

# Detekcija vremenskih uvjeta u okolini automobila

---

Triva, Jakob

Master's thesis / Diplomski rad

2019

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:519652>

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

Download date / Datum preuzimanja: **2024-08-26**

Repository / Repozitorij:

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



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**  
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I**  
**INFORMACIJSKIH TEHNOLOGIJA**  
**Sveučilišni diplomski studij računarstva**

**DETEKCIJA VREMENSKIH UVJETA U OKOLINI**  
**AUTOMOBILA**

**Diplomski rad**

**Jakob Triva**

**Osijek, 2019.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 09.09.2019.

Odboru za završne i diplomske ispite

**Imenovanje Povjerenstva za obranu diplomskog rada**

<b>Ime i prezime studenta:</b>	Jakob Triva
<b>Studij, smjer:</b>	Diplomski sveučilišni studij Računarstvo
<b>Mat. br. studenta, godina upisa:</b>	D-946R, 24.09.2018.
<b>OIB studenta:</b>	61369380319
<b>Mentor:</b>	Doc.dr.sc. Ratko Grbić
<b>Sumentor:</b>	
<b>Sumentor iz tvrtke:</b>	Matija Pul
<b>Predsjednik Povjerenstva:</b>	Izv. prof. dr. sc. Mario Vranješ
<b>Član Povjerenstva:</b>	Izv. prof. dr. sc. Marijan Herceg
<b>Naslov diplomskog rada:</b>	Detekcija vremenskih uvjeta u okolini automobila
<b>Znanstvena grana rada:</b>	<b>Umjetna inteligencija (zn. polje računarstvo)</b>
<b>Zadatak diplomskog rada:</b>	Odluke koje donosi autonomno vozilo uzimaju u obzir stanje vozila i vozača, ali i stanje okoline oko samog vozila. Da bi automno vozilo moglo samostalno voziti, mora biti svjesno svih opasnosti na cesti i vozilu kao &scaron;to su vremenske nepogode. U ovom diplomskom radu potrebno je izraditi algoritam za prepoznavanje opasnosti od nepogodnih vremenskih uvjeta, kao &scaron;to su snijeg, ki&scaron;a ili magla, na temelju slika dobivenih s kamere na prednjem kraju vozila. Nakon razvoja algoritma u željenom programskom jeziku, potrebno je algoritam prebaciti u programski jezik C, izvr&scaron;iti njegovu optimizaciju i ispitati mogućnost implementacije na ADAS (engl. Advanced Driver Assistance System) ploču kako bi se testirala njegova učinkovitost na realnoj platformi pri
<b>Prijedlog ocjene pismenog dijela ispita (diplomskog rada):</b>	Izvrstan (5)
<b>Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:</b>	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
<b>Datum prijedloga ocjene mentora:</b>	09.09.2019.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

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

Osijek, 16.09.2019.

**Ime i prezime studenta:**

Jakob Triva

**Studij:**

Diplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

D-946R, 24.09.2018.

**Ephorus podudaranje [%]:**

4

Ovom izjavom izjavljujem da je rad pod nazivom: **Detekcija vremenskih uvjeta u okolini automobila**

izrađen pod vodstvom mentora Doc.dr.sc. Ratko Grbić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

## IZJAVA

Ja, Jakob Triva, OIB: 61369380319, student/ica na studiju: Diplomski sveučilišni studij Računarstvo, dajem suglasnost Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek da pohrani i javno objavi moj **diplomski rad**:

**Detekcija vremenskih uvjeta u okolini automobila**

u javno dostupnom fakultetskom, sveučilišnom i nacionalnom repozitoriju.

Osijek, 18.09.2019.

---

potpis

# Sadržaj

1. UVOD .....	1
2. DETEKCIJA VREMENSKIH UVJETA.....	3
3. NEURONSKE MREŽE.....	8
3.1. Uvod .....	8
3.2. Duboke neuronske mreže .....	10
3.2.1. Konvolucijske neuronske mreže .....	10
3.2.2. Potpuno povezani sloj .....	12
3.2.3. Konvolucijski sloj .....	13
3.2.4. Sloj sažimanja .....	14
3.2.5. <i>SoftMax</i> sloj.....	15
3.2.6. Aktivacijska funkcija.....	15
3.3. Učenje i evaluacija neuronskih mreža .....	18
3.3.1. Funkcija gubitka .....	18
3.3.2. Slučajno izostavljanje neurona.....	19
4. DETEKCIJA VREMENSKIH UVJETA U OKOLINI AUTOMOBILA POMOĆU DUBOKIH NEURONSKIH MREŽA .....	21
4.1. Prikupljanje i predobrada podataka .....	21
4.2. Prijedlog rješenja detekcije vremenskih uvjeta zasnovano na konvolucijskoj neuronskoj mreži .....	25
4.2.1. Odabir strukture i podešavanje parametara konvolucijske neuronske mreže .....	26
4.2.2. Kružni međuspremnik .....	30
5. VERIFIKACIJA RADA RJEŠENJA .....	32
5.1. Evaluacija izgrađenih konvolucijskih neuronskih mreža .....	32
5.2. Evaluacija predloženog rješenja za klasifikaciju vremenskih uvjeta .....	34
6. ZAKLJUČAK .....	39
LITERATURA .....	40
SAŽETAK.....	43
ABSTRACT .....	44
ŽIVOTOPIS .....	45

# 1. UVOD

Autonomna vožnja postaje sve popularnija među općom populacijom zbog svih komfornih i sigurnosnih boljitaka nad dosadašnjim vozilima upravljanim čovjekom. Znanstvenici su procijenili kako je 75% nesreća na cestama uzrokovano izričito ljudskom greškom, dok je u 95% nesreća uključen određen nivo ljudske greške [1]. Uvođenjem koncepta autonomnosti u prijevozna vozila jedan od ciljeva je eliminirati ljudsku pogrešku i time osigurati veći nivo sigurnosti putnika. Prvo i osnovno pitanje kojim se može ući u problematiku autonomne vožnje jest što je to po čemu možemo neko vozilo okarakterizirati kao autonomno.

Prema NHTSA (Nacionalna uprava za sigurnost prometa na cestama, SAD) razlikujemo pet razina autonomnosti [2]. Nulta razina autonomnosti predstavlja razinu autonomnosti gdje je vozač u potpunom i isključivom nadzoru primarnih kontrola vozila u svakom trenutku te odgovoran za nadzor kolnika i za siguran rad svih kontrola vozila. Prva razina omogućava vozaču predaju ograničenih ovlasti nad primarnom kontrolom (npr. dinamička podrška pri kočenju u hitnim slučajevima). Vozila na drugoj razini automatizacije mogu koristiti zajednička ovlaštenja u određenim ograničenim situacijama vožnje, ali je vozač i dalje odgovaran za praćenje kolnika i siguran rad. Vozila na trećoj razini automatizacije omogućuju vozaču da prepusti potpunu kontrolu svih sigurnosno kritičnih funkcija pod određenim uvjetima, gdje se od vozača očekuje da bude dostupan za povremenu kontrolu. Na četvrtoj razini autonomnosti vozila su dizajnirana da izvrše sve sigurnosno kritične upravljačke funkcije te da nadziru uvjete na cesti tijekom cijelog putovanja.

Iako koncepti autonomnih vozila postoje već dulje vrijeme, napretkom poluvodičkih i komunikacijskih tehnologija, pravi pomaci prema potpuno autonomnim vozilima vidljivi su u posljednjih nekoliko godina. Jedan od spomenutih čimbenika koje mora zadovoljavati autonomno vozilo jest da nadzire uvjete na cesti tijekom cijelog putovanja. Pod uvjete na cesti možemo ubrojiti sve ono što se nalazi/događa u okolini vozila što može posredno ili neposredno utjecati na vožnju automobila, u što je važno ubrojiti i vremenske uvjete.

Uz sve napretke autonomne vožnje, sigurna vožnja za vrijeme nepovoljnih vremenskih uvjeta još uvijek predstavlja veliku prepreku prema potpunoj autonomnosti vozila. Opći problemi koje stvaraju nepovoljni vremenski uvjeti su smanjena vidljivost na cesti, skliska cesta, nakupine vode i vodeni klinovi. Neki od izazova koje autonomno vozilo u takvim uvjetima mora

riješiti su prilagodba vožnje po skliskoj cesti (prilagodba ubrzavanja i kočenja), detekcija nakupina vode i odluka o najboljem pristupu prelaska ili obilaska nakupine, tehnika održavanja ili ponovnog stjecanja kontrole nad vozilom nakon nailaska na vodeni klin i sl. Problemi se mogu manifestirati i neispravnim radom senzora. Problem s kojim se rad lidara (engl. *Light Detection and Ranging*) susreće za vrijeme nepovoljnih vremenskih uvjeta je odbijanje odaslanog signala od kapljica kiše ili snježnih pahuljica, što može rezultirati pogrešnom detekcijom objekta u blizini vozila. Naime, lidar je uređaj koji se koristi u sustavima percepcije i koji radi na principu odašiljanja svjetlosnog signala i mjerenja trajanja njegovog puta do objekta i natrag. Osim lidara, nepovoljni vremenski uvjeti otežavaju rad kamerama, radarima i GPS-u. Kamere se susreću s otežanom detekcijom rubova što može uzrokovati nemogućnost prepoznavanja različitih objekata (poput prometnih znakova) čime se može drastično smanjiti sigurnost vožnje autonomnog vozila [3]. Ulaskom u problematiku autonomne vožnje zaključeno je da je dobro prepoznavanje vremenskih uvjeta u radnoj okolini sustava važna stavka u donošenju adekvatnih odluka.

Neka od rješenja detekcija vremenskih uvjeta u okolini automobila koriste algoritme koji se baziraju na podacima sa senzora poput kamere i lidara. Zbog visoke cijene lidara poželjno je probleme s kojima se susreće autonomno vozilo riješiti uz pomoć informacija koje dolaze s jeftinijih senzora, poput kamere. Postoje različita rješenja problema detekcije vremenskih uvjeta na temelju slika koji koriste različite algoritme strojnog učenja (engl. *Machine Learnig* - ML) poput AdaBoost algoritma, konvolucijskih neuronskih mreža (engl. *Convolutional Neural Network* – CNN) itd.

U ovom radu predstavljena je izgradnja softverske podrške koja na temelju niza okvira dobivenih s kamere montirane na prednjoj strani vozila daje informaciju o trenutnim vremenskim uvjetima na cesti. Osnova softverske podrške je klasifikator koji je zasnovan na konvolucijskoj neuronskoj mreži i koji svakom ulaznom okviru dodjeljuje oznaku vremenskog uvjeta: čisto, snježno, kišovito i maglovito vrijeme.

Rad je strukturiran na sljedeći način. U drugom poglavlju dan je pregled nekih rješenja problema detekcije vremenskih uvjeta te uvod u samu problematiku. Trećim poglavljem predstavljena je teorijska osnova rada različitih matematičkih struktura, metoda i funkcija korištenih u izradi rješenja. U četvrtom poglavlju predstavljeno je samo rješenje te su dani rezultati testiranja dobivenog rješenja. Na kraju rada dan je zaključak koji uokviruje sve navedeno.



## 2. DETEKCIJA VREMENSKIH UVJETA

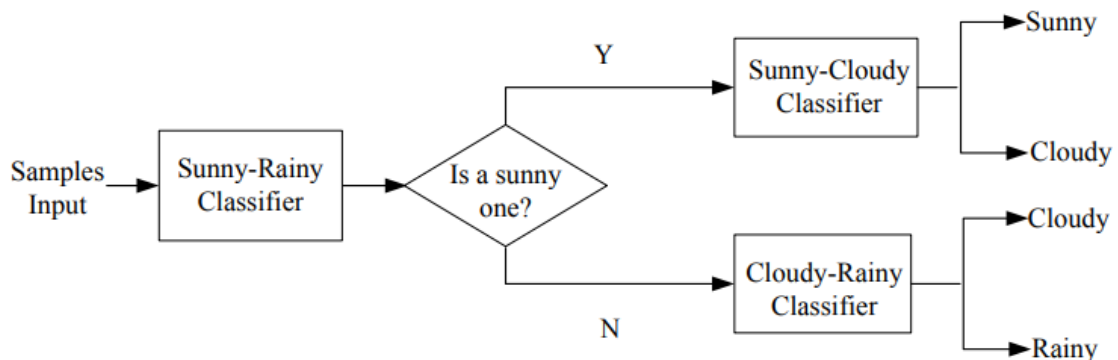
Prvi radovi koji se bave rješavanjem problema detekcije vremenskih uvjeta u okolini automobila pojavljuju se u zadnja dva desetljeća. U ovom poglavlju predstavljeni su najznačajniji radovi koji se bave sličnom ili istom problematikom.

U [4] predstavljeno je rješenje problema prepoznavanja vremenskih uvjeta na temelju značajki izračunatih iz slika snimljenih kamerom iz vozila. Pristup rješenju bio je definiranje spomenutih značajki:

- histogram amplitudnog gradijenta (engl. *Histogram of Gradient Amplitude* - HGA),
- HSV histogram boja, i
- informacije o cesti predstavljene srednjom vrijednosti sive boje u odabranim ROI-ma (engl. *Region of interest*) slike,

te implementacija algoritma baziranog na Real AdaBoost algoritmu koristeći kategoričke strukture za rješavanje problema klasifikacije.

Kategorička struktura rješenja prikazana je slikom 2.1. Takva je struktura korištena pod pretpostavkom da pogrešno klasificirane slike klasifikatorom *Sunny-Rainy* imaju veću vjerojatnost pripadanja oblačnom vremenu jer distribucije značajki sunčanog i kišovitog vremena ne dijele isti prostor.



**Slika 2.1.** Procedura algoritma baziranog na kategoričkoj strukturi [4].

Tablicom 2.1. prikazani su rezultati, tj. točnost različitih testiranih rješenja problema. Prezentirane su točnosti predikcije modela realiziranih pomoću kNN (engl. *k Nearest Neighbor*), AdaBoost.MH koji se može direktno koristiti za višeklasnu klasifikaciju, *one-vs-all* AdaBoost

algoritma koji koristi K klasifikatora za K klasa i AdaBoost algoritma s kategoričkom strukturom. Iako *one-vs-all* AdaBoost algoritam pokazuje bolje rezultate, kategorički AdaBoost algoritam ima značajno manje vrijeme izvršavanja.

**Tablica 2.1.** Rezultati klasifikacije algoritama [4].

	Točnost	Točnost predikcije sunčanog vremena	Točnost predikcije oblačnog vremena	Točnost predikcije kišovitoг vremena
kNN	89.10%	93.51%	83.91%	89.61%
AdaBoost.MH	89.29%	94.22%	83.82%	89.82%
AdaBoost ( <i>one-vs-all</i> )	92.15%	95.76%	88.81%	91.88%
AdaBoost (kategorička struktura)	91.92%	96.00%	89.35%	90.41%

U radu [5] predstavljena je metoda automatskog prepoznavanja vremenskih uvjeta u okolini vozila bazirana na podacima dobivenih s kamere automobila i lidara. Rješenje detekcije magle bazirano je na tezi kako sliku maglovitog vremena opisuje smanjen kontrast i zamućenje koje su prezentirane nižim frekvencijskim spektrom. Algoritam koji detektira maglu baziran je na slikama dobivenih s kamere automobila. Iz slika su prvotno izračunate značajke korištenjem prozora i izračunavanjem brze Fourierove transformacije (engl. *Fast Fourier Transform* – FFT) na tim prozorima. Nakon izračunavanja značajki, njihov broj je smanjen korištenjem PCA tehnike (engl. *Principal Component Analysis*). Dobivene značajke predstavljaju ulaz u SVM (engl. *Support Vector Machine*) klasifikator s RBF (engl. *Radial Basis Function*) jezgrom. Preciznost detekcije maglovitog vremena na temelju slika s kamere iznosi 96%.

Slikom 2.3. prikazana je scena vožnje automobila s vidljivim oblacima detektiranih kapljica kiše korištenjem lidara označenih zelenim i crvenim točkama. Najveća mana ovog rješenja je visoka cijena lidar senzora.



**Slika 2.3.** Scenarij s detektiranim kapljicama kiše [5].

U radu [6] se pokušava pružiti uvid u tehničku problematiku i rješenje problema klasifikacije dvaju vremenskih uvjeta: sunčano i oblačno. U radu su adresirane različite varijacije rješenja na temelju općenitih slika: rješenje pomoću standardne konvolucijske mreže s i bez korištenja umjetnog povećanja skupa za učenje (engl. *Data Augmentation* – DA) te rješenje koje je predstavljeno u radu, također s i bez umjetnog povećanja skupa za učenje. Povećanje podataka proces je umjetnog proširivanja baze dobivanjem podataka iz već postojećih podataka različitim tehnikama. Transformacije metode povećanja podataka predstavljaju veliki spektar operacija iz područja manipulacije i obrade slika, kao što su pomaci, zum, zakretanje za određeni kut itd. Slikom 2.4. prikazan je primjer kreiranih uzoraka korištenjem lijeve rotacije, desne rotacije i okomitog zakretanja.



**Slika 2.4.** Primjer DA korištenjem lijeve rotacije, desne rotacije i okomitog zakretanja [7].

Za rješenje problema klasifikacije dvaju vremenskih uvjeta korišten je model koji na ulaz prima sljedeće značajke koje se prvotno izvlače iz slike:

1. vremenske značajke:
  - nebo, predstavljeno značajkom koja sadrži SIFT deskriptor (engl. *Scale-invariant feature transform*) i srednju vrijednost HSV (engl. *Hue, saturation, value*),
  - sjene, koje se puno više ističu za vrijeme sunčanog vremena,
  - refleksije, koje su također značajka sunčanog vremena,
  - kontrast, koji se računa lokalno i globalno te se značajno razlikuje za vrijeme sunčanog i oblačnog vremena, i
  - izmaglica, koja se može pojaviti s oblačnim vremenom.
2. CNN značajku, koja uključuje globalne diskriminacijske informacije.

Korištenjem vremenskih značajki uz rad konvolucijske mreže postižu se bolji rezultati od rješenja koje koristi samo konvolucijsku neuronsku mrežu. No to dodatno povećava kompleksnost rješenja, a samim time i konzumaciju procesnih resursa koji su od velike važnosti u sustavima stvarnog vremena kao što su autonomna vozila. Rješenje postiže najveću točnost klasifikacije uz korištenje umjetnog povećanja skupa za učenje. Usporedbom rješenja s i bez umjetnog povećanja skupa za učenje možemo zaključiti da umjetno povećanje skupa za učenje može znatno doprinijeti povećanju točnosti klasifikacije. Nedostatak ovog rada je što predstavljeno rješenje razaznaje samo dvije klase dok je u autonomnoj vožnji potrebna detaljna klasifikacija trenutnih vremenskih uvjeta u okolini vozila.

U radu [8] predstavljeno je rješenje problema klasifikacije vremenskih uvjeta korištenjem konvolucijskih neuronskih mreža i SVM klasifikatora. Slike su najprije razdijeljene na područja (engl. *Supapixel Map - SP*), nakon čega su dovedene na ulaz unaprijed trenirane konvolucijske mreže. Izlazi konvolucijske mreže, takozvane duboke značajke (engl. *Deep Features*), su zatim poslani na ulaz SVM klasifikatora. Predstavljena su rješenja koja koriste cijelu sliku i ona koja sliku razdjeljuju na dvadeset pet, pedeset, sedamdeset pet i sto SP-a. Testirano je deset rješenja koja koriste deset različitih unaprijed treniranih mreža, od kojih su samo četiri rješenja pokazala dosljedno povećanje točnosti povećanjem broja SP-a do sto. Nedostaci ovog rješenja najviše se očituju u kompleksnosti rješenja te veličini unaprijed treniranih konvolucijskih mreža koje imaju velike memorijske i računске zahtjeve.

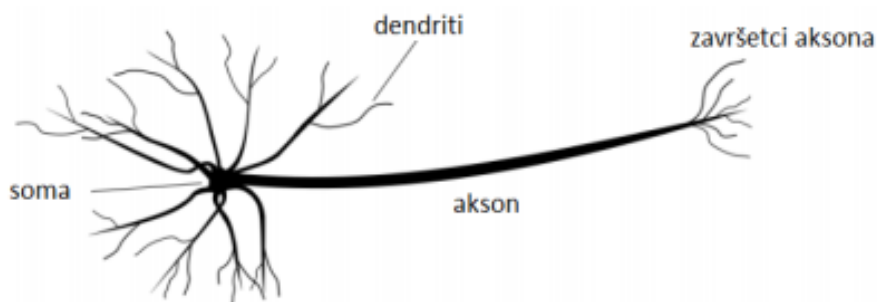
Radom [9] predstavljeno je rješenje problema klasifikacije vremenskih uvjeta u okolini automobila korištenjem lidara. Rješenje predstavljeno u radu klasificira tri vremenska uvjeta:

čisto, kišovito i maglovito vrijeme, a za rješavanje problema korištena su dva lidar senzora: VLP16 tvrtke Velodyne i Scala tvrtke Valeo. VLP16 koristi rotor koji okreće odašiljač i prijemnik dok su kod Scalae odašiljač i prijemnik fiksni, a signal odašilju uz pomoć rotirajućeg ogledala. Testirana su dva modela učena nad podacima dobivenih lidarom: kNN i SVM Rješenje bazirano na VLP16 lidarom daje bolje rezultate s točnosti od 97.14% korištenjem SVM modela i s točnosti od 96.39% korištenjem kNN modela. Iako pristup daje iznimno dobre rezultate, detektira samo tri vremenska uvjeta (kišovito, maglovito i čisto) i koristi cjenovno nepristupačne lidar senzore.

## 3. NEURONSKE MREŽE

### 3.1. Uvod

Dugačak je tijekom evolucije dao ljudskom mozgu mnogo poželjnih karakteristika poput masivne paralelizacije, distribuiranog prikazivanja i računanja, sposobnosti učenja, sposobnosti generalizacije, adaptivnosti, inherentne obrade kontekstualnih informacija, tolerancije na pogreške i niske potrošnje energije za navedeno. Ljudi se ne mogu uspoređivati s modernim računalima kada je riječ o numeričkom računanju i manipulaciji podataka. No ljudi mogu bez većeg napora riješiti kompleksne probleme zapažanja koji su računalima još uvijek računski zahtjevni i nedostižni [10]. Iz tog se razloga mnoge karakteristike koje ljudi imaju nastoje emulirati u modernim računalima. Iako postoje strukture koje pokušavaju opisati biološku neuronsku mrežu, te strukture su još uvijek velikim dijelom različite od tih mreža. Slikom 3.1. prikazan je izgled biološkog neurona, temeljne jedinice središnjega živčanog sustava. Biološki neuron sastoji se od tijela (soma), produžetaka (dendriti) koji dovode živčane impulse s drugih živčanih stanica, te produžetka (akson) koji prenosi živčane impulse na druge živčane stanice.

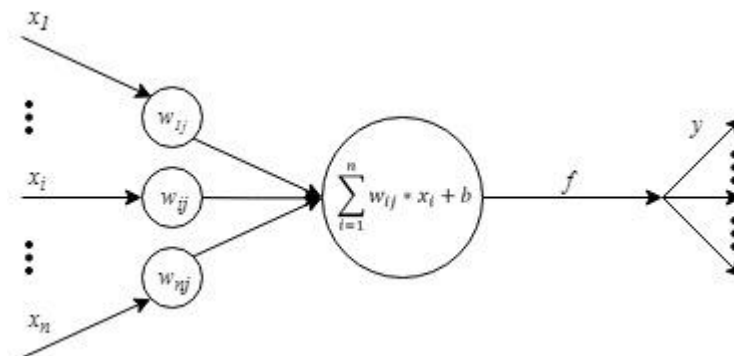


**Slika 3.1.** Izgled biološkog neurona [11].

Umjetna neuronska mreža (engl. *Artificial Neural Network* - ANN) predstavlja matematički model koji je nastao po uzoru na biološku neuronsku mrežu. ANN paralelna je struktura čiji je zadatak procesiranje distribuiranih informacija i koja se sastoji od elemenata za obradu (umjetnih neurona) međusobno povezanih s jednosmjernim signalnim kanalima zvanim veze. Svaki element obrade ima jednu izlaznu vezu koja se grana u mnogo srodnih veza koje nose jednaki signal. Element obrade može biti matematički bilo kojeg oblika, a sva obrada koja

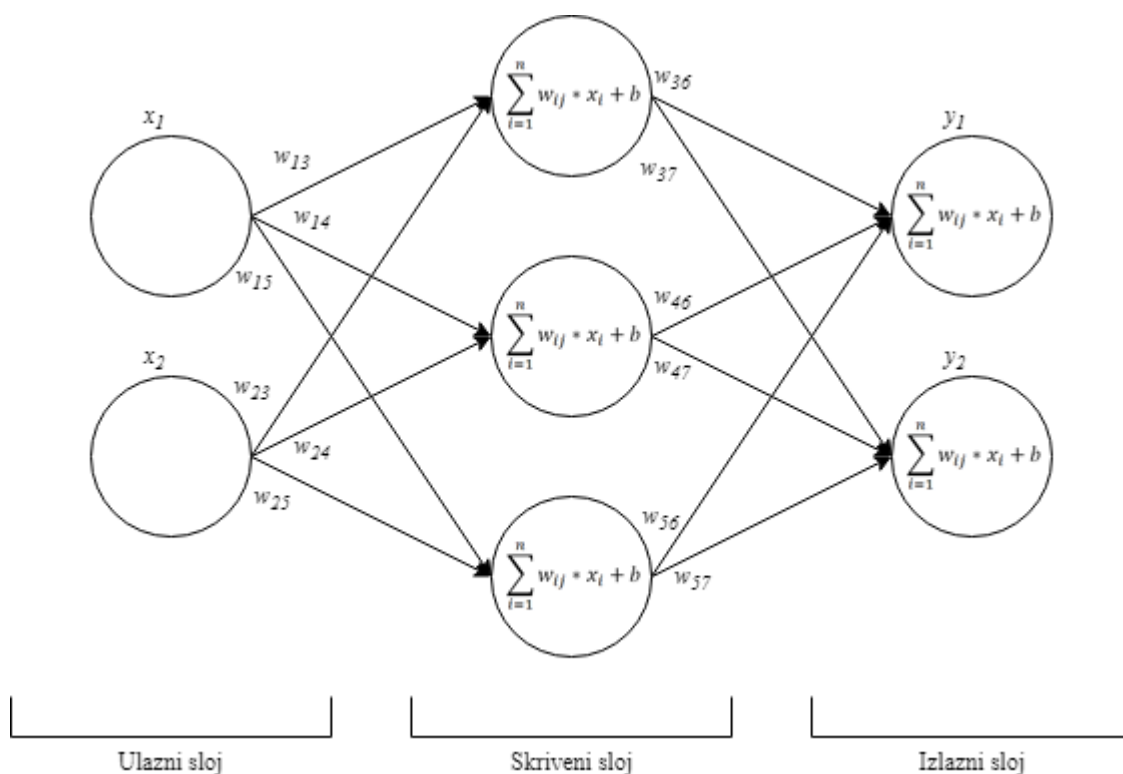
se događa unutar svakog elementa mora biti potpuno lokalna, tj. mora ovisiti samo o trenutnim vrijednostima ulaznog signala i vrijednostima koje su spremljene u lokalnoj memoriji elementa [12]. Umjetne neuronske mreže koriste se za rješavanje različitih problema poput klasifikacije uzoraka, grupiranja, kategorizacije, aproksimacije funkcija, prognoziranja, optimiziranja (npr. problem trgovačkog putnika), upravljanja dinamičkim sustavima (automatizacija) i slično.

Umjetna neuronska mreža se sastoji od ulaznog, izlaznog i skrivenih slojeva. Svaki sloj se sastoji od neurona koje sadrže pomak (engl. *bias*), a međusobno su povezani vezama koje se predstavljaju težinama (engl. *weights*). Težine i pomaci su parametri neuronske mreže koji se određuju postupkom učenja nad danim podacima. Slikom 3.2. prikazan je matematički model umjetnog neurona, i usporedbom slike 3.1. sa slikom 3.2. može se primijetiti da su dendriti u umjetnoj neuronskoj mreži prezentirani ulazima u ćeliju, završetci aksona (telodendroni) izlazima iz ćelija ( $y_1, y_2, \dots, y_n$ ), težine sinapsama ( $w_{1j}, w_{2j}, \dots, w_{nj}$ ), a tijelo neurona samom ćelijom u kojoj se zbrajaju svi ulazi ( $x_1, x_2, \dots, x_n$ ) pomnoženi pripadajućim težinama i pomak ( $b$ ) tog neurona. Bitno je spomenuti i aktivacijsku funkciju, na slici 3.2. označenoj s  $f$ , koja oblikuje vrijednosti izlaza neurona.



**Slika 3.2.** Prikaz matematičkog modela neurona.

Slikom 3.3. prikazana je struktura unaprijedne potpuno povezane neuronske mreže s jednim skrivenim slojem, dva ulaza, dva izlaza i tri neurona u skrivenom sloju.



**Slika 3.3.** Struktura neuronske mreže s jednim skrivenim slojem.

## 3.2. Duboke neuronske mreže

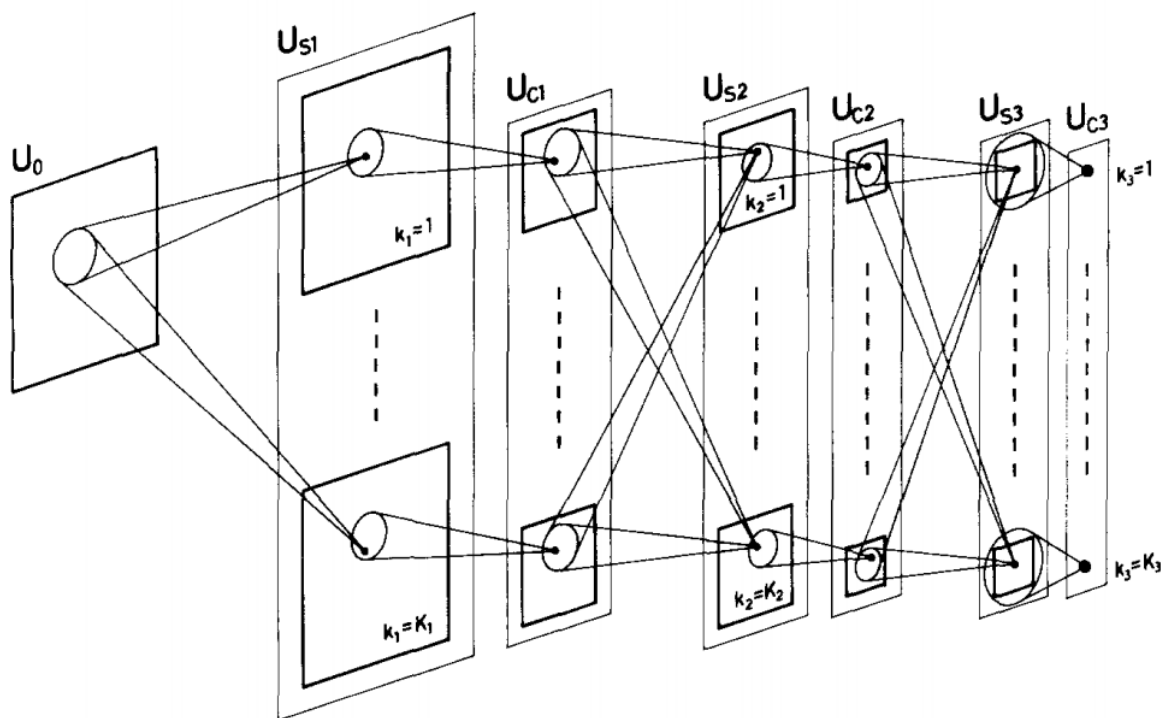
Duboke neuronske mreže su neuronske mreže koje imaju više od jednog skrivenog sloja između ulaznog i izlaznog sloja. Svaki je sloj neurona duboke neuronske mreže treniran na skupu značajki koje se temelje na izlazu prethodnog sloja što rezultira prepoznavanjem kompleksnijih značajki. Ta hijerarhija povećavanja kompleksnosti i apstrakcije poznata je pod nazivom hijerarhija značajki. Dubinu same neuronske mreže definira broj slojeva same mreže. Neke od značajnijih primjena dubokih neuronskih mreža su obrada slike (konvolucijske neuronske mreže, kapsulne mreže), obradi teksta (rekurentne neuronske mreže) itd.

### 3.2.1. Konvolucijske neuronske mreže

Konvolucijska neuronska mreža jedan je od najkorištenijih modela dubokih neuronskih mreža u rješavanju različitih problema iz domene računalnog vida. Konvolucijska neuronska mreža sadrži više slojeva neurona, od kojih svaki predstavlja nelinearnu operaciju nad linearnom transformacijom izlaza prethodnog sloja. Svoj naziv, konvolucijska neuronska mreža, dobila je po matematičkoj operaciji konvoluciji, koja je osnova konvolucijskog sloja.



Prvu neuronsku mrežu za raspoznavanje vizualnih uzoraka predstavio je Kunihiro Fukushima 1980. godine [13]. Neuronska mreža nazvana je Neocognitron, a imala je sposobnost prepoznavanja obrazaca baziranih na geometrijskoj sličnosti oblika bez utjecaja njihove pozicije na slici. Neocognitron se sastojao od ulaznog sloja nakon kojega je slijedila kaskada modularnih struktura, od kojih se svaka struktura sastojala iz dva sloja ćelija povezanih u kaskadu. Struktura neuronske mreže Neocognitron prikazana je slikom 3.4. Iako su u to vrijeme matematički modeli poput Neocognitrona postojali, kompjuterska moć bila je daleko od adekvatne za realizaciju takvih modela.

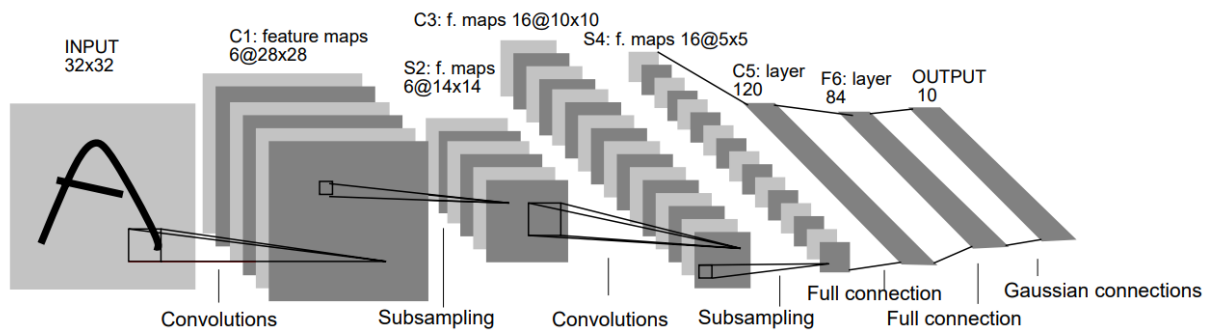


**Slika 3.4.** Prikaz strukture Neocognitron mreže [13].

Vrijedan spomena je i rad [14] u kojem je opisano korištenje umjetne neuronske mreže koja je posebna po tome što su se generalizacijska svojstva neurona poboljšala zamjenom množenja s konvolucijom.

Za razvoj modernih konvolucijskih neuronskih mreža bitan je i rad [15] koji se često citira kao rad u kojem je prvi put u potpunosti opisana konvolucijska neuronska mreža. U tom radu je predstavljena LeNet-5 konvolucijska neuronska mreža, čija je struktura prikazana slikom 3.5. Konvolucijska mreža kombinira tri arhitektonske ideje kako bi osigurala određen stupanj pomaka, mjerila i invarijantnosti na izobličenja: lokalna receptivna polja, dijeljene težine i

prostorno poduzorkovanje. Lokalizirano povezivanje unutar neuronske mreže i ranije je korišteno u modelima korištenim u računalnom vidu.

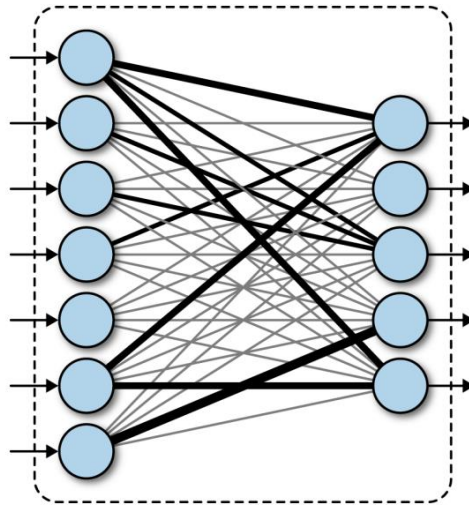


**Slika 3.5.** Struktura LeNet-5 konvolucijske neuronske mreže [15].

LeNet-5 je konvolucijska neuronska mreža koja se sastoji od sedam slojeva, ne uključujući izlazni sloj. Sloj C1 je konvolucijski sloj sa šest mapa značajki veličine 28x28. Svaki član mape je povezan s 5x5 susjednih članova ulaznog sloja. S2 sloj predstavlja sloj poduzorkovanja sa šest mapa značajki veličine 14x14. Svaki član pojedine mape značajki je povezan s 2x2 susjednih članova prvog konvolucijskog sloja. Nakon S2 sloja slijedi još jedan konvolucijski sloj (C3) i sloj poduzorkovanja (S4) nakon kojih slijede potpuno povezani slojevi (C5 i F6) i izlazni sloj.

### 3.2.2. Potpuno povezani sloj

Dva susjedna sloja su potpuno povezana (engl. *Fully Connected* - FC) ako je svaki neuron jednog sloja povezan sa svim neuronima prethodnog sloja. Potpuno povezani slojevi neuronske mreže prikazani su slikom 3.6. gdje su vrijednosti težina predstavljene debljinom veza između neurona. Takvi slojevi imaju mnogo parametara, čak i za relativno mali broj ulaza.



**Slika 3.6.** Potpuno povezani sloj [18].

### 3.2.3. Konvolucijski sloj

Konvolucijski sloj je bazna jedinica konvolucijskih neuronskih mreža i u njemu se obavlja većina transformacije podataka. Konvolucija nad dvodimenzionalnim signalima opisana je matematičkim izrazima (3-1) i (3-2), gdje su  $x(m,n)$ ,  $y(m,n)$  i  $h(m,n)$  dvodimenzionalni vremenski diskretni signali. U modelu konvolucijskih neuronskih mreža diskretni signal  $x(m,n)$  predstavlja podatke na ulazu,  $h(m,n)$  predstavlja filter, a proizvod konvolucije, odnosno izlazni diskretni signal  $y(m,n)$  predstavlja aktivacijsku mapu. Treba napomenuti da je indeksiranje filtera drugačije i da je  $h(0, 0)$  pozicioniran u sredini filtera, a ne na početku.

$$y(m, n) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x(i, j)h(m - i, n - j) \#(3 - 1)$$

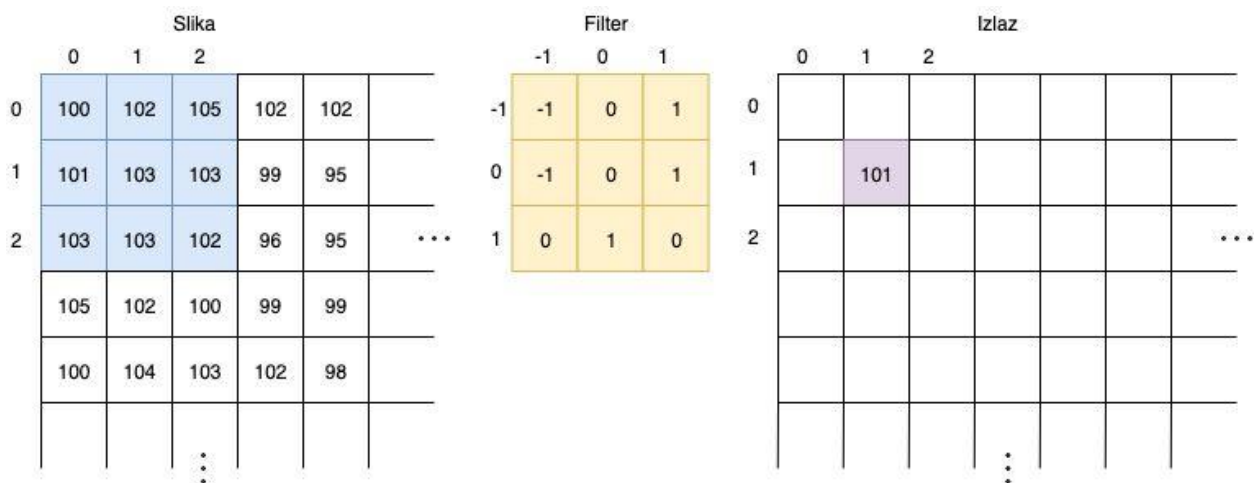
$$y(m, n) = x(m, n) * h(m, n) \#(3 - 2)$$

Konvolucijski sloj se sastoji od seta filtera čiji se parametri prilagođavaju procesom učenja neuronske mreže. Dimenzije filtra se unaprijed definiraju i moraju biti manje od dimenzija ulaza. Filter se pomiče po dužini i širini ulaza definiranim hodom i tako kreira dvodimenzionalnu mapu vrijednosti koje su produkt ulaza i filtra. Izlaz konvolucijskog sloja dobiva se slaganjem aktivacijskih mapa svih filtera. Svaka čestica aktivacijske mape je povezana s malom regijom ulaznih podataka koja korespondira veličini korištenog filtra. Lokalna povezanost konvolucijskog sloja omogućuje neuronskoj mreži da prilagodi filtre koji će maksimalno reagirati na lokalnu regiju ulaza i tako iskoristavati lokalnu korelaciju ulaza. Korištenje ovakvih

modela smanjuje broj parametara, u odnosu na situaciju kada koristimo FC sloj, čime se dobiva veća učinkovitost učenja, odnosno lakše učenje i bolja generalizacija [16].

Dvodimenzionalna konvolucija prikazana je slikom 3.7. Izračun elementa izlaza  $y(1,1)$  prikazan je sljedećim:

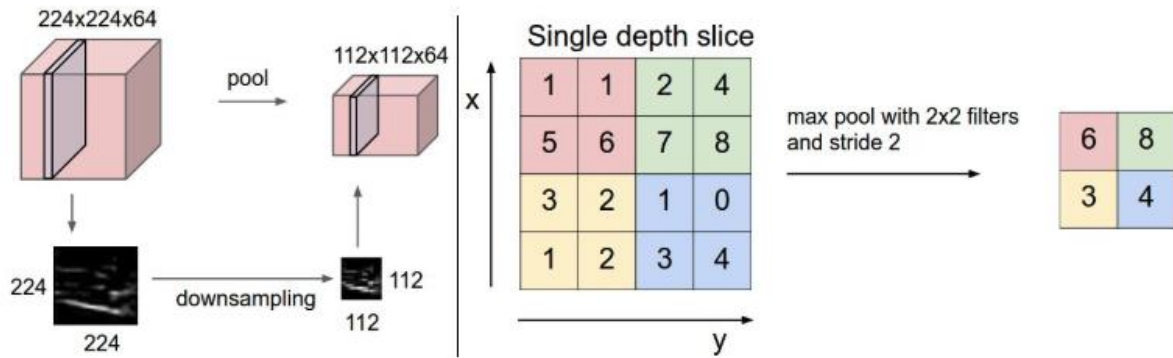
$$\begin{aligned}
 y(1,1) &= \sum_{j=0}^2 \sum_{i=0}^2 x(i,j)h(1-i, 1-j) \\
 &= 0 + 102 + 0 + 101 + 0 - 103 + 103 + 0 - 102 \\
 &= 101
 \end{aligned}$$



Slika 3.7. Konvolucija dvodimenzionalnog signala.

### 3.2.4. Sloj sažimanja

Sloj sažimanja (engl. *Pooling layer*) je sloj čija je zadaća postupno smanjenje prostorne veličine objekta s ciljem smanjenja količine učenih parametara u mreži. Postoje dvije metode sažimanja, a to su sažimanje po maksimalnoj vrijednosti (engl. *Max Pooling*), i sažimanje po srednjoj vrijednosti (engl. *Mean Pooling*). Sloj sažimanja se obično koristi između uzastopnih konvolucijskih slojeva i djeluje nezavisno na svakoj dubini ulaza. Sloj sažimanja po maksimalnoj vrijednosti prostorno mijenja ulaz koristeći operaciju MAX. Najčešći oblik sloja sažimanja je sloj s filtrima veličine 2x2, koji na svim dubinama poduzorkuju ulaz za 2 po visini i širini s korakom od 2. Slikom 3.8. prikazan je rad sloja sažimanja po maksimalnoj vrijednosti s filtrima veličine 2x2 i korakom 2.



Slika 3.8. Sloj sažimanja [17].

### 3.2.5. SoftMax sloj

*SoftMax* je sloj koji u višeklasnu klasifikaciju uvodi ideju izlazne vrijednosti kao vjerojatnosti. Ta je ideja u jednoklasnoj klasifikaciji predstavljena logističkom regresijom koja kao rezultat daje broj u rasponu od nula do jedan koji se može interpretirati kao sigurnost točne klasifikacije. Ako je izlaz logističke regresije na primjer 0.7, to znači da je klasifikator 70% siguran da objekt pripada nekoj klasi, dok je 30% siguran da ne pripada. Izlaz u tom obliku je puno informativniji od izlaza nekih aktivacijskih funkcija i iz tog se razloga na izlazu višeklasnog klasifikatora često koristi *SoftMax* sloj, koji na izlaz daje vjerojatnosti pripadanja klasificiranog objekta svakoj od  $K$  klasa. *SoftMax* sloj predstavlja FC sloj koji kao aktivacijsku funkciju koristi *SoftMax* aktivacijsku funkciju definiranu izrazom (3-4) gdje veličina  $z_i$  predstavlja zbroj  $n$  ponderiranih ulaznih veličina i pomaka  $i$ -tog neurona, a opisana je izrazom (3-3).

$$z_i = \sum_{j=1}^n w_{ji} * x_j + b_i \quad \#(3-3)$$

$$z_K = \frac{e^{z_K}}{\sum_{k=1}^K e^{z_k}} \quad \#(3-4)$$

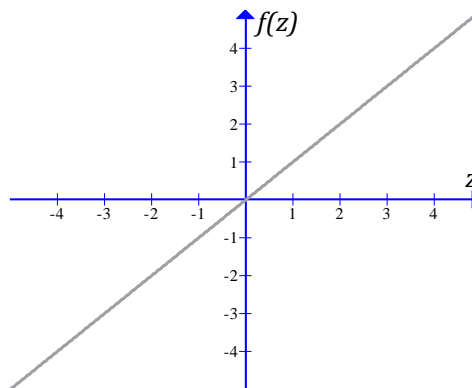
### 3.2.6. Aktivacijska funkcija

Glavna zadaća umjetnog neurona je da zbroji ulazne veličine pomnožene pripadajućim težinama, vrijednosti pridoda pomak i rezultat prosljedi na izlaz. Zadaća aktivacijske funkcije je

da za navedeni zbroj izračuna izlaznu vrijednost neurona. Postoje različiti oblici aktivacijskih funkcija poput linearne, sigmoidne, Tanh, ReLu, *SoftMax* itd.

Linearna aktivacijska funkcija dana je jednadžbom (3-5) i prikazana slikom 3.9. gdje je veličina  $z$  opisana izrazom (3-3). Aktivacija je proporcionalna ulazu, odnosno težinskoj sumi neurona.

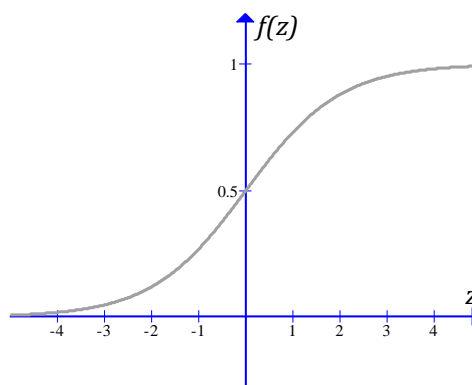
$$y = f(z) = z \quad (3-5)$$



**Slika 3.9.** Linearna aktivacijska funkcija.

Sigmoidna funkcija je ograničena realna funkcija definirana za sve realne ulazne vrijednosti i koja svugdje ima pozitivnu derivaciju [19]. Koristi se u slučajevima koji zahtijevaju vjerojatnost kao izlaz jer izlaz sigmoidne funkcije poprima vrijednost u intervalu  $[0, 1]$ . Svoj naziv sigmoidna funkcija dobila je po svom obliku i u prijevodi sigmoid znači „S-oblika“. Sigmoidna aktivacijska funkcija opisana je jednadžbom (3-6) [20] i prikazana slikom 3.10.

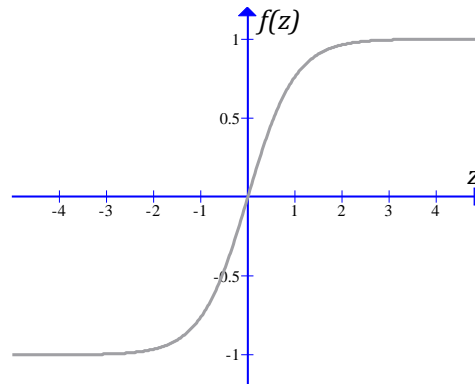
$$f(z) = \frac{1}{1 + e^{-z}} \quad (3-6)$$



**Slika 3.10.** Sigmoidna aktivacijska funkcija.

Hiperbolična tangent funkcija ima slične karakteristike kao sigmoidna funkcija, ali poprima vrijednosti u intervalu  $[-1, 1]$  kao što je prikazano slikom 3.11. Funkcija je definirana kao omjer hiperbolične sinusne i kosinusne funkcije kao što je prikazano jednadžbom (3-7) [20].

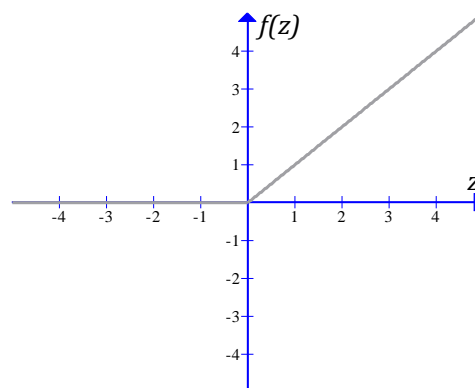
$$f(z) = \tanh(z) = \frac{\sinh(z)}{\cosh(z)} \quad \#(3 - 7)$$



**Slika 3.11.** Tanh aktivacijska funkcija.

Ispravljena linearna aktivacijska funkcija (engl. *Rectified Linear Activation Function* - ReLu) je djelomice linearna funkcija koja aktivira neuron za sve pozitivne ulaze, dok za negativne ulaze na izlaz daje vrijednost nula. ReLu aktivacijska funkcija koristi se za različite vrste neuronskih mreža jer je model koji ga koristi lako trenirati i često daje dobre performanse. ReLu aktivacijska funkcija opisana je jednadžbom (3-8) [21] i prikazana slikom 3.12.

$$f(z) = \max(0, z) \quad \#(3 - 8)$$



**Slika 3.12.** ReLu aktivacijska funkcija.

### 3.3. Učenje i evaluacija neuronskih mreža

Učenje neuronskih mreža podrazumijeva podešavanje parametara mreže tako da što bolje rješavaju dani problem prema nekom unaprijed definiranom kriteriju. U slučaju konvolucijskih neuronskih mreža parametri su težine i pomaci filtara konvolucijskih slojeva te težine i pomaci potpuno povezanih slojeva mreže. Količinu parametara konvolucijske neuronske mreže određuju: dimenzija ulaznog podatka, broj i veličine filtera konvolucijskih slojeva te broj neurona FC slojeva. Postoje razne metode učenja neuronskih mreža koje se odabiru ovisno o problemu kojeg neuronska mreža treba riješiti, tipu ulaznih podataka itd. Neuronske mreže koje rješavaju problem klasifikacije uče se metodom nadziranog učenja. Za nadzirano učenje je specifično to što su nam poznate oznake klasa kojima pripadaju uzorci iz prostora ulaznih podataka. Učenje se odvija u epohama (engl. *epoch*) koje se nadalje izvode u više koraka. Broj koraka definiran je omjerom količine podataka za treniranje i veličine grupe podataka (engl. *batch size*) korištene u procesu učenja. Da bi generalizacijska svojstva učene mreže bila što bolja bitno je imati što raznovrsnije podatke.

#### 3.3.1. Funkcija gubitka

Funkcija gubitka je matematička funkcija korištena u treniranju neuronske mreže čija je zadaća evaluacija modela na danim podacima za učenje. Evaluacija modela se provodi tako što funkcija gubitka mjeri odstupanje izlaza iz modela od stvarnih vrijednosti pridruženih ulaznim podacima. Što su izlazi iz modela „bliži“ stvarnim vrijednostima, odnosno što je više objekata dobro klasificirano, gubitak će biti manji. Funkcija gubitka ima veliku ulogu u procesu učenja mreže, jer se na parametre (težine i pomake) utječe u smjeru njene minimizacije.

Postoje dvije vrste prolaska mrežom za vrijeme učenja, prolazak prema naprijed (engl. *forward propagation*) i prolazak unatrag (engl. *back propagation*). Prolazak prema naprijed označava prolazak ulaznih podataka kroz cijelu mrežu do izlaza i taj prolazak ne utječe na parametre mreže. Prolazak unatrag s druge strane služi za prijenos pogreške s viših slojeva mreže na niže što rezultira hijerarhijskim podešavanjem parametara. Metoda najbržeg spusta (engl. *Gradient Descent method* – GD) je najčešće korišten algoritam optimizacije u području ML-a. Metoda najbržeg spusta započinje inicijalizacijom parametara, stope učenja i ciljane vrijednosti, nakon čega započinje iterativni postupak podešavanja parametara mreže koji završava kada funkcija gubitka vrati manju vrijednost od ciljane vrijednosti. U iterativnom postupku metoda najbržeg spusta zadanom stopom mijenja parametre u smjeru minimizacije funkcije gubitka [22].



Neki od primjera funkcije gubitka su srednja kvadratna pogreška (engl. *Mean Squared Error* - MSE), vjerojatnosna funkcija gubitka (engl. *Likelihood Loss*), unakrsna entropija (engl. *Cross Entropy Loss*), itd.

Neka je  $y_i$  oznaka ulaznog podatka  $x_i$ ,  $\hat{y}_i$  izlaz klasifikatora za ulazni podatak  $x_i$ , a  $n$  broj raspoloživih podataka. Srednja kvadratna pogreška najčešće je korištena regresijska funkcija gubitka. MSE predstavlja sumu kvadrata razlike između stvarne i predviđene vrijednosti za svaki uzorak ulaznih veličina  $x_i$ . MSE funkcija gubitka definirana je jednadžbom:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad \# (3 - 9)$$

Unakrsna entropija se pojavljuje u binarnom i kategoričkom obliku. Kategorička unakrsna entropija se koristi kada je poželjna probabilistička interpretacija rezultata. Kategorička unakrsna entropija mjeri različitost između stvarne distribucije  $y$  i predviđene distribucije  $\hat{y}$ , a definirana je jednadžbom [23]:

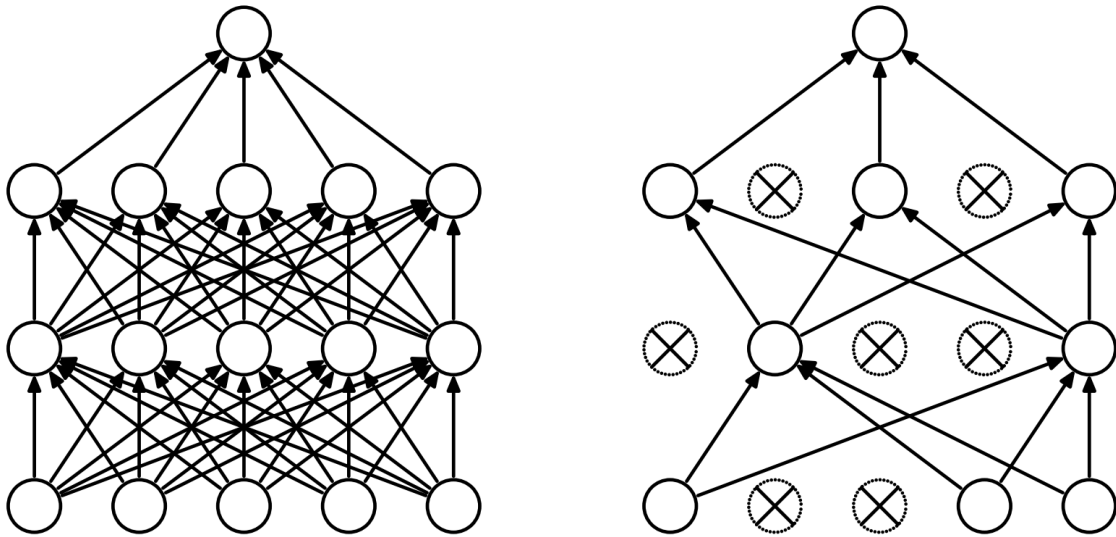
$$L_{unakrsna\ entropija}(\hat{y}, y) = - \sum_i y_i \log(\hat{y}_i) \quad \#(3 - 10)$$

### 3.3.2. Slučajno izostavljanje neurona

Duboke neuronske mreže sadrže više nelinearnih skrivenih slojeva, što ih čini vrlo ekspresivnim modelima koji mogu naučiti vrlo komplicirane odnose između ulaza. S obzirom na to da su podaci za treniranje ograničeni, mnogi od tih kompliciranih odnosa će biti rezultat različitih šumova sadržanih u skupu za treniranje. Učenje takvih odnosa rezultira pre-naučenosti (engl. *Overfitting*) mreže. Zbog toga se javlja potreba za rješavanjem tog problema. Neke od metoda prevencije pre-naučenosti mreže jesu uvođenje penalizacije težina poput L1 i L2 regularizacija te mekog dijeljenja težina.

Slučajno izostavljanje neurona prilikom učenja (engl. *Dropout*) je tehnika koja se koristi u neuronskim mrežama s ciljem prevencije pre-naučenosti mreže. Jedan je od pristupa regularizacije u neuronskim mrežama koji pomaže u smanjenju međuovisnog učenja među neuronima. Tijekom treninga mreže dio je neurona na pojedinom sloju deaktiviran, čime se poboljšava generalizacija mreže jer prisiljava slojeve da uče isti koncept s različitim vezama, odnosno neuronima. Nakon treniranja mreže, za vrijeme predviđanja, neuroni izostavljeni prilikom učenja se ponovno aktiviraju. Pod izostavljanje se misli na privremeno izbacivanje

neurona iz neuronske mreže, uključujući sve ulazne i izlazne veze s izbačenim neuronom, kao što je prikazano slikom 3.13.



**Slika 3.13.** Model neuronske mreže s dva skrivena sloja. Lijevo: Standardna neuronska mreža.  
Desno: Neuronska mreža dobivena primjenom slučajnog izostavljanja [24].

## 4. DETEKCIJA VREMENSKIH UVJETA U OKOLINI AUTOMOBILA POMOĆU DUBOKIH NEURONSKIH MREŽA

Kako bi bilo moguće kreirati rješenje problema detekcije vremenskih uvjeta u okolini automobila pomoću dubokih neuronskih mreža, potrebno je prvotno prikupiti dovoljnu količinu slika, odrediti strukturu mreže te provesti učenje i validaciju mreže. Za izradu modela korišten je Keras API nad Tensorflow platformom.

Tensorflow je *open-source* platforma za strojno učenje koja sadrži sveobuhvatan, fleksibilan ekosustav alata, knjižnica i zajednica koji razvojnim inženjerima omogućuju laku izradu i implementaciju aplikacija strojnog učenja. Tensorflow omogućava jednostavno modeliranje korištenjem sučelja za programiranje aplikacija (engl. *Application Programming Interface* - API) više razine poput Kerasa, robusnu izradu ML-a na različitim platformama i pruža velike mogućnosti eksperimentiranja za različite istraživačke skupine [25].

Keras je API visoke razine primjenjiv za modeliranje neuronskih mreža, napisan u Python skriptnom jeziku i sposoban za rad nad TensorFlow, CNTK ili Theano platformama za strojno učenje. Razvijen je s naglaskom na omogućavanje brzog eksperimentiranja. Keras omogućuje jednostavnu i brzu izradu prototipova modela kroz modularnost, jednostavnost korištenja i proširivost, podržava konvolucijske i rekurentne mreže, kao i kombinacije navedenih te omogućava rad na CPU-u (engl. *Central Processing Unit*) i GPU-u (engl. *Graphics Processing Unit*) [26].

### 4.1. Prikupljanje i predobrada podataka

Podaci, odnosno fotografije okoline automobila, dobivene su iz dva glavna izvora: BDD100K baza podataka [27] i vlastito prikupljene slike koje su dobivene iz video sekvenci snimljenih u gradu Osijeku za vrijeme snježnog i maglovitog vremena. BDD100K baza sadrži najviše slika s oznakama vremenskih uvjeta i vremena dana u više različitih scena od ostalih pronađenih baza podataka.

BDD100K je trenutno najveća baza podataka napravljena da služi kao pomoć u razvoju različitih algoritama autonomne vožnje. Svaka označena slika unutar baze sastoji se iz dva dijela: slike čiji je naziv ID slike te opisa slike i objekata na slici sadržanih u pripadajućoj json datoteci.

Pomoću ID-a, slika se može povezati sa svojim opisom u json datoteci. Primjer objekta sadržanog u json datoteci *bdd100k\_labels\_images\_val.json* s pripadajućim atributima nalazi se u listingu 4.1.:

```
{
  "name": "b1c66a42-6f7d68ca.jpg",
  "attributes": {
    "weather": "rainy",
    "scene": "city street",
    "timeofday": "daytime"
  },
  "timestamp": 10000,
  "labels": [
    {
      "category": "traffic sign",
      "attributes": {
        "occluded": false,
        "truncated": false,
        "trafficLightColor": "none"
      },
      "manualShape": true,
      "manualAttributes": true,
      "box2d": {
        "x1": 270.762627,
        "y1": 288.231184,
        "x2": 309.443003,
        "y2": 301.956478
      },
      "id": 1
    },
  ]
}
```

**Listing 4.1.** Primjer objekta sadržanog u json datoteci s pripadajućim atributima.

Opis slike sadrži dvije korisne informacije za razvoj predloženog rješenja unutar ovog rada: vremenski uvjet i vrijeme dana. Postoji pet klasa vremenskih uvjeta sadržanih u bazi: čisto, kišovito, snježno, maglovito i oblačno te tri klase vremena dana: dan, noć i zora/sumrak. Uzorci, odnosno slike BDDK100K baze dobivene su uzimanjem okvira iz video sekvenci, i to svaku desetu sekundu svakog videa u bazi s pripadajućim opisima. Svaka slika je, uz osnovne informacije, označena na sljedećim razinama: okviri (engl. *Bounding boxes*) objekata na cesti, oznake traka, segmentacija slike i vozna područja.

Uz pomoć oznaka vremenskih uvjeta i vremena dana slike su raspoređene u dvije baze slika: *Weather Detection Classifier dataset* (WDC\_dataset) koja će služiti za treniranje klasifikatora vremenskih uvjeta i *Day Night Classifier dataset* (DNC\_dataset) koja će služiti za treniranje dan-noć klasifikatora.

Uvidom u WDC\_dataset bazu nakon raščlanjivanja slika u datoteke primijećena je prisutnost određenog broja pogrešno označenih slika što je zahtijevalo njihovo uklanjanje iz baze. Također je primijećen manjak slika za snježno i maglovito vrijeme, pa su slike za te dvije

klase dobivene izvlačenjem okvira iz snimljenih video sekvenci. Za izvlačenje okvira korištena je OpenCV (engl. *Open Computer Vision Library*) *open-source* biblioteka za računalni vid i ML. Izvlačenje okvira prikazano je sljedećim isječkom *Python* koda.

```

vid='D:/WorkspaceTriva/vids/foggy_day_mf.mp4'
destinationDir='D:/WorkspaceTriva/Baz/train/foggy/'

vidObj = cv2.VideoCapture(vid)
success = 1
count = 0
while success:
    success, image = vidObj.read()
    if count % 400 == 0:
        cv2.imwrite(destinationDir + "c_d_mf%d.tif" % (count), image)
    count += 1

```

**Listing 4.2.** Kod za izvlačenje okvira iz video sekvenci.

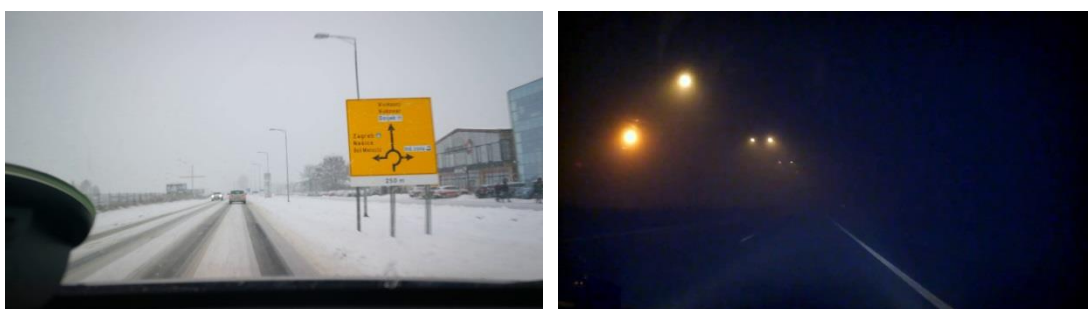
Nakon izvlačenja slika dobivena je konačna WDC\_dataset baza podataka. Tablicom 4.1. prikazan je broj uzoraka za treniranje i validaciju WDC\_dataset i DNC\_dataset baza podataka po klasama. Na slici 4.1. prikazani su primjeri slika dobiveni iz BDD100K baze podataka, dok su slikom 4.2. prikazane slike dobivene izvlačenjem okvira iz video sekvenci.

**Tablica 4.1.** Broj uzoraka za treniranje i validaciju.

		Broj uzoraka za treniranje	Broj uzoraka za validaciju
WDC_dataset	Čisto	3000	500
	Maglovito	2266	500
	Kišovito	3000	500
	Snježno	3000	500
DNC_dataset	Dan	3000	500
	Noć	3000	500



**Slika 4.1.** Primjer slika iz BDD100K baze podataka za čisto i kišovito vrijeme.

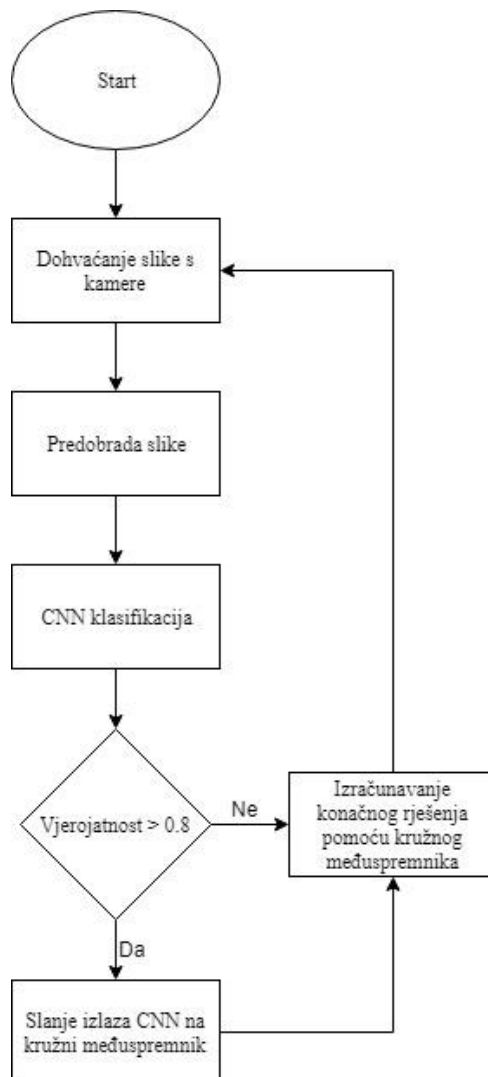


**Slika 4.2.** Primjer slika dobivenih izvlačenjem okvira iz snimljenih video sekvenci za snježno i maglovito vrijeme.

Slike su originalno veličine 1280x720 elemenata. Kao što je spomenuto u prethodnom poglavlju, veličina ulaznog podatka ima direktan utjecaj na broj parametara mreže. Zato što su računalna moć i memorija ugradbenih sustava za autonomnu vožnju često ograničena, poželjno je ispitati rješenje s manjim memorijskim i računalnim zahtjevima. Stoga je veličina ulaznog podatka svedena na 128x72 elemenata. Zbog pretpostavke da krajnji model, koji bi služio za samu klasifikaciju, ne bi mogao dobro prepoznavati kišu, odnosno kapljice kiše, zbog male razlučivosti, napravljena je i baza podataka sa slikama veličina 256x144 elemenata. Podaci su automatski skalirani tijekom postupka učenja pomoću kerala i to tako da je svaki podatak podijeljen s 255, što je maksimalna moguća vrijednost elementa slike.

## 4.2. Prijedlog rješenja detekcije vremenskih uvjeta zasnovano na konvolucijskoj neuronskoj mreži

Dijagram toka predloženog rješenja detekcije vremenskih uvjeta zasnovanog na konvolucijskoj neuronskoj mreži prikazan je slikom 4.3. Rješenje na ulaz prima sliku s kamere koja je nakon toga svedena na veličinu 128x72 elemenata i skalirana dijeljenjem s 255 u bloku predobrade slike. Nakon predobrade, slika se dovodi na ulaz konvolucijske neuronske mreže. Izlazak iz mreže potom prolazi kroz prag koji odbacuje sve izlaze konvolucijske neuronske mreže čije maksimalne vjerojatnosti ne prelaze vrijednost praga koja je određena empirijskim putem. Ako izlaz konvolucijske neuronske mreže prijeđe prag, ulazi u kružni međuspremnik (engl. *circular buffer*), koji nakon toga izračunava krajnji rezultat klasifikacije i vraća se na prvi korak, dohvaćanje slike s kamere.

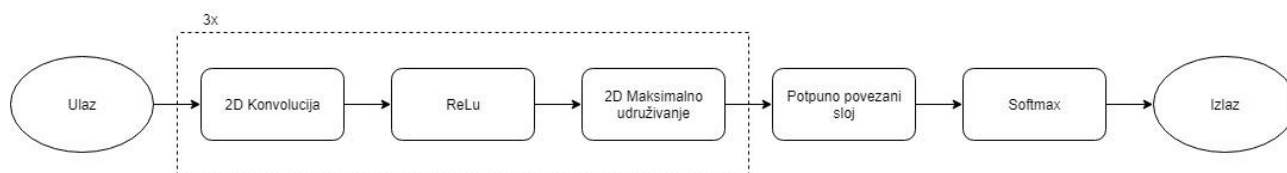


**Slika 4.3.** Dijagram toka predloženog rješenja za detekciju vremenskih uvjeta u okolini automobila.

### 4.2.1. Odabir strukture i podešavanje parametara konvolucijske neuronske mreže

Nakon prikupljanja i predobrade podataka bilo je potrebno istražiti pomoću kojeg modela strojnog učenja pristupiti problemu klasifikacije. Nakon istraživanja postojećih rješenja, tri su se rješenja činila kao dobar pristup rješavanju problema klasifikacije vremenskih uvjeta: podjela slike na interesne regije i izračunavanje vremenskih značajki, konvolucijska neuronska mreža te kombinacija oba. Detaljnijim istraživanjem literature zaključeno je da konvolucijska neuronska mreža ima veći potencijal postati kvalitetan klasifikator od prvog pristupa na temelju čega je odabrana konvolucijska neuronska mreža kao osnova predloženog rješenja u ovom radu.

Kako je rješenje problema namijenjeno autonomnoj vožnji te bi se algoritam trebao moći implementirati na ugradbenom sustavu koji ima ograničenu računalnu moć i memoriju, bitno je odabrati strukturu s što manjim brojem parametara. Odabrana struktura konvolucijske neuronske mreže prikazana je slikom 4.4. te sadrži tri konvolucijska sloja s pripadajućim ReLu aktivacijskim funkcijama i slojevima sažimanja koji koriste sažimanje po maksimalnoj vrijednosti, potpuno povezani sloj nakon kojega slijedi *SoftMax* aktivacija i izlazni sloj. Korištenjem manjeg broja konvolucijskih slojeva mreža daje značajno lošije rezultate, dok se s više od tri konvolucijska sloja nisu postigli bolji rezultati na validacijskom skupu. Prva dva konvolucijska sloja sadrže svaki po trideset dva filtra dimenzija 3x3, dok zadnji konvolucijski sloj sadrži šezdeset četiri filtra istih dimenzija. Filar dimenzija 3x3 je odabran jer je vrlo često korišten i osnova je mnogih modernih konvolucijskih neuronskih mreža. Odabrani filtri slojeva sažimanja su dimenzija 2x2 s korakom 2 jer je to minimalna veličina sažimanja kako prolaskom kroz mrežu ne bi brzo opadala rezolucija slike. Izlazni sloj sadrži četiri neurona koji predstavljaju četiri klase, odnosno četiri vremenska uvjeta: čisto, maglovito, kišovito i snježno vrijeme.



**Slika 4.4.** Struktura predložene konvolucijske neuronske mreže.



Kako je ranije navedeno, kreirane su dvije baze podataka, jedna sa slikama dimenzija 128x72 i jedna sa slikama dimenzija 256x144 elementa. Analogno tome su kreirane i dvije konvolucijske neuronske mreže prikazane slikom 4.4 koje na ulazu primaju slike tih dimenzija. Tablicom 4.2. prikazani su brojevi parametara, veličine modela i brojevi operacija konvolucijskih neuronskih mreža koje na ulaz primaju slike dimenzija 128x72 i 256x144 elementa. Možemo primijetiti da konvolucijska neuronska mreža koja na ulaz prima slike dimenzija 256x144 elementa ima značajno veće računalne i memorijske zahtjeve od mreže koja na ulaz prima slike dimenzija 128x72 elementa.

**Tablica 4.2.** Usporedba predloženih konvolucijskih neuronskih mreža.

Model	Broj parametara	Veličina modela	Broj operacija
CNN (128x72)	430372	1,67 MB	860359
CNN (256x144)	1995044	7.64 MB	3989703

Prilikom učenja se koristila veličina grupe podataka od trideset slika i stopa učenja od 0.0005 koja se tijekom učenja prilagođava smanjenjem za  $10^{-7}$  sa svakom epohom. Za vrijeme učenja mreže korišteno je slučajno izostavljanje neurona od 50%. Funkcija gubitka korištena za vrijeme učenja mreže je kategorička unakrsna entropija. Listingom 4.3. prikazan je kod koji opisuje kreiranje sekvencijalnog modela konvolucijske neuronske mreže, gdje su: *Conv2D* konstruktor za dvodimenzionalni konvolucijski sloj, *MaxPooling2D* konstruktor za sloj sažimanja koji koristi sažimanje po maksimalnoj vrijednosti, *Activation* konstruktor za aktivacijsku funkciju, *Dense* konstruktor za potpuno povezani sloj i *Dropout* konstruktor za uvođenje slučajnog izostavljanja neurona

```

model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

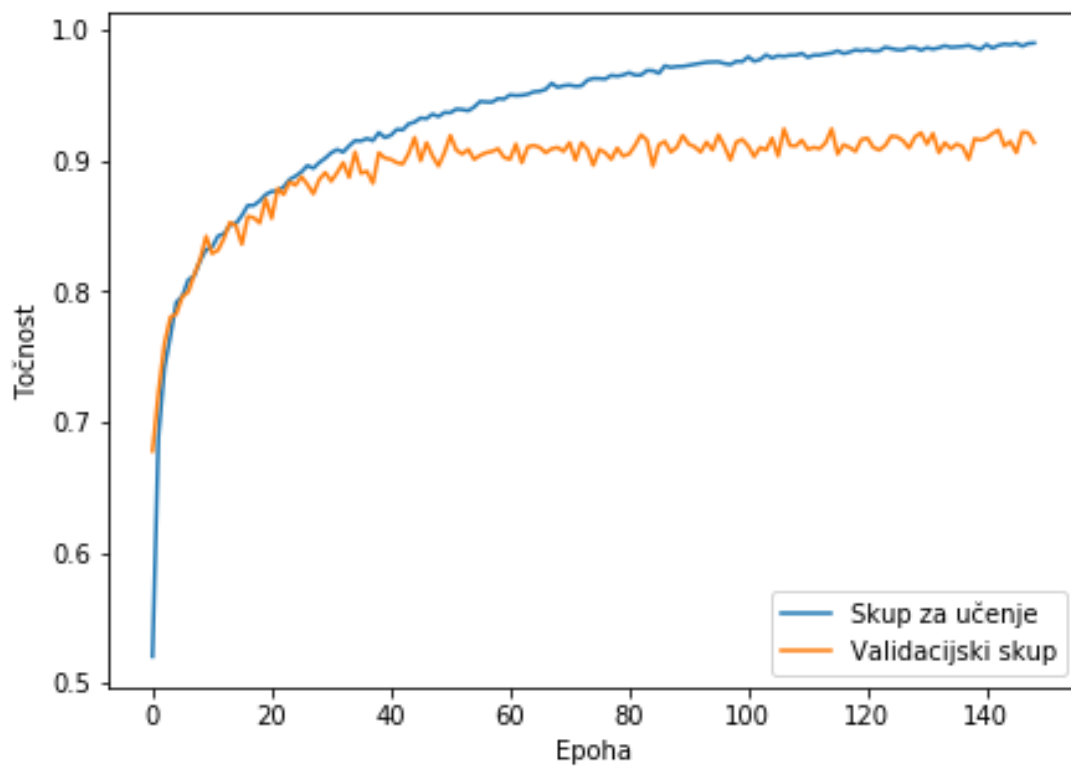
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(4, activation='softmax'))

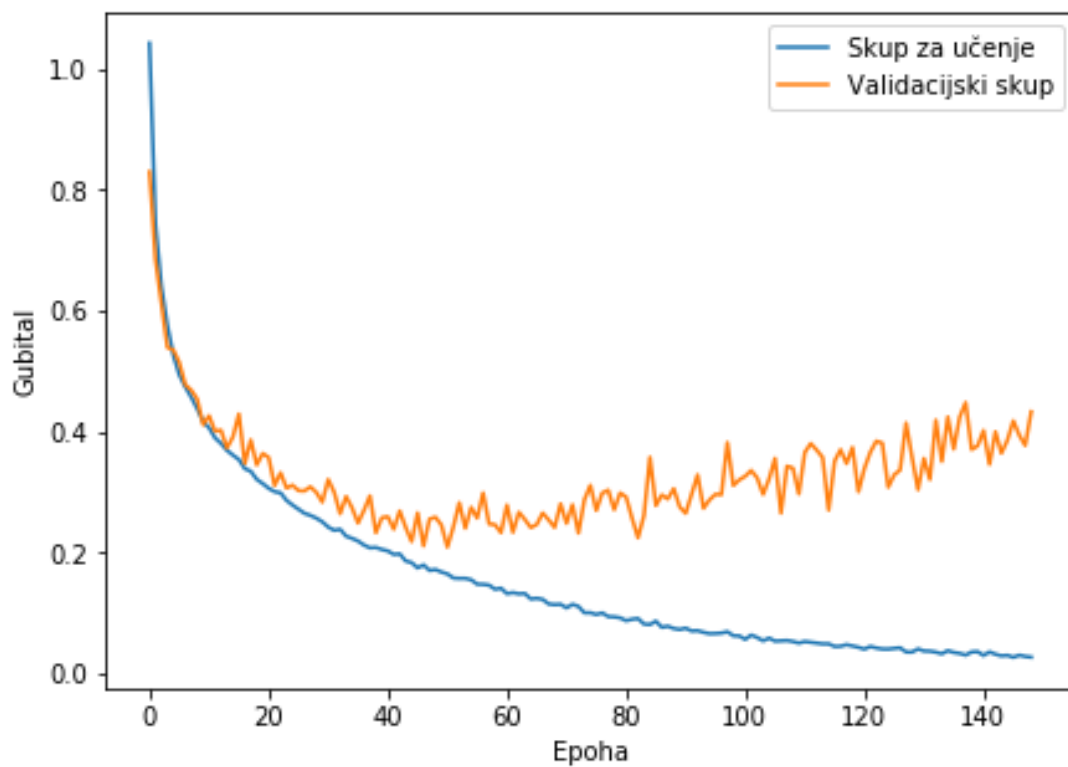
```

**Listing 4.3.** Kod za kreiranje sekvencijalnog modela konvolucijske neuronske mreže.

Učenje mreže provodilo se u sto pedeset epoha, a promjena točnosti i gubitka za vrijeme procesa učenja prikazana je slikama 4.5 i 4.6. Može se primjetiti da je došlo do prenaučnosti mreže tako da je spremljen model s najboljom točnom na validacijskom skupu. Konvolucijska neuronska mreža je nakon sto pedeset epoha treniranja postigla točnost od 97.97% i gubitak od 0.05 na skupu za učenje, dok je na validacijskom skupu podataka postignuta točnost od 92.43% i gubitak od 0.26.



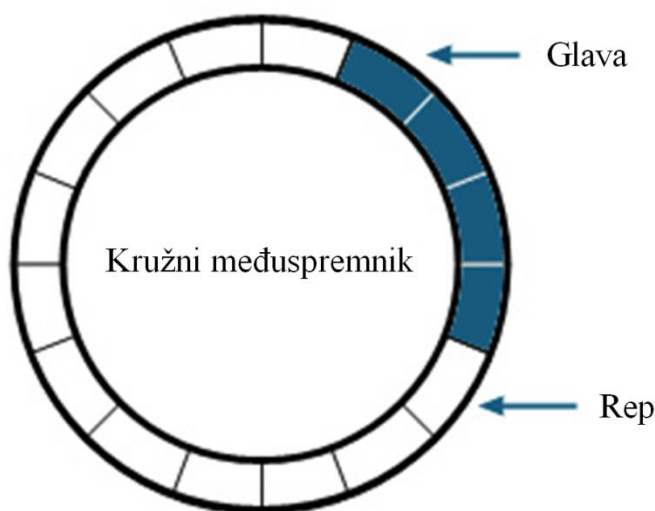
**Slika 4.5.** Promjena točnosti klasifikacije tijekom učenja mreže.



**Slika 4.6.** Promjena gubitka tijekom učenja mreže.

## 4.2.2. Kružni međuspremnik

Kako bi se povećala točnost rješenja korišten je kružni međuspremnik kojem je zadaća izbaciti predikcije vremenskih uvjeta dobivene od konvolucijske neuronske mreže koji se pojavljuju u nelogičnim trenucima. Za primjer možemo uzeti situaciju u kojoj se sporadično pojavljuju krive klasifikacije uzastopnih ulaznih okvira. U tom slučaju možemo zaključiti da se radi o pogreškama jer su vremenski uvjeti sporo promjenjiva veličina. Ove pogreške nastoje se u radu filtrirati pomoću kružnog međuspremnika. Kružni međuspremnik je FIFO (engl. *First In First Out*) struktura podataka koja koristi jednodimenzionalni međuspremnik fiksne veličine čiji je kraj povezan s početkom tako da se, kad dođe do kraja, pokazivač ponovno usmjeri na početak međuspremnika. Rad kružnog međuspremnika opisan je slikom 4.7. gdje glava (engl. *head*) predstavlja pokazivač na element koji je prvi zapisan u međuspremnik, a rep (engl. *tail*) pokazivač na prvi slobodni element spreman za zapisivanje. Obojeni elementi međuspremnika označavaju elemente u kojima je nešto zapisano, dok bijeli predstavljaju slobodne elemente.



Slika 4.7. Prikaz rada kružnog međuspremnika.

Kružni međuspremnik se popunjava izlazima konvolucijske neuronske mreže koji prelaze prag smješten prije ulaza u međuspremnik. Testirano je rješenje s pragovima veličine 0.6, 0.7, 0.8 i 0.9. Prag veličine 0.8 je postizao najbolja rješenja uz prihvatljiv broj filtriranih slika. Nakon odluke prolaska izlaza konvolucijske neuronske mreže kroz prag, izlaz se ili zapisuje u kružni međuspremnik ili odbacuje. Nakon toga slijedi izračunavanje srednje vrijednost svih rezultata spremljenih u međuspremnik, na temelju koje se potom određuje izlaz rješenja.

Testirana su rješenja koja koriste kružne međuspremnik različite veličine uzimajući u obzir i kašnjenje izlaza s obzirom na promjenu vremenskog uvjeta (npr. nailazak na mokr kolnik). Kružni međuspremnik veličine sto dvadeset elemenata se pokazao kao najbolji u filtriranju predikcija vremena koje se pojavljuju u nelogičnim trenucima uz prihvatljivo kašnjenje s obzirom na promjenu vremenskih uvjeta.

## 5. VERIFIKACIJA RADA RJEŠENJA

Nakon odabira strukture i učenja konvolucijske neuronske mreže, predloženo rješenje je testirano na prikupljenim video sekvencama. Prikupljeno je četrdeset video sekvenci, po deset za svaki tip vremenskog uvjeta, a svaka video sekvenca sadrži približno tisuću okvira. Rezultati su prikazani uz pomoć veličina: preciznost (5-1), opoziv (5-2) i  $F1$  (5-3), gdje su  $TP$  (engl. *True Positive*) uzorci koji pripadaju klasi C i koje je klasifikator klasificirao pod klasu C,  $FP$  (engl. *False Positive*) uzorci koji ne pripadaju klasi C, a koje je klasifikator klasificirao pod klasu C,  $TN$  (engl. *True Negative*) uzorci koji ne pripadaju klasi C i koje klasifikator nije klasificirao pod klasu C i  $FN$  (engl. *False Negative*) uzorci koji pripadaju klasi C, a koje klasifikator nije klasificirao pod klasu C.

$$\text{Preciznost} = \frac{TP}{TP + FP} \quad \#(5 - 1)$$

$$\text{Opoziv} = \frac{TP}{TP + FN} \quad \#(5 - 2)$$

$$F1 = 2 * \frac{\text{Opoziv} * \text{Preciznost}}{\text{Opoziv} + \text{Preciznost}} \quad \#(5 - 3)$$

Preciznost je mjera koja daje informaciju o broju točno klasificiranih uzoraka. Opoziv govori koliko je uzoraka, od svih koji pripadaju klasi C, klasificirano kao da joj pripadaju.  $F1$  uzima u obzir i opoziv i preciznost, a računa se kao harmonijska srednja vrijednost opoziva i preciznosti.

### 5.1. Evaluacija izgrađenih konvolucijskih neuronskih mreža

Tablicom 5.1. prikazana je matrica zabune nad skupom za učenje, a tablicom 5.2. rezultati konvolucijske neuronske mreže trenirane nad ulaznim slikama dimenzije 128x72 elemenata, dok su rezultati konvolucijske neuronske mreže trenirane nad ulaznim skupom dimenzija 256x144 elemenata prikazani tablicama 5.3. i 5.4. Dijagonala matrice zabune prikazuje broj pravilno klasificiranog uzoraka ulaznih slika, dok ostali članovi matrice prikazuju broj pogrešno klasificiranih uzoraka. Možemo zaključiti da model treniran nad skupom podataka dimenzija 256x144 elemenata ne daje bolje rezultate iako je rezolucija ulazne slike dvostruko veća. Kako je rješenje namijenjeno ugradbenom računalnom sustavu, bitno je uzeti u obzir da memorijski i računski zahtjevi rješenja budu što manji. Iz navedenih razloga odlučeno je korištenje

konvolucijske neuronske mreže trenirane nad ulaznim slikama dimenzije 128x72 elementa. Nadalje, može se primijetiti kako naučene konvolucijske neuronske mreže najčešće pogrešno klasificiraju, odnosno zamjenjuju uzorke kišovito i čistog vremena.

**Tablica 5.1.** Matrica zabune nad skupom za učenje (128x72).

	Čisto (klasificirano)	Maglovito (klasificirano)	Kišovito (klasificirano)	Snježno (klasificirano)
Čisto (stvarno)	3063	0	21	0
Maglovito (stvarno)	0	2264	2	0
Kišovito (stvarno)	30	0	2968	0
Snježno (stvarno)	1	0	0	3060

**Tablica 5.2.** Rezultati klasifikacije nad skupom za učenje (128x72).

Vremenski uvjet	Preciznost	Opoziv	F1
Čisto	0.9931	0.989	0.991
Maglovito	0.999	1	0.999
Kišovito	0.989	0.992	0.991
Snježno	0.999	1	0.999
<b>Ukupno</b>	<b>0.995</b>	<b>0.995</b>	<b>0.995</b>

**Tablica 5.3.** Matrica zabune nad skupom za učenje (256x144).

	Čisto (klasificirano)	Maglovito (klasificirano)	Kišovito (klasificirano)	Snježno (klasificirano)
Čisto (stvarno)	3075	0	8	1
Maglovito (stvarno)	2	2263	1	0
Kišovito (stvarno)	35	0	2963	0
Snježno (stvarno)	1	1	5	3054

**Tablica 5.4.** Rezultati klasifikacije nad skupom za učenje (256x144).

Vremenski uvjet	Preciznost	Opoziv	F1
Čisto	0.997	0.982	0.989
Maglovito	0.998	0.999	0.998
Kišovito	0.988	0.995	0.991
Snježno	0.997	0.999	0.998
<b>Ukupno</b>	<b>0.995</b>	<b>0.994</b>	<b>0.994</b>

## 5.2. Evaluacija predloženog rješenja za klasifikaciju vremenskih uvjeta

Nakon treniranja klasifikatora rješenje problema detekcije vremenskih uvjeta u okolini automobila testirano je na bazi prikupljenih video sekvenci. Tablicom 4.6. predstavljena je preciznost rješenja nad prikupljenim video sekvencama. Može se zaključiti da se rješenje susreće s problemima detekcije za vrijeme noćne vožnje zbog manjih dinamičkih razlika između slika različitih vremenskih uvjeta. Neke tehnike posvjetljivanja slika po noći poput gama korekcije i izjednačavanja histograma testirane su, ali nisu dale dobre rezultate.



**Tablica 5.5.** Rezultati testiranja predloženog rješenja za različite scene.

Vrijeme dana	Scena	Čisto	Maglovito	Snježno	Kišovito
Dan	Grad	1	1	1	1
	Otvorena cesta	1	1	1	1
	Šuma/Planina	1	1	1	1
Noć	Grad	1	1	0.9732	0.9927
	Otvorena cesta	1	1	1	0.9433
	Šuma/Planina	0.9076	0.8047	1	0.9298

Tablicama 5.6. i 5.8. prikazane su matrice zabune nad okvirima testnih video sekvenci, a rezultati rješenja prikazani su tablicama 5.7. i 5.9. Tablice 5.6. i 5.7. predstavljaju rezultate predloženog rješenja uz isključeni međuspremnik, dok tablice 5.8. i 5.9. predstavljaju rezultate predloženog rješenja. Uvođenjem kružnog međuspremnika značajno se povećala točnost rješenja, za 4.3%. Iz tablica 5.6. i 5.8. je vidljivo da je uvođenje kružnog međuspremnika u potpunosti riješilo problem krivog klasificiranja snježnog vremena pod čisto i kišovito vrijeme pod snježno i maglovito.

Analogno matrici zabune nad skupom za učenje (tablica 5.1.), na matricama zabune nad testnim video sekvencama (tablice 5.6. i 5.8.) vidljivo je da konvolucijska neuronska mreža, kao i predstavljeno rješenje najčešće miješaju uzorke čistog i kišovito vrijeme. Možemo primijetiti da je opoziv uzoraka čistog i kišovito vrijeme najmanji, što znači da je najveći broj krivo klasificiranih uzoraka klasificiran upravo pod te dvije klase. Najmanji opoziv uzoraka čistog i kišovito vrijeme možemo pripisati manjem broju scena sadržanih u bazi podataka za snježno i maglovito vrijeme.

**Tablica 5.6.** Matrica zabune nad testnim video sekvencama predloženog rješenja uz isključeni kružni međuspremnik.

	Čisto (klasificirano)	Maglovito (klasificirano)	Kišovito (klasificirano)	Snježno (klasificirano)
Čisto (stvarno)	9406	51	563	45
Maglovito (stvarno)	79	9572	116	308
Kišovito (stvarno)	513	175	9328	36
Snježno (stvarno)	317	2	200	9539

**Tablica 5.7.** Rezultati klasifikacije nad testnim video sekvencama predloženog rješenja uz isključeni kružni međuspremnik.

Vremenski uvjet	Preciznost	Opoziv	F1
Čisto	0.9345	0.9118	0.923
Maglovito	0.9501	0.9767	0.963
Kišovito	0.928	0.9138	0.9208
Snježno	0.9484	0.9608	0.9545
<b>Ukupno</b>	<b>0.9402</b>	<b>0.9407</b>	<b>0.9403</b>

**Tablica 5.8.** Matrica zabune nad testnim video sekvencama predloženog rješenja.

	Čisto (klasificirano)	Maglovito (klasificirano)	Kišovito (klasificirano)	Snježno (klasificirano)
Čisto (stvarno)	9889	22	133	21
Maglovito (stvarno)	0	9798	0	277
Kišovito (stvarno)	136	94	9818	0
Snježno (stvarno)	0	0	0	10056

**Tablica 5.9.** Rezultati klasifikacije nad testnim video sekvencama predloženog rješenja.

<b>Vremenski uvjet</b>	<b>Preciznost</b>	<b>Opoziv</b>	<b>F1</b>
Čisto	0.9825	0.9864	0.984
Maglovito	0.9725	0.9883	0.98
Kišovito	0.977	0.9866	0.982
Snježno	1	0.9712	0.985
<b>Ukupno</b>	<b>0.983</b>	<b>0.983</b>	<b>0.983</b>

Slikom 4.8. prikazane su neke od krivo klasificiranih slika predloženim rješenjem. Možemo primijetiti da su slike ili lošije kvalitete ili nose slična obilježja pridodanih klasa (primjer: raspršenje vode prolaskom vozila može izgledati kao magla).



Classified: Foggy  
True: Clear



Classified: Snowy  
True: Clear



Classified: Rainy  
True: Foggy



Classified: Snowy  
True: Foggy



Classified: Foggy  
True: Rainy



Classified: Rainy  
True: Snowy

Slika 5.1. Pogrešno klasificirane slike iz skupa za testiranje.

## 6. ZAKLJUČAK

Detekcija vremenskih uvjeta u autonomnoj vožnji od iznimne je važnosti jer vremenski uvjeti direktno utječu na ponašanje rada automobila. Ispravna detekcija vremenskih uvjeta i stanja na cesti omogućuje kreiranje različitih algoritama s ciljem optimizacije rada autonomnog vozila za vrijeme različitih vremenskih uvjeta.

Predloženo rješenje detekcije vremenskih uvjeta u autonomnoj vožnji temelji se na klasifikatoru realiziranom pomoću konvolucijske neuronske mreže s tri konvolucijska sloja trenirane nad slikama dobivenih s kamere montirane na prednjoj strani vozila. Konvolucijska neuronska mreža učena je na podacima dobivenim iz BDD100K baze podataka i izvlačenjem okvira iz vlastito prikupljenih video sekvenci. Izgrađeni model zauzima 1.6 MB memorije, a ulazni podaci prolaze kroz 860359 matematičkih operacija do izlaza. Tijekom klasifikacije uzastopnih video okvira korišten je i kružni međuspremnik od sto dvadeset elemenata koji značajno pospešuje točnost predikcije vremenskih uvjeta filtriranjem predikcija vremena koje se pojavljuju u nelogičnim trenucima.

Dobiveno rješenje postiže točnost od 98.3% nad testnim skupom podataka, gdje najbolje rezultate postiže kod detekcije snježnog vremena (100%), a najlošije kod detekcije maglovitog vremena (97.25%). Predloženo rješenje daje pogrešne predikcije većinom za vrijeme noći, jer dobivene slike, zbog smanjene količine svjetlosti, imaju manje dinamičke razlike za različite vremenske uvjete.

Iako rješenje daje dobre rezultate, daljnje poboljšanje točnosti rješenja moguće je proširivanjem baze podataka za učenje jer unutar baze podataka za snježno i maglovito vrijeme nije sadržano mnogo različitih scena što vjerojatno rezultira i lošijim generalizacijskim svojstvima izgrađene konvolucijske neuronske mreže za navedene vremenske uvjete.

## LITERATURA

- [1] European Commission, eSafety, Final Report of the eSafety Working Group on Road Safety, EU publications, studeni 2003.
- [2] U.S. Department of Transportation, National Highway Traffic Safety Administration, Preliminary Statement of Policy Concerning Automated Vehicles, svibanj 2013.
- [3] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, M. A. Kaafar, The impact of adverse weather conditions on autonomous vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car, IEEE Vehicular Technology Magazine, vol. 14, stranice 103-111, ožujak 2019.
- [4] X. Yan, Y. Luo, X. Zheng, Weather Recognition Based on Images Captured by Vision System in Vehicle, ISSN 2009: Advances in Neural Networks, stranice 390-398, 2009.
- [5] C. Dannheim, C. Icking, M. Mäder, P. Sallis, Weather Detection in Vehicles by Means of Camera and LIDAR Systems, Sixth International Conference on Computational Intelligence, Communication Systems and Networks, svibanj 2015.
- [6] C. Lu, D. Lin, J. Jia, C. K. Tang, Two-Class Weather Classification, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, stranice 2510-2524, prosinac 2016.
- [7] P. Wang, W. Hao, Z. Sun, S. Wang, E. Tan, L. Li, Y. Jin, Regional Detection of Traffic Congestion Using in a Large-Scale Surveillance System via Deep Residual TrafficNet, IEEE Access, vol. 6, stranice 68910 – 68919, studeni 2018.
- [8] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, K. McDonald-Maier, Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks, kolovoz 2018.
- [9] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, W. Stork, Weather Influence and Classification with Automotive Lidar Sensors, lipanj 2019.
- [10] A. K. Jain, J. Mao, K. M. Mohiuddin, Artificial neural networks: a tutorial, Computer, vol. 29, stranice 31-44, ožujak 1996.
- [11] K. Džomba, Konvolucijske neuronske mreže, Zagreb, srpanj 2018.

- [12] R. Hecht-Nielsen, Theory of the Backpropagation Neural Network, Neural Networks for Perception, stranice 65-93, 1992.
- [13] K. Fukushima, Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, Biological Cybernetics, vol. 36, stranice 193-202, travanj 1980.
- [14] L. E. Atlas, T. Homma, R. J. Marks II, An artificial neural network for spatiotemporal bipolar patterns: Application to phoneme classification, Neural Information Processing Systems, 1987.
- [15] Y. LeCun, P. Haffner, L. Bottou, Y. Bengio, Object Recognition with Gradient-Based Learning, Shape, Contour and Grouping in Computer Vision, London, studeni 1999.
- [16] Convolutional Neural Networks for Visual Recognition [Na internetu]. Dostupno na: <http://cs231n.github.io/convolutional-networks/> [17.6.2019.].
- [17] Q. Ke, J. Liu, M. Bennamoun, S. An, F. Sohel, F. Boussaid, Computer Vision for Human-Machine Interaction, Computer Vision for Assistive Healthcare, 2018.
- [18] R. B. Zadeh, B. Ramsundar, TensorFlow for Deep Learning, O'Reilly Media, Inc., ožujak 2018.
- [19] J. Han, C. Moraga, The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning, IWANN '96 Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation, stranice 195-201, lipanj 1995.
- [20] B. Karlik, A. V. Olgac, Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks, International Journal of Artificial Intelligence And Expert Systems, prosinac 2010.
- [21] V. Nair, G. E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, ICML'10 Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Izrael, lipanj 2010.
- [22] N. Cui, Applying Gradient Descent in Convolutional Neural Networks, Journal of Physics: Conference Series, vol. 1004, 2018.

- [23] Dr. K. Koidl, Loss Functions in Classification Tasks, School of Computer Science and Statistic Trinity College, Dublin.
- [24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15, lipanj 2014.
- [25] TensorFlow [Na internetu]. Dostupno na: <https://www.tensorflow.org/> [17.06.2019.].
- [26] Keras Documentation [Na internetu]. Dostupno na: <https://keras.io/> [17.06.2019.].
- [27] Y. Fisher, naslov: BDD100K: A Large-scale Diverse Driving Video Database [Na internetu], Berkeley Artificial Intelligence Research, BAIR, 30.05.2018., dostupno na: <https://bair.berkeley.edu/blog/2018/05/30/bdd/> [17.06.2019.].



## SAŽETAK

Problematika kojom se bavi diplomski rad je detekcija vremenskih uvjeta u okolini automobila. U prvom dijelu diplomskog rada opisan je problem detekcije vremenskih uvjeta, kao i neki dosadašnji radovi koji se bave sličnim problemima. Drugim dijelom rada predstavljena je teorijska osnova potrebna za rješavanje praktičnog dijela rada. Rješenje praktičnog dijela rada, kao i rezultati testiranja rješenja opisani su u trećem dijelu diplomskog rada.

Za potrebe treniranja klasifikatora izrađeni su skupovi podataka za treniranje i validaciju iz dva izvora: BDD100K baze podataka i izvlačenjem okvira iz prikupljenih video sekvenci. Kreirane su dvije baze podataka: WDC\_dataset i DNC\_dataset baza podataka, gdje je WDC\_dataset korištena pri učenju klasifikatora koji bi trebao raspoznavati vremenske uvjete, a DNC\_dataset pri učenju klasifikatora koji bi trebao raspoznavati vrijeme dana, odnosno dan i noć.

Predloženo rješenje je realizirano korištenjem konvolucijske neuronske mreže s tri konvolucijska sloja između kojih su korištene ReLu aktivacijske funkcije i slojevi sažimanja po maksimalnoj vrijednosti. Rješenje bazirano na konvolucijskoj neuronskoj mreži je uz korištenje empirijski određenog praga i kružnog međuspremnik za dodatno filtriranje loših predikcija pokazalo zadovoljavajuće rezultate nad testnim video sekvencama.

Dobiveno rješenje postiže točnost od 98.3% nad testnim video sekvencama, gdje najbolje rezultate postiže kod detekcije snježnog vremena (100%), a najlošije kod detekcije maglovitog vremena (97.25%).

**Ključne riječi:** autonomna vožnja, konvolucijska neuronska mreža, kružni međuspremnik, prag, vremenski uvjeti

# ABSTRACT

## **Weather detection in the environment of a vehicle**

This diploma thesis details the detection of weather conditions in the car environment. The first part of the thesis describes the problem of weather detection, as well as some previous work dealing with same or similar problems. The second part of the paper presents the theoretical basis for dealing with the practical part of the paper. The solution of the practical part of the thesis, as well as the results of testing the solution are described in the third part of the thesis.

For training purposes, training and validation data sets were created from two main sources: the BDD100K database and by extracting frames from the collected video sequences. Two databases were created: WDC\_dataset database and DNC\_dataset database, where WDC\_dataset was used for weather conditions classifier training, and DNC\_dataset for day-night classifier training.

The proposed solution was implemented using convolutional neural network with three convolutional layers, between which ReLu activation functions and max pooling layers were used. The solution based on the convolutional neural network is using a empirically determined threshold and a circular buffer to further filter out false predictions.

The proposed solution achieved an accuracy of 98.3% for tested video sequences, where the solution is showing the best results in snowy weather detection (100%) and worst in foggy weather detection (97.25%).

**Key words:** autonomous driving, convolutional neural network, circular buffer, threshold, weather conditions

## ŽIVOTOPIS

Jakob Triva rođen je 20.03.1996. godine u Osijeku. Osnovno obrazovanje započinje upisom u čepinsku osnovnu školu Vladimira Nazora 2002. godine. U drugom razredu osnovne škole upisuje i osnovnu glazbenu školu Franje Kuhača u Osijeku, smjer violina, koju pohađa narednih šest godina. U osmom razredu osnovne škole upisuje srednju glazbenu školu Franje Kuhača, smjer violina, koju pohađa naredne četiri godine. Završetkom osnovne škole 2010. godine nastavlja obrazovanje upisom u III. gimnaziju Osijek. 2014. godine završava srednju školu i upisuje se na preddiplomski studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

2017. godine stječe akademski naziv sveučilišnog prvostupnika (baccalaureus) inženjera računarstva. Iste godine upisuje diplomski studij računarstva, izborni blok Robotika i umjetna inteligencija. Od kolovoza 2017. do prosinca 2017. pohađa studentsku praksu u firmi Bamboo Lab d.o.o. te nastavlja suradnju do srpnja 2018. godine. 2018. godine sudjeluje na STEM games natjecanju, na kojemu sa svojim timom osvaja sedmo mjesto. 2019. godine povodom obilježavanja 41. godišnjice Fakulteta elektrotehnike, računarstva i informacijskih tehnologija dobiva priznanje za postignut uspjeh o studiranju. Od listopada 2018. do srpnja 2019. stipendist je instituta RT-RK iz Osijeka gdje pohađa studentsku praksu i odrađuje diplomski rad. Dobitnik je stipendije za deficitarna zanimanja Nacionalne zaklade za potporu učeničkom i studentskom standardu za 2015./2016., 2016./2017., 2017./2018. i 2018./2019. akademsku godinu.

Potpis:

---