Univerzitet u Sarajevu

Elektrotehnički fakultet

Odsjek za računarstvo i informatiku

**Prepoznavanje i klasifikacija saobraćajnih znakova**

Predmet: Vještačka inteligencija

**Profesor:** **Studenti:**

V. prof. dr Amila Akagić, dipl.ing.el. Nekić Elma, 19383

Alihodžić Adna, 19450

Pešković Dinela, 19359

**SADRŽAJ**

[**1. Opis problema 1**](#_kj00tcogory2)

[1.1. Postojeći datasetovi 1](#_trdvrejc07gu)

[**2. Trenutno stanje u oblasti problema 2**](#_v6vq0mifoqp1)

[2.1. TSCLIP: Robust CLIP Fine-Tuning for Worldwide Cross-Regional Traffic Sign Recognition 3](#_p6qzr0sycp2k)

[2.2. Strategija “razmisli dvaput prije prepoznavanja”: Veliki Multimodalni Modeli (LMM) za opće fino prepoznavanje saobraćajnih znakova 4](#_lokq7pew8ppq)

[2.3. Prepoznavanje saobraćajnih znakova korištenjem lokalnog vida Transformatora 6](#_mtx7ldxmksi6)

[**3. Izbor, analiza i pretprocesiranje dataset-a 8**](#_fgctmie42d5d)

[**4. Odabir, formiranje, treniranje i testiranje modela 13**](#_c8hwpgvml970)

[4.1. Odabrana metoda: 13](#_6xix9uc790mh)

[4.2. Korištene tehnologije: 13](#_hmhak1bofnql)

[4.3. Priprema podataka: 13](#_bfrga3xksri0)

[4.4. Treniranje modela: 14](#_j0bxvcn01xlo)

[4.5. Testiranje modela: 15](#_5fxawkprohxw)

[4.6. Rezultati 16](#_oy1hhtr42j4c)

[4.7. Vizualizacija rješenja: 16](#_9cd9lp79yy2u)

[**5. Cjelokupni osvrt na problem i dobijeno rješenje 18**](#_fpyvopwkwgo6)

[**6. Literatura 19**](#_h2tufjd33aba)

# Opis problema

Glavni cilj projekta je kreirati model koji će sa slike ili kamere prepoznati saobraćajni znak. Pored toga je potrebno implementirati aplikaciju koja će koristiti model, te za svaki prepoznati znak prikazati i njegovo značenje.

U zadnjoj deceniji svjedočimo naglom tehnološkom napretku i povećanoj automatizaciji u automobilskoj industriji. Jedan od glavnih fokusa ovog razvoja je povećanje sigurnosti u saobraćaju kroz gradnju sigurnosnih sistema u vozilima. Unapređenje inteligentnih transportnih sistema sve više ističe ulogu vještačke inteligencije u sigurnosti vozila. **Advanced Driving Assistance Systems (ADAS)** predstavlja širok spektar tehnologija korištenih da pruže pomoć vozaču, poboljšaju sigurnost u saobraćaju i cjelokupno iskustvo vožnje. ADAS koristi kombinaciju senzora, kamera i radara koja radi sa podacima u realnom vremenu, prepoznaje potencijalne rizike, te nastoji spriječiti nesreće. Ova vrsta sistema radi tiho u pozadini, što znači da ne ometa vozača tokom vožnje, a istovremeno pruža podršku pri vožnji.[[1]](#footnote-0) ADAS je skupina velikog broja različitih tehnologija, a naš projekat je fokusiran na prepoznavanje saobraćajnih znakova ili Traffic-sign recognition (TSR).

Saobraćajni znakovi sa svojim standardiziranim i uočljivim tekstom i simbolima su od samog početka bili primarni fokus tehnologije prepoznavanja slike u oblasti saobraćaja. **Traffic sign recognition** je relativno nova tehnologija čija je primarna svrha povećati fokus vozača obavještavanjem o saobraćajnim znakovima. Glavni zadaci TSR-a su detektovati i prepoznati saobraćajne znakove koje je vozač možda propustio, te ga upozoriti na njihovo prisustvo. Ova tehnologija koristi napredne kamere pričvršćene na vjetrobransko staklo koje skeniraju stranu puta u odnosu na vozilo. Kada kamera detektuje znak, sistemski softver obrađuje sliku da bi odredio njenu klasifikaciju i značenje. Ova informacija se zatim prenosi vozaču prikazom na instrument tabli. TSR je nekada implementiran i kao dio sistema za kontrolu brzine vozila, pri čemu se nakon detekcije znaka o ograničenju brzine provjerava trenutna brzina vozila, pa se automatski prilagođava ograničenju ukoliko je to potrebno. Iako je ova oblast od svog nastanka dostigla značajan napredak kroz mnoge razvijene algoritme i modele za prepoznavanje saobraćajnih znakova, još uvijek postoji izazov kako ostvariti što veću tačnost i brzinu u prepoznavanju.

## 1.1. Postojeći datasetovi

Postoji nekoliko datasetova u oblasti prepoznavanja saobraćajnih znakova koji se koriste pri procjeni performansi algoritama za prepoznavanje. Datasetovi trebaju sadržati raznovrsnu kolekciju scenarija iz stvarnog svijeta, sa različitim kategorijama saobraćajnih znakova. Na taj način se algoritmi mogu trenirati i testirati u realnim okruženjima.

**Njemački dataset saobraćajnih znakova - German Traffic Sign Dataset i German Traffic Sign Recognition Benchmark (GTSDB i GTSRB)**

Njemački dataset sadrži kolekciju od 51 839 znakova, organizovanih u 43 kategorije. Ključna činjenica je da se ovaj dataset primarno sastoji od njemačkih saobraćajnih znakova, što nije najreprezentativnija skupina za predstavljanje raznolikosti znakova u drugim regijama. To može ograničiti generalizaciju algoritama, jer negativno utiče na preformanse kada se primijeni na druge regije. Čak i pored tog nedostatka, jako je popularan zbog količine podataka i predstavljanja scenarija iz realnog svijeta.

**Belgijski skup podataka saobraćajnih znakova - Belgium Traffic Sign Dataset (BTSD)**

Sastoji se od 7095 slika visoke rezolucije, podijeljenih u 62 jedinstvene kategorije saobraćajnih znakova iz Belgije i Nizozemske. Iako je manji u odnosu na druge skupove podataka, kvalitet i raznovrsnost su ono što ga čini popularnim, pogotovo pri validaciji algoritama treniranih na većim skupinama poput GTSRB.

**Kineska baza podataka o saobraćajnim znakovima - Chinese Traffic Sign Database (TSRD)**

Ova baza podataka predstavlja kolekciju od 6164 slike podijeljene u 58 jedinstvenih klasa, prikupljenih iz različitih izvora poput BAIDU Street View, kao i drugih kamera iz realnog svijeta. Različitosti izvora podataka pružaju mogućnost treniranja modela na podacima nastalim u različitim vremenskim uvjetima, osvjetljenjima i vidljivošću. Također sadrži značajnu skupinu djelimično skrivenih znakova, što pruža dodatnu priliku za usavršavanje modela i njegovo treniranje u kompleksnijim uslovima.[[2]](#footnote-1)

# Trenutno stanje u oblasti problema

Trenutna istraživanja u oblasti prepoznavanja saobraćajnih znakova se generalno mogu podijeliti u dvije kategorije: metode bazirane na feature-based mašinskom učenju (feature-based machine learning) i deep learning metode za automatsku ekstrakciju osobina.

Machine learning metode podrazumijevaju tradicionalne metode, poput Support Vector Machines (SVM), k-Nearest Neighbor i stabla odluke. Ovi algoritmi su bazirani na principu terniranja modela na skupu podataka i korištenju tog modela za predviđanja na novom skupu podataka. Machine learning je popularan u oblasti prepoznavanja saobraćajnih znakova zbog svoje jednostavnosti i efikasnosti.

Deep learning algoritmi koriste neuronske mreže za modeliranje kompleksnih veza između ulaza i izlaza. Postali su popularni u prepoznavanju saobraćajnih znakova zbog sposobnosti da automatski uče i izdvajaju kompleksne osobine iz skupa podataka, što dovodi do veće preciznosti i eliminiše potrebu za ručnim izdvajanjem osobina. Najčešće korišteni deep learning algoritmi su Convolutional Neural Networks (CNN) i Recurrent Neural Networks (RNN). U nastavku slijedi pregled nekih najnovijih postignuća iz oblasti prepoznavanja saobraćajnih znakova.

## 2.1. TSCLIP: Robust CLIP Fine-Tuning for Worldwide Cross-Regional Traffic Sign Recognition

Posljednjih godina su konvolucione neuronske mreže (CNN) postigle automatsko izdvajanje karakteristika i učenje u visokodimenzionalnim prostorima, što je značajno smanjilo poteškoće dizajna karakteristika i poboljšalo performanse prepoznavanja. Korištenjem metoda dubokog učenja, postignuti su rezultati visoke tačnosti u oblasti prepoznavanja saobraćajnih znakova, što je nadmašilo tradicionalne feature based metode. Međutim, CNN-ovi se obučavaju samo na svojim odgovarajućim skupovima podataka, pa se zbog toga performanse značajno pogoršavaju prilikom testiranja na skupovima podataka iz različitih okruženja ili regiona, čak i za iste kategorije. Ovaj problem ima mali uticaj u oblasti opšte klasifikacije, ali je značajan u slučaju zadataka koji se oslanjaju na semantičke informacije znakova za razumijevanje okoline.

Ovi problemi su riješeni uvođenjem Contrastive Language-Image Pre-training (CLIP). CLIP je multimodalni model koji ima sposobnost da direktno razumije odnos između slika i opisa prirodnog jezika koji su povezani s njima.[[3]](#footnote-2) Multimodalni modeli predstavljaju vrstu modela mašinskog učenja koji može procesirati i integrisati informacije iz više modaliteta/tipova podataka. Za razliku od tradicionalnih modela koji rade sa samo jednom vrstom podataka (npr. samo tekst ili samo slika), multimodalni modeli mogu istovremeno učiti iz različitih tipova podataka.

CLIP koristi dvije vrste enkodera, jedan za slike, a drugi za tekst. Ovi enkoderi pretvaraju odgovarajuće podatke u vektore karakteristika fiksne dimenzije. Za treniranje modela kako bi povezivao parove slika-tekst, koristi se kontrastno učenje. Cilj je približiti vektore karakteristika podudarnih parova slika-tekst u prostoru za ugrađivanje, dok se vektori nepodudarnih parova udaljavaju. To se postiže kontrastnom funkcijom gubitka. Za svaki par slika-tekst u grupi za obuku, CLIP računa kosinusnu sličnost između svih mogućih vektora karakteristika slike i teksta. Model maksimizira sličnost za ispravne parove, a minimizira za neispravne parove. Ovaj proces podučava model da efektivno uskladi slike sa njihovim odgovarajućim tekstualnim opisima. Na ovaj način CLIP postaje zero-shot klasifikator, tj. klasifikator koji ima sposobnost da prepozna i kategorizira objekte ili koncepte bez potrebe da prethodno vidi primjere tih objekata ili koncepata. Nad općim modelom CLIP-a se primjenjuju tehnike fine-tuninga koji će ga specijalizirati za određene tipove zadataka. U ovom radu se taj cilj postiže kreiranjem TSCLIP modela.

Tri glavna koraka u kreiranju TSCLIP-a su: kreiranje raznolikog dataseta, korištenje prompt engineeringa (inženjering upita) i Adaptive Dynamic Weight Ensembling (ADWE) tehnike. Korišten je Cross-regional traffic sign (CRTS) dataset koji se sastoji iz 46 kategorija saobraćajnih znakova izdvojenih iz 10 različitih datasetova. Time je obezbijeđena raznolikost u oblicima, bojama i jezicima znakova. Inženjering upita podrazumijeva shemu koja obuhvata opise scena saobraćajnih znakova u realnim okruženjima, kao i opise različitih kategorija i njihova odgovarajuća saobraćajna pravila. Ovo je prva metoda prepoznavanja saobraćajnih znakova koja pruža sveobuhvatan opis, što značajno poboljšava tačnost prepoznavanja i generalizaciju u različitim scenarijima vožnje. Tokom fine-tuninga TSCLIP modela koristi se ADWE tehnika koja uzima težine modela koji se trenira kroz fine-tuning i kombinuje ih sa težinama originalnog zero-shot CLIP modela. Glavni cilj ove tehnike je sačuvati sposobnost generalizacije koju ima početni CLIP model, ali da istovremeno stiče nova znanja koje sadrži više specifikacija vezanih za željenu oblast, u ovom slučaju prepoznavanje saobraćajnih znakova. U opsežnim međuregionalnim testovima, TSCLIP model je značajno nadmašio uobičajene metode i postigao je najbolje rezultate u poređenju sa svim dosadašnjim fine-tuning tehnikama na datom zadatku.

## 2.2. Strategija “razmisli dvaput prije prepoznavanja”: Veliki Multimodalni Modeli (LMM) za opće fino prepoznavanje saobraćajnih znakova

Ranije metode TSR tehnologije koristile su ručno rađene karakteristike (kao što su HOG i SIFT), dok savremeni pristupi koriste neuronske mreže (CNN) i vizualne transformere. Ove metode zahtijevaju velike količine treniranih podataka što dovodi do problema da ne funkcionišu dobro na znakovima iz drugih zemalja zbog vizuelnih razlika ili znakovima sa oštećenjima nastalim usljed saobraćajnih nesreća ili vremenskih neprilika. Veliki multimodalni modeli (LMM) i jezički modeli (LLM) nude novo rješenje zahvaljujući svojoj sposobnosti da obrađuju i tekstualne i vizualne podatke. LMM modeli pružaju pomoć vozačima tako da prepoznaju znakove, shvate kontekst iz okoline i reaguju u realnom vremenu. Najveći problemi LMM modelima su veličina znakova, jer su često mali i jedva vidljivi, te sličnost između znakova. Zbog navedenih razloga LMM modeli, bez određene pomoći, mogu pogriješiti. Stoga je potrebno razviti specifične strategije da se iskoristi njihov potencijal. Strategija “razmisli dvaput prije prepoznavanja”, “tjera” LMM sisteme da razmišljaju korak po korak prije prepoznavanja.

Glavni doprinosi ove strategije su:

* LMM sistemi ne donese brze i pogrešne odluke već da idu kroz detaljan i složen proces prepoznavanja znakova
* nije nam potrebno treniranje podataka jer je strategija bazirana na već postojećim podacima i generisanim opisima
* može se koristiti za znakove iz različitih zemalja

Ova strategija je testirana na 5 skupova iz različitih zemalja podataka gdje se pokazala veoma uspješnom.

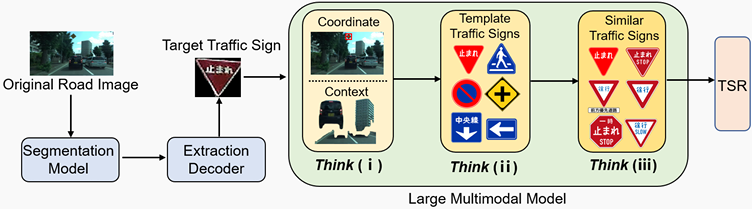
Prije nego dođemo do upotrebe strategije “razmisli dvaput prije prepoznavanja” treba da prođemo kroz faze segmentacije i ekstrakcije saobraćajnih znakova. Segmentacija je proces gdje se orginalna slika ceste unosi u model za segmentaciju. Taj model označava sve objekte na slici različitim bojama, te od tih objekata izdvaja saobraćajne znakove tako što se generiše maska slike - u kojoj je znak izolovan od pozadine i drugih objekata. Zatim prelazimo u drugu fazu, fazu ekstakcije gdje se pomoću dizajniranog detektora za ekstrakciju, iz maske automatski dobijaju koordinate znaka, koji se izrezuje iz orginalne slike. Time se eliminišu pozadinski šumovi i dobija se čista slika znaka spremna za prepoznavanje. Prepoznavanje je dio gdje dolazi do upotrebe strategije prolazeći kroz faze generisanja prethodnog znanja i zaključivanja kroz više koraka. U fazi generisanja prethodnog znanja imamo tri podfaze:

Prva podfaza su kontekstualni opisi. Ova podfaza podrazumijeva opisivanje okoline sa znakom, čime se mogu eliminisati određeni znakovi koji se po pravilu ne nalaze u takvom okruženju. Također se koriste i središnje koordinate koje pomažu da se pronađe određeni znak ukoliko ih je više na tom mjestu.

Druga podfaza su karakteristični opisi. U ovoj podfazi se uvodi učenje s nekoliko primjera gdje model vidi šablonske primjere za svaki znak. Primjeri se opisuju ključnim karakteristikama kao što su: oblik, boja i sadržaj znaka.

U treću podfazu spadaju diferencijalni opisi koji naglašavaju razlike između sličnih znakova. Ova podfaza dodatno poboljšava preciznost LMM-a.

Druga faza strategije je zaključivanje kroz više koraka. LMM koristi sve prethodno generisane opise kako bi u više koraka analizirao ciljnu sliku znaka. Model prvo razumije kontekst, zatim poredi znak sa poznatim šablonima, i na kraju analizira razlike među sličnim znacima. Ovaj pristup značajno poboljšava preciznost, posebno u slučajevima kada su znakovi teško prepoznatljivi ili iz stranih domena.[[4]](#footnote-3)



**Slika 1. Faze strategije “razmisli dvaput prije prepoznavanja”[[5]](#footnote-4)**

## 2.3. Prepoznavanje saobraćajnih znakova korištenjem lokalnog vida Transformatora

Prepoznavanje saobraćajnih znakova je ključna komponenta autonomne vožnje i sistema asistencije vozaču. Zadatak prepoznavanja znakova postaje sve značajniji u oblasti računalnog vida. CNN mreže se često koriste u tim zadacima, ali uvođenjem vision transformer modela omogućeno je efikasnije učenje globalnih karakteristika. U ovom radu predlažemo novi model koji spaja prednosti konvolucijskih i transformer mreža. Model uključuje konvolucione blokove za učenje lokalnih veza te transformer blokove za učenje globalnih odnosa. Uveden je i modul za lokalnost kako bi se dodatno pojačala lokalna percepcija.

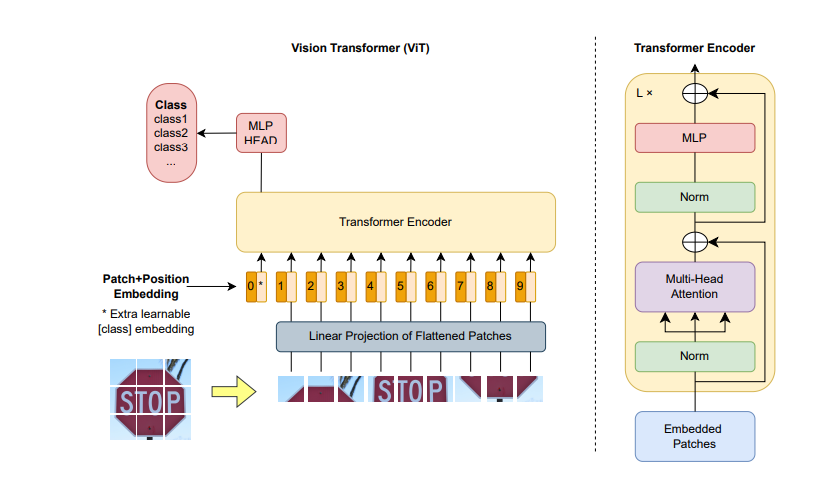
Sigurnosni sistemi u autonomnim vozilima i sistemima pomoći vozaču postali su ključni, a s tim i zadatak prepoznavanja saobraćajnih znakova (TSR) dobija sve veću pažnju. Danas u tim zadacima dominiraju metode dubokog učenja, posebno konvolucijske neuronske mreže (CNN), poznate po učenju lokalnih karakteristika slike.

Nakon uspjeha CNN-ova, uvedeni su transformeri – prvo u obradi jezika, a zatim prilagođeni i za računarski vid kroz Vision Transformere (ViT). Oni su bolji u učenju globalnih odnosa, ali im nedostaje lokalna percepcija. Kako bi se spojile prednosti oba pristupa, razvijeni su hibridni modeli koji kombinuju konvolucione i transformer komponente.

**Metodologija:**

**A. Vision Transformer – pregled**

Transformer se sastoji od enkodera (koristi se za klasifikaciju slika). Slike se dijele na manje „patch“-eve, pretvaraju se u niz tokena, a zatim prolaze kroz self-attention mehanizam i MLP blokove.

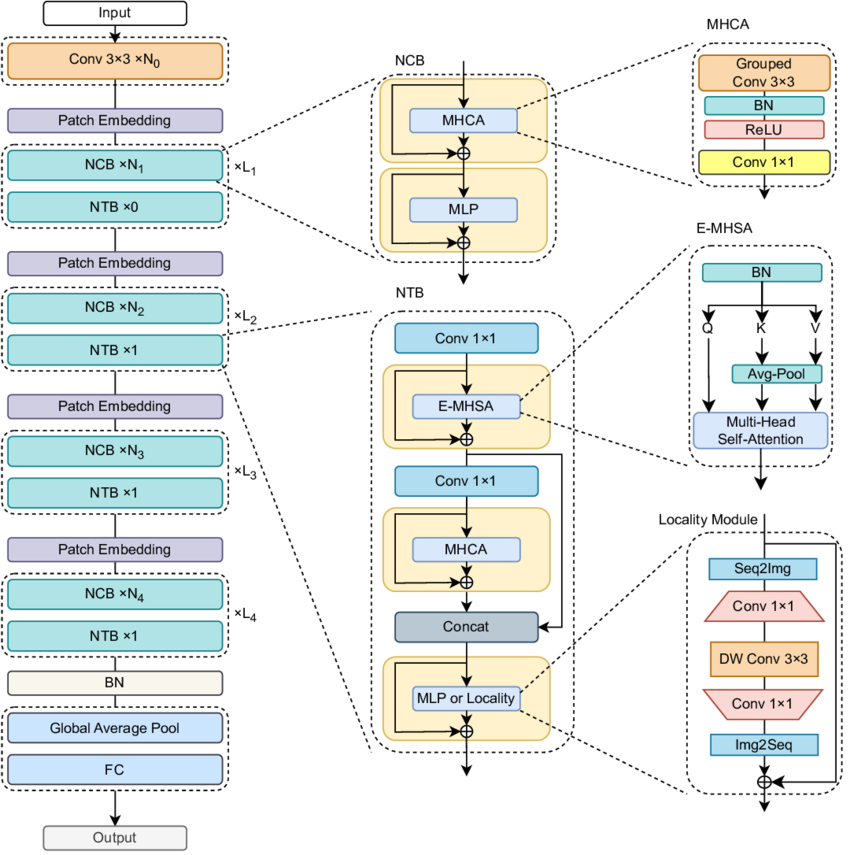


**Slika 3 : Mrežna arhitektura osnovnog ViT-a.[[6]](#footnote-5)**

**B. Nova generacija lokalnog vision transformera**

Hibridna arhitekturu koja koristi:

* **NCB (Next Convolution Block)**: Kombinacija konvolucijske pažnje (MHCA) i MLP-a.
* **NTB (Next Transformer Block)**: Koristi efikasnu multi-head self-attention (E-MHSA) i MHCA, uz lokalni feedforward modul (LFF) za pojačanje lokalne percepcije.
* **NHS (Next Hybrid Strategy)**: Naizmjenično korištenje NCB i NTB blokova za kombinaciju lokalnih i globalnih karakteristika u svim fazama mreže.[[7]](#footnote-6)



**Slika 4: Arhitektura predloženog lokalnog vision transformera[[8]](#footnote-7)**

# Izbor, analiza i pretprocesiranje dataset-a

Dataset koji će biti korišten za rješavanje postavljenog problema je Traffic Sign Preprocessed dataset, autora Valentyna Sichkara. Dataset je dostupan za besplatno preuzimanje uz registraciju na Kaggle.com platformi na sljedećem linku:

<https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-preprocessed/data?select=datasets_preparing.py>.

Nakon registracije se dataset može preuzeti klikom na opciju “Download”, pri čemu se preuzima .zip fajl. Dataset je organizovan u više .pickle fajlova, te su podaci podijeljeni u grupe za treniranje, validaciju i testiranje. Pored toga postoje i dodatni fajlovi (data0.pickle, data1.pickle….) koji se pomoću datasets\_prepraing.py mogu iskoristiti tako da se postojeći dataset proširi. Za analizu dataset-a korištena je sljedeća skripta:

import pickle

import numpy as np

import matplotlib.pyplot as plt

from collections import Counter

# Funkcija za učitavanje podataka

def load\_pickle(file\_path):

with open(file\_path, 'rb') as f:

return pickle.load(f)

# Učitavanje train/valid/test

train\_data = load\_pickle('train.pickle')

valid\_data = load\_pickle('valid.pickle')

test\_data = load\_pickle('test.pickle')

X\_train, y\_train = train\_data['features'], train\_data['labels']

X\_valid, y\_valid = valid\_data['features'], valid\_data['labels']

X\_test, y\_test = test\_data['features'], test\_data['labels']

# Broj instanci

print("Train:", X\_train.shape[0])

print("Valid:", X\_valid.shape[0])

print("Test:", X\_test.shape[0])

print("Ukupno:", X\_train.shape[0] + X\_valid.shape[0] + X\_test.shape[0])

print("Dimenzije slika:", X\_train.shape[1:])

broj\_klasa = len(set(y\_train))

print("Broj klasa:", broj\_klasa)

train\_counts = Counter(y\_train)

print("Broj instanci po klasama:", train\_counts)

print(X\_train.min(), X\_train.max())

Saznajemo da postoji 34799 instanci za treniranje, 4410 instanci za validaciju, te 12630 instanci za testiranje, što ukupno čini 51839 instanci. Svaka instanca je slika veličine 32x32 piksela u 3 kanala (RGB). Dataset sadrži 43 različite klase saobraćajnih znakova, a raspored instanci po svakoj klasi je sljedeći:

**Klasa 0 (Speed limit (20km/h)):** 180 instanci

**Klasa 1 (Speed limit (30km/h)):** 1980 instanci

**Klasa 2 (Speed limit (50km/h)):** 2010 instanci

**Klasa 3 (Speed limit (60km/h)):** 1260 instanci

**Klasa 4 (Speed limit (70km/h)):** 1770 instanci

**Klasa 5 (Speed limit (80km/h)):** 1650 instanci

**Klasa 6 (End of speed limit (80km/h)):** 360 instanci

**Klasa 7 (Speed limit (100km/h)):** 1290 instanci

**Klasa 8 (Speed limit (120km/h)):** 1260 instanci

**Klasa 9 (No passing):** 1320 instanci

**Klasa 10 (No passing for vehicles over 3.5 metric tons):** 1800 instanci

**Klasa 11 (Right-of-way at the next intersection):** 1170 instanci

**Klasa 12 (Priority road):** 1890 instanci

**Klasa 13 (Yield):** 1920 instanci

**Klasa 14 (Stop):** 690 instanci

**Klasa 15 (No vehicles):** 540 instanci

**Klasa 16 (Vehicles over 3.5 metric tons prohibited):** 360 instanci

**Klasa 17 (No entry):** 990 instanci

**Klasa 18 (General caution):** 1080 instanci

**Klasa 19 (Dangerous curve to the left):** 180 instanci

**Klasa 20 (Dangerous curve to the right):** 300 instanci

**Klasa 21 (Double curve):** 270 instanci

**Klasa 22 (Bumpy road):** 330 instanci

**Klasa 23 (Slippery road):** 450 instanci

**Klasa 24 (Road narrows on the right):** 240 instanci

**Klasa 25 (Road work):** 1350 instanci

**Klasa 26 (Traffic signals):** 540 instanci

**Klasa 27 (Pedestrians):** 210 instanci

**Klasa 28 (Children crossing):** 480 instanci

**Klasa 29 (Bicycles crossing):** 240 instanci

**Klasa 30 (Beware of ice/snow):** 390 instanci

**Klasa 31 (Wild animals crossing):** 690 instanci

**Klasa 32 (End of all speed and passing limits):** 210 instanci

**Klasa 33 (Turn right ahead):** 599 instanci

**Klasa 34 (Turn left ahead):** 360 instanci

**Klasa 35 (Ahead only):** 1080 instanci

**Klasa 36 (Go straight or right):** 330 instanci

**Klasa 37 (Go straight or left):** 180 instanci

**Klasa 38 (Keep right):** 1860 instanci

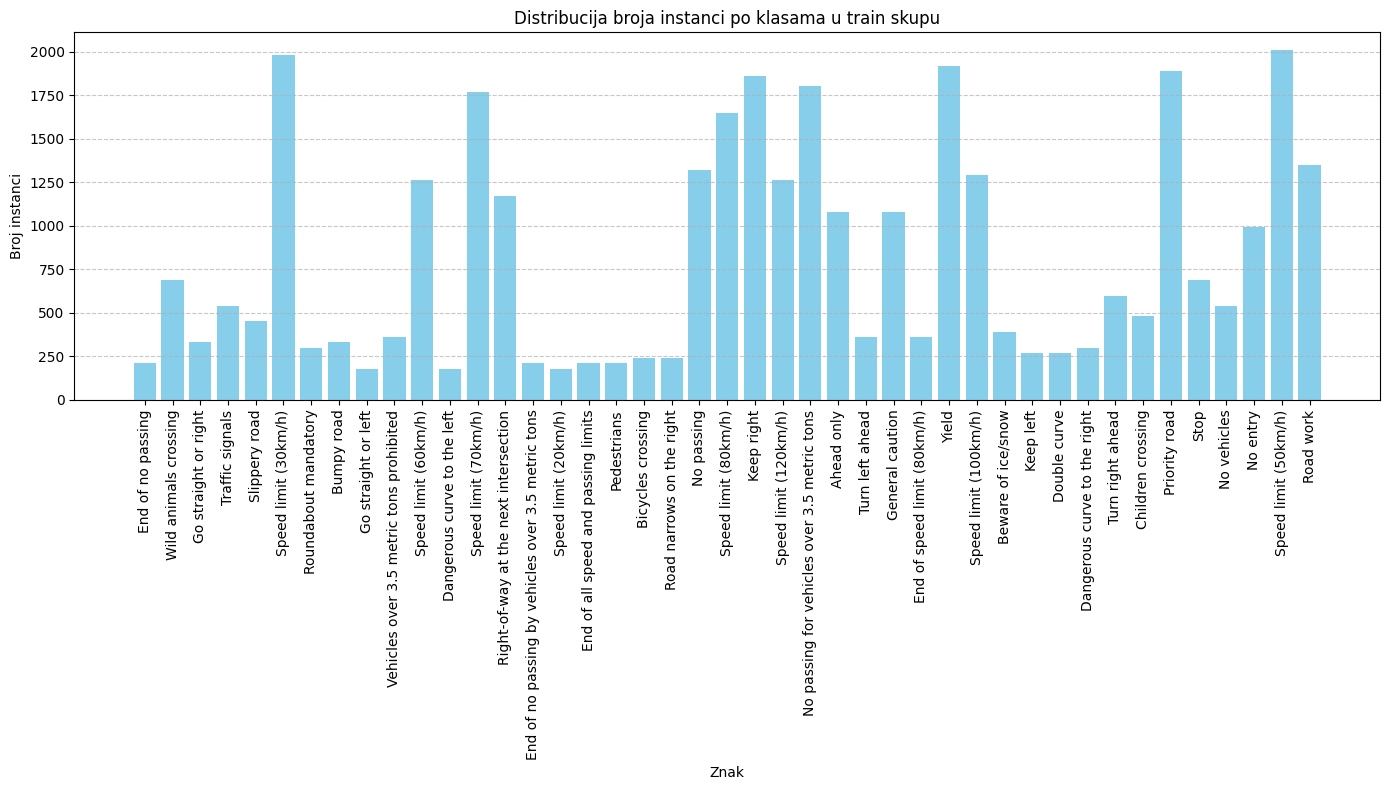
**Klasa 39 (Keep left):** 270 instanci

**Klasa 40 (Roundabout mandatory):** 300 instanci

**Klasa 41 (End of no passing):** 210 instanci

**Klasa 42 (End of no passing by vehicles over 3.5 metric tons):** 210 instanci.

Distribuciju instanci po klasama u skupu za treniranje jednostavno možemo predstaviti preko sljedećeg histograma:



**Graf 1: Prikaz distribucije instanci po klasama**

Ono što primjećujemo je da distribucija broja instanci po klasi varira, sa nekim klasama koje imaju značajno više slika od drugih. Dataset zauzima približno 4GB memorije. Nad podacima su već izvršene neke metode pretprocesiranja, što nam govori i samo ime dataset-a. Prvenstveno, sve slike su skalirane na 32x32 piksela, što osigurava konzistentnost u input modela. Svaka slika je označena odgovarajućom klasom (labelom). Pored toga, podaci su već podijeljeni u grupe za treniranje, validaciju i testiranje. Izvršena je i provjera vrijednosti piksela, koje obuhvataju raspon od 0 do 255, što znači da bi u pretprocesiranju trebalo uključiti normalizaciju ovih vrijednosti, kako bismo povećali efikasnost i stabilnost učenja neuronske mreže.

Prvi rizik koji smo identifikovali je imbalans klasa, jer se u nekim klasama javlja nekoliko hiljada, a drugim svega par stotina slika, pa može doći do pristrasnosti modela prema češće zastupljenim klasama, a da lošije prepoznaje znakove iz manjih klasa. Probleme može izazvati i mala veličina slika, jer zbog toga neki detalji mogu biti izgubljeni, što se može negativno odraziti na performanse modela. Dodatno predstavlja problem i sličnost određenih kategorija znakova koje se razlikuju po sitnim detaljima. Jedan od primjera za to je grupa klasa “Speed limit” koje se razlikuju jedino po ograničenju brzine, tačnije broju koji je prikazan na tabli. U slučaju da broj na slici nije dovoljno jasan, može doći do pogrešnog prepoznavanja znaka.

# Odabir, formiranje, treniranje i testiranje modela

## 4.1. Odabrana metoda:

Metoda koja je izabrana za rješavanje ovog problema je korištenje konvolucione neuronske mreže (CNN) za klasifikaciju slika. CNN je dobro poznata metoda za obradu i klasifikaciju slika zbog svoje sposobnosti da automatski uči karakteristike sa slika.

## 4.2. Korištene tehnologije:

Za implementaciju rješenja koristile smo programski jezik Python, koji je danas jedan od najčešće korištenih jezika u oblasti mašinskog učenja zbog svoje jednostavnosti i bogate podrške za razne biblioteke. Za definisanje, treniranje i evaluaciju neuronske mreže korišten je TensorFlow zajedno sa API ekstenzijom Keras, koji omogućava jednostavno kreiranje i obučavanje modela dubokog učenja. Biblioteke NumPy i Pandas smo koristile za manipulaciju i analizu podataka, dok nam je Matplotlib služio za vizualizaciju rezultata treniranja i evaluacije modela. Podaci su učitavani pomoću modula Pickle, a podaci o klasama učitani su iz CSV fajla pomoću Pandasa. Projekat je realizovan u okruženju Google Colab.

## 4.3. Priprema podataka:

Kako bi podaci bili u odgovarajućem formatu za treniranje konvolucione neuronske mreže, uradile smo normalizaciju piksela dijeljenjem vrijednosti sa 255.0, čime su svi pikseli transformisani u raspon od 0 do 1, što poboljšava efikasnost učenja modela. Dalje, ciljne vrijednosti (labeli) smo pretvorile u one-hot vektore, što je neophodno pri korištenju funkcije gubitka categorical\_crossentropy, jer se radi o višeklasnoj klasifikaciji. Ulazni oblik podataka (input\_shape) postavljen je na (visina, širina, 3), što odgovara standardnom RGB formatu slike sa tri kanala, jer su nam sve slike veličine 32x32 sa 3 kanala. Također, važno je napomenuti da su slike učitane iz .pickle datoteka, što nam govori da su prethodno već bile obrađene – vjerovatno su rezirane, centrirane i balansirane po klasama, čime se dodatno osigurava konzistentnost i kvalitet podataka za treniranje modela.

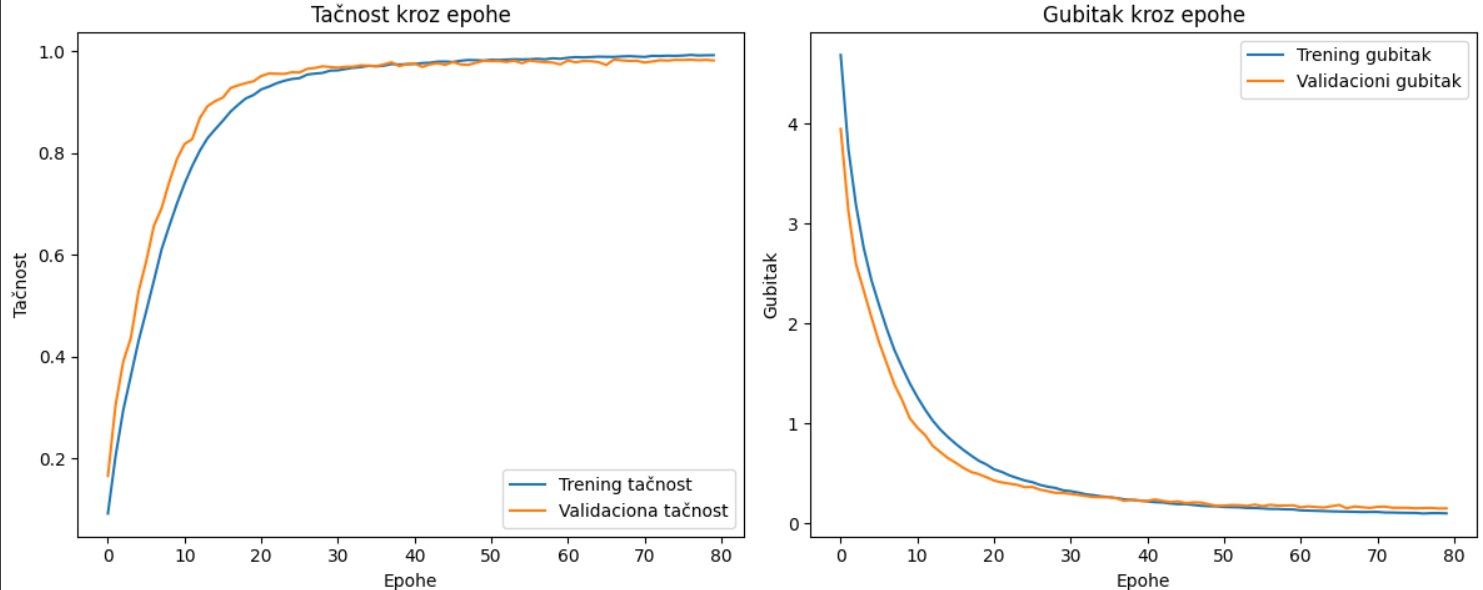
## 4.4. Treniranje modela:

Za treniranje konvolucione neuronske mreže koristile smo funkciju model.fit(...), kojom se pokreće proces učenja modela nad treniranim skupom podataka. Model smo trenirali tokom punih 80 epoha, uz veličinu mini-batcheva od 64 primjera po iteraciji.

U cilju sprječavanja overfittinga, implementirale smo dvije callback funkcije: EarlyStopping, koji bi zaustavio treniranje ako se validacioni gubitak ne poboljša u toku deset uzastopnih epoha (patience=10), te ReduceLROnPlateau, koji automatski smanjuje stopu učenja kada se validacioni rezultat stabilizuje. Ipak, u našem slučaju, kriteriji za rano zaustavljanje nisu bili ispunjeni, pa se treniranje odvijalo do kraja, odnosno svih 80 epoha. Ova činjenica ukazuje na to da validacioni gubitak nije stagnirao ili rastao kroz duži period, što sugeriše da nije došlo do pretreniranosti (overfittinga), te da je model stabilno učio tokom cijelog procesa treniranja. Arhitektura modela se sastoji od tri konvoluciona sloja, svaki sa ReLU aktivacijskom funkcijom, nakon kojih se koristi Batch Normalization za stabilizaciju i ubrzanje procesa učenja, te MaxPooling2D za redukciju dimenzionalnosti. Nakon konvolucionih slojeva koristi se GlobalAveragePooling2D sloj, koji sabira informacije bez značajnog gubitka performansi. Slijedi Dense sloj sa 128 neurona i ReLU aktivacijom, te Dropout sloj sa stopom od 0.6, koji služi kao dodatna regularizacija i štiti mrežu od overfittinga. Završni sloj koristi softmax aktivaciju, čime omogućava klasifikaciju slika u više različitih klasa.

Za funkciju gubitka koristile smo categorical\_crossentropy, koja je pogodna za višeklasne klasifikacijske zadatke, dok je optimizaciju modela vršio Adam algoritam sa početnom stopom učenja od 1e-4.

Tokom treniranja bilježile smo vrijednosti tačnosti i gubitka na trening i validacionom skupu, a rezultati prikazani putem grafika pokazuju da su tačnost i gubitak napredovali u kontinuitetu, bez ozbiljnog odstupanja između trening i validacionih performansi. Ovo potvrđuje da model nije pretrpio pretreniranost i da generalizuje dobro na nepoznatim podacima.



**Graf 2: Tačnost i gubitak kroz epohe treniranja modela**

## 4.5. Testiranje modela:

Nakon procesa treniranja i validacije, model je testiran na posebno odvojenom testnom skupu podataka, koji model nikada nije vidio tokom treniranja. Za testiranje smo koristile funkciju model.evaluate(X\_test, y\_test\_cat). Ova funkcija automatski izračunava dvije osnovne metrike tačnost (accuracy) i gubitak (loss) na test skupu. Time se mjeri koliko dobro model generalizira na nepoznate podatke. Dobijena tačnost na testnom skupu bila je 0.9762 (97.62%), dok je gubitak 0.1619, što je pokazatelj dobre sposobnosti modela da prepoznaje saobraćajne znakove i van skupa za treniranje. Pored ove dvije osnovne metrike koristile smo i dodatne metrike za dublju analizu:

* **Preciznost (precision)** – pokazuje koliko su tačni primjeri koje je model klasifikovao u određenu klasu.
* **Odziv ili pokrivenost (recall)** – pokazuje koliko dobro model pronalazi sve stvarne primjere određene klase.
* **F1-mjera (f1-score)** – predstavlja kombinaciju preciznosti i odziva u jedan broj, koji daje uravnoteženu sliku tačnosti modela za svaku klasu.
* **Podrška (support)** – ukupan broj stvarnih primjera te klase u testnom skupu.
* **Matrica konfuzije** – prikazuje tačnost po klasama, odnosno gdje se model najčešće greškom zabunio između sličnih znakova.

Na osnovu klasifikacionog izvještaja i matrice konfuzije, možemo zaključiti da model vrlo uspješno klasifikuje većinu saobraćajnih znakova. Za skoro sve klase, model ima visoku preciznost i odziv, što znači da rijetko pravi greške i da pronalazi većinu primjera tih klasa. Ipak, kod nekih klasa, kao što su klase broj 21 i 27, odziv je nešto niži, što znači da model ponekad ne prepozna sve njihove primjere. Ove klase imaju i manji broj podataka, pa model teže uči te primjere. Također, klasa broj 30 pokazuje nešto niže rezultate, što ukazuje da bi bilo dobro dodatno raditi na poboljšanju modela za te klase ili prikupiti više podataka.

## 4.6. Rezultati

Model pokazuje odlične performanse na poznatim (trening i validacionim) podacima, što potvrđuju visoke vrijednosti tačnosti i niska vrijednost gubitka tokom procesa treniranja vidljive na graficima (Graf 2.). Trening i validaciona tačnost su se kretale blizu maksimalnih vrijednosti, bez značajnih oscilacija, što ukazuje na dobru sposobnost modela da uči i generalizuje iz dostupnih podataka, što smo uspjele postići dodavanjem Dropout slojeva, L2 regularizacije i mehanizma EarlyStopping.

Tokom procesa testiranja slika koje nisu dio training skupa primjetile smo da model pokazuje slabije performanse. Na primjer, slike sa interneta koje smo koristile prilikom testiranja su imale svoje određene karakteristike (razlike u kvaliteti, osvjetljenju…) koje su uticale da model prikaže pogrešnu predikciju. Ove razlike nisu bile prisutne u trening skupu, pa model nije imao priliku da nauči takve varijacije. U nekim slučajevima rezultati su bili tačni, što pokazuje da model ima potencijal, ali i da postoji prostor za poboljšanje.

Sveukupno, možemo reći da model pokazuje vrlo dobre performanse i na testnom skupu i da je pouzdan u klasifikaciji saobraćajnih znakova. Postignuta tačnost od 97.62%, niska vrijednost gubitka i visoke metrike po klasama potvrđuju da model uspješno generalizuje. Ipak, kod nekoliko slabije zastupljenih klasa u podacima, performanse su nešto lošije, pa bi bilo dobro poboljšati model – dodavanjem augmentacije, više podataka i slično.

## 4.7. Vizualizacija rješenja:

Da bismo pojednostavile korištenje modela i unos podataka za predikciju, kreirale smo korisnički interfejs korištenjem platforme Anvil. Anvil omogućava povezivanje aplikacije sa Python kodom koji radi na nekoj drugoj platformi. Naša implementacija modela u Google Colab-u predstavlja serversku stranu aplikacije, a uz to je potrebno dodati i još nekoliko linija koda kako bi se uspostavila veza između servera i aplikacije. Preko Anvila je implementiran frontend aplikacije sa jednostavnim komandama. Ono što omogućava interfejs je učitavanje slike saobraćajnog znaka, koji se zatim šalje modelu da bi se izvršila predikcija. Rezultat predikcije se jasno ispisuje u polju iznad učitane slike, zajedno sa procentom pouzdanosti.

Da bi aplikacija uspješno radila potrebno je prvo aktivirati serversku stranu pokretanjem ćelije u Google Colab-u (posljednja ćelija u fajlu). Nakon toga se aplikacija otvara putem linka i može se koristiti. Pristupni link aplikacije je: <https://blond-thorough-national.anvil.app>.



**Slika 5. Izgled korisničkog interfejsa**

# Cjelokupni osvrt na problem i dobijeno rješenje

Smatramo da bi bolje rezultate na testnom skupu dobile ukoliko bi koristile augmentaciju u kodu, jer nam trenutno kod radi samo na originalnim slikama, bez nekih varijacija. Augmentacija bi nam pomogla da model prepoznaje znakove sa slika koje imaju različite karakteristike, što bi poboljšalo tačnost na testnom skupu. Posebno bi bila korisna rotacija, skaliranje i promjene osvjetljenja, kako bi model bio otporniji na slike koje se razlikuju od onih viđenih u trening fazi.

Generalno, dobijeni rezultati pokazuju da implementirani CNN model vrlo uspješno klasificira većinu saobraćajnih znakova. Postignuta tačnost od 97.62% na testnom skupu uz nisku vrijednost gubitka pokazuje da model ima visoku sposobnost generalizacije na nove podatke koji su slični treniranom skupu. Također, klasifikacioni izvještaj i matrica konfuzije potvrđuju da su i rjeđe klase većinom uspješno prepoznate, iako postoji prostor za dodatno unapređenje.

U poređenju s radovima iz prethodne faze, gdje su modeli imali jednostavnije arhitekture i nižu tačnost (uglavnom između 90% i 94%), ovaj model predstavlja značajno poboljšanje. Napredak se ogleda u dubljoj arhitekturi mreže, upotrebi regularizacijskih tehnika poput Dropout-a, Batch Normalization-a, te uvođenju callback funkcija koje su pomogle u stabilizaciji učenja.

Ono što se moglo uraditi bolje jeste uključivanje tehnike augmentacije još u fazi pripreme podataka, čime bi model imao širi spektar primjera za učenje, posebno za slabije zastupljene klase (npr. klase 21, 27 i 30). Pored toga, moglo se dodatno eksperimentisati s različitim arhitekturama modela – kao što su ResNet ili MobileNet – koje mogu dati još bolje rezultate uz manji broj parametara.

Također, evaluacija modela na potpuno vanjskim podacima (slike sa interneta) ukazuje na potrebu za boljom generalizacijom, što se može postići finijim balansiranjem skupa, dodatnim slojevima regularizacije i analizom pogrešnih klasifikacija (tzv. "error analysis").

Na kraju, integracija modela u aplikaciju preko Anvil platforme dodatno povećava upotrebljivost rješenja, jer korisnicima omogućava lako testiranje modela putem grafičkog interfejsa. Ova funkcionalnost pokazuje kako se istraživački rad može transformisati u konkretno, praktično rješenje.

U budućem radu, predlažemo fokus na proširenje skupa podataka, implementaciju data augmentacije, upotrebu naprednijih arhitektura i poboljšanje korisničkog interfejsa kako bi se sistem dodatno optimizovao za stvarnu upotrebu.

# Literatura

[1] Ziebinski, A., Cupek, R., Grzechca, D., & Chruszczyk, L. (2017). Review of advanced driver assistance systems (ADAS). *AIP Conference Proceedings*. Preuzeto sa: <https://doi.org/10.1063/1.5012394>

[2] Lim, X.R., Lee, C.P., Lim, K.M., Ong, T.S., Alqahtani, A., Ali, M. (2023). Recent Advances in Traffic Sign Recognition: Approaches and Datasets. *Sensors*. Preuzeto sa: <https://doi.org/10.3390/s23104674>

[3] Zhao, G., Ma, F., Qi, W., Zhang, C., Liu, Y., Liu, M., & Ma, J. (2024). TSCLIP: Robust CLIP Fine-Tuning for Worldwide Cross-Regional Traffic Sign Recognition. *arXiv (Cornell University)*. Preuzeto sa: <https://arxiv.org/pdf/2409.15077>

[4] Gan, Y., Li, G., Togo, R., Maeda, K., Ogawa, T., & Haseyama, M. (2024). Think twice before recognizing: large multimodal models for general fine-grained traffic sign recognition. *JOURNAL OF LATEX CLASS FILES*, *14*(8). Preuzeto sa: <https://arxiv.org/pdf/2409.01534>

[5] Farzipour, A., Manzari, O. N., & Shokouhi, S. B. (2023). Traffic Sign Recognition Using Local Vision Transformer. *13th International Conference on Computer and Knowledge Engineering*. Preuzeto sa: <https://arxiv.org/pdf/2311.06651>

Podaci korišteni za obradu u ovom projektu su dio dataset-a autora Valentyna Sichkara, a preuzeti su sa linka:

<https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-preprocessed/data?select=datasets_preparing.py>

1. Ziebinski, A., Cupek, R., Grzechca, D., & Chruszczyk, L. (2017). Review of advanced driver assistance systems (ADAS). *AIP Conference Proceedings*. [↑](#footnote-ref-0)
2. Lim, X.R., Lee, C.P., Lim, K.M., Ong, T.S., Alqahtani, A., Ali, M. (2023). Recent Advances in Traffic Sign Recognition: Approaches and Datasets. *Sensors*. [↑](#footnote-ref-1)
3. Zhao, G., Ma, F., Qi, W., Zhang, C., Liu, Y., Liu, M., & Ma, J. (2024). TSCLIP: Robust CLIP Fine-Tuning for Worldwide Cross-Regional Traffic Sign Recognition. *arXiv (Cornell University)*. [↑](#footnote-ref-2)
4. Gan, Y., Li, G., Togo, R., Maeda, K., Ogawa, T., & Haseyama, M. (2024). Think twice before recognizing: large multimodal models for general fine-grained traffic sign recognition. *JOURNAL OF LATEX CLASS FILES*, *14*(8). [↑](#footnote-ref-3)
5. Gan, Y., Li, G., Togo, R., Maeda, K., Ogawa, T., & Haseyama, M. (2024). Think twice before recognizing: large multimodal models for general fine-grained traffic sign recognition. *JOURNAL OF LATEX CLASS FILES*, *14*(8). str 1. [↑](#footnote-ref-4)
6. Farzipour, A., Manzari, O. N., & Shokouhi, S. B. (2023). Traffic Sign Recognition Using Local Vision Transformer. *13th International Conference on Computer and Knowledge Engineering*. str. 3. [↑](#footnote-ref-5)
7. Farzipour, A., Manzari, O. N., & Shokouhi, S. B. (2023). Traffic Sign Recognition Using Local Vision Transformer. *13th International Conference on Computer and Knowledge Engineering*. [↑](#footnote-ref-6)
8. Farzipour, A., Manzari, O. N., & Shokouhi, S. B. (2023). Traffic Sign Recognition Using Local Vision Transformer. *13th International Conference on Computer and Knowledge Engineering*. str. 4. [↑](#footnote-ref-7)