems*. Elsevier, 2020.
- [2] K. Sjoberg, P. Andres, T. Buburuzan, and A. Brakemeier, “Cooperative intelligent transport systems in europe: Current deployment status and outlook,” *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 89–97, 2017.
- [3] D. F. Llorca, A. H. Martínez, and I. G. Daza, “Vision-based vehicle speed estimation: A survey,” *IET Intelligent Transport Systems*, vol. 15, no. 8, pp. 987–1005, may 2021. [Online]. Available: <https://doi.org/10.1049%2Fitr2.12079>
- [4] J. D. Trivedi, S. D. Mandalapu, and D. H. Dave, “Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation,” *Journal of Industrial Information Integration*, vol. 27, p. 100280, 2022.
- [5] G. Cheng, Y. Guo, X. Cheng, D. Wang, and J. Zhao, “Real-time detection of vehicle speed based on video image,” in *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. IEEE, 2020, pp. 313–317.
- [6] J. Sochor, R. Juránek, and A. Herout, “Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement,” *Computer Vision and Image Understanding*, vol. 161, pp. 87–98, 2017.
- [7] C. Liu, D. Q. Huynh, Y. Sun, M. Reynolds, and S. Atkinson, “A vision-based pipeline for vehicle counting, speed estimation, and classification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7547–7560, 2020.
- [8] D. Bell, W. Xiao, and P. James, “Accurate vehicle speed estimation from monocular camera footage,” in *XXIV ISPRS Congress*. Newcastle University, 2020.

- [9] H. Dong, M. Wen, and Z. Yang, “Vehicle speed estimation based on 3d convnets and non-local blocks,” *Future Internet*, vol. 11, no. 6, p. 123, 2019.
- [10] D. C. Luvizon, B. T. Nassu, and R. Minetto, “Vehicle speed estimation by license plate detection and tracking,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6563–6567.
- [11] L. Yang, M. Li, X. Song, Z. Xiong, C. Hou, and B. Qu, “Vehicle speed measurement based on binocular stereovision system,” *IEEE Access*, vol. 7, pp. 106 628–106 641, 2019.
- [12] M. Famouri, Z. Azimifar, and A. Wong, “A novel motion plane-based approach to vehicle speed estimation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1237–1246, 2018.
- [13] E. Vakili, M. Shoaran, and M. R. Sarmadi, “Single-camera vehicle speed measurement using the geometry of the imaging system,” *Multimedia Tools and Applications*, vol. 79, pp. 19 307–19 327, 2020.
- [14] Q. Li, R. Li, K. Ji, and W. Dai, “Kalman filter and its application,” in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 2015, pp. 74–77.
- [15] J. Sochor, R. Juránek, J. Špaňhel, L. Maršík, A. Široký, A. Herout, and P. Zemčík, “Comprehensive data set for automatic single camera visual speed measurement,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1633–1643, 2018.
- [16] M. Dubská, A. Herout, and J. Sochor, “Automatic camera calibration for traffic understanding,” in *BMVC*, vol. 4, no. 6, 2014, p. 8.
- [17] L. Medsker and L. C. Jain, *Recurrent neural networks: design and applications*. CRC press, 1999.
- [18] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

- [19] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.
- [20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [21] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- [22] A. Gruslys, R. Munos, I. Danihelka, M. Lanctot, and A. Graves, “Memory-efficient backpropagation through time,” *Advances in neural information processing systems*, vol. 29, 2016.
- [23] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, “Recent advances in recurrent neural networks,” *arXiv preprint arXiv:1801.01078*, 2017.
- [24] T. Moon, H. Choi, H. Lee, and I. Song, “Rnndrop: A novel dropout for rnns in asr,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 65–70.
- [25] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to construct deep recurrent neural networks. arxiv 2013,” *arXiv preprint arXiv:1312.6026*, 2019.
- [26] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [27] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, pp. 303–338, 2010.
- [29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, pp. 211–252, 2015.

- [30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
- [31] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov *et al.*, “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale,” *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.
- [32] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, pp. 137–154, 2004.
- [33] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.
- [34] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *2008 IEEE conference on computer vision and pattern recognition*. Ieee, 2008, pp. 1–8.
- [35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [36] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International journal of computer vision*, vol. 104, pp. 154–171, 2013.
- [37] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [38] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [39] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proceedings of the IEEE*, 2023.

- [40] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
- [43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [45] J. Solawetz, “Vehicles-openimages dataset,” <https://universe.roboflow.com/roboflow-gw7yv/vehicles-openimages>, June 2022.
- [46] S. Djukanović, N. Bulatović, and I. Čavor, “A dataset for audio-video based vehicle speed estimation,” in *2022 30th Telecommunications Forum (TELFOR)*. IEEE, 2022, pp. 1–4.