**UNIVERZITET U ZENICI**

**POLITEHNIČKI FAKULTET**

**SEMINARSKI RAD IZ PREDMETA:**

Prepoznavanje oblika i obrada slike

|  |  |
| --- | --- |
| **Tema rada:** | **Detekcija i klasifikacija saobraćajnih znakova** |

|  |  |
| --- | --- |
| **Predmetni nastavnik:** | van. prof. dr. Nermin Goran |

|  |  |
| --- | --- |
| **Student:** | Ajdin Bukvić |
| Broj indeksa: | II-89 |
| Usmjerenje: | Softversko inženjerstvo |
| Godina studija: | 1. godina, 2. ciklus |
| **Rezultat rada:** | Primjena različitih AI/ML algoritama za detekciju i klasifikaciju saobraćajnih znakova na posebno pripremljenom skupu podataka, uz osvrt i analizu na rezultate implementacije korištenjem testnih slika i videa, radi evaluacije performansi. |

Datum: 05.02.2024.

**SADRŽAJ**

[1. UVOD 2](#_Toc157864402)

[2. DETEKCIJA 3](#_Toc157864403)

[3. KLASIFIKACIJA 4](#_Toc157864404)

[4. PRAKTIČNI DIO 5](#_Toc157864405)

[4.1. Alati, biblioteke i radno okruženje 5](#_Toc157864406)

[4.2. Skup podataka (dataset) 6](#_Toc157864407)

[4.3. Priprema i analiza podataka 8](#_Toc157864408)

[5. IMPLEMENTACIJA 10](#_Toc157864409)

[5.1. YOLO algoritam 10](#_Toc157864410)

[5.2. Analiza rezultata detekcije 12](#_Toc157864411)

[5.3. CNN model 13](#_Toc157864412)

[5.4. SVM algoritam 13](#_Toc157864413)

[5.5. Poređenje rezultata testiranja klasifikacije 13](#_Toc157864414)

[5.6. Detekcija i klasifikacija na videu 13](#_Toc157864415)

[6. ZAKLJUČAK 13](#_Toc157864416)

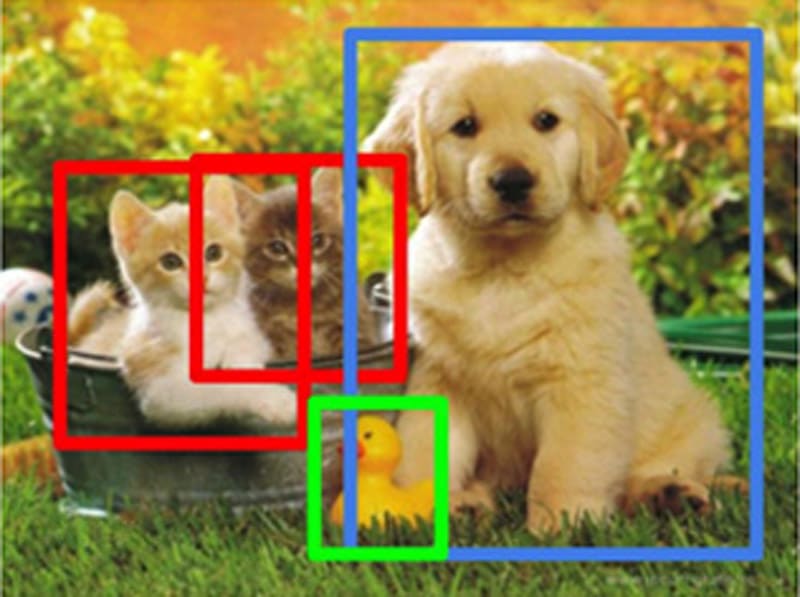
[7. LITERATURA 13](#_Toc157864417)

# UVOD

Razvojem tehnologije zadnjih nekoliko godina došlo je do velikog porasta za uključivanje tehnologije u razne poslovne procese i zadatke koji se mogu na vrlo brz i efikasan način riješiti primjenom različitih tehnologija. Područje mašinskog učenja i vještačke inteligencije je znatno napredovalo i postalo je neophodan alat za izvršavanje raznih zadataka. Pošto se svakodnevno pojavljuju novi podaci na internetu generisani iz raznih izvora, ovi algoritmi procesom treniranja modela su sve uspješniji i precizniji. Poseban dio vještačke inteligencije odnosi se na područje kompjuterske vizije (eng. computer vision). Upotreba naprednih algoritama omogućava izvršavanje različitih operacija na slikama ili videu, koji se tretiraju kao poseban tip podataka, za razliku od standardnih tekstualnih podataka. Kroz ovaj seminarski rad će biti prikazani procesi detekcije i klasifikacije na primjeru saobraćajnih znakova. Za odvijanje ovih procesa prethodi nekoliko operacija nad slikama (ili videu). Ove operacije mogu uključivati obradu, procesiranje, segmentaciju i analizu slike. Digitalna slika je predstavljena skupom piksela, te je prilikom rada sa slikama potrebno uzeti u obzir različite osobine slike, kao što su boje i veličina. Naknadni zadaci obrade su otkrivanje ivica, oblika i objekata. Praktična primjena detekcije i klasifikacije obuhvata razne segmente, koji olakšavaju svakodnevne poslove, a na konkretnom primjeru saobraćajnih znakova može se koristiti u vozilima.

# DETEKCIJA

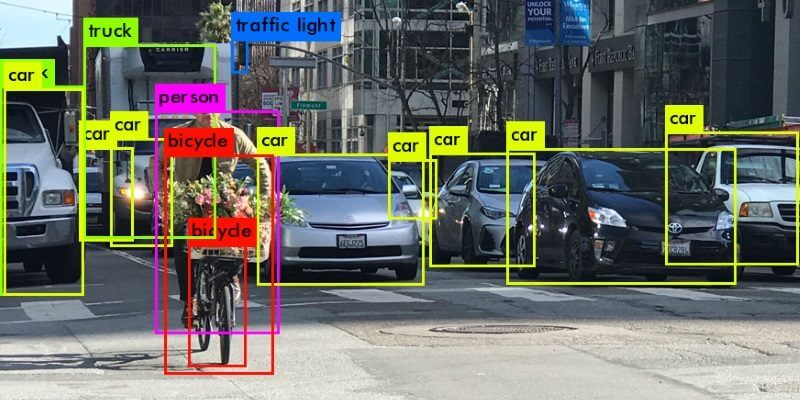
Detekcija slike je proces koji obrađuje sliku i otkriva objekte u njoj. Detekcija obično prethodi procesu klasifikacije, jer je prvo potrebno odrediti (izdvojiti) tražene objekte sa slike, kako bi se oni nakon toga mogli klasifikovati, odnosno dodijeliti jednoj od klasa (grupa, kategorija). Procesom detekcije moguće je utvrditi tačnu lokaciju nekog objekta na slici ili saznati tačan broj objekata na slici. Objekat na slici može biti bilo šta iz prirode, kao naprimjer čovjek, životinja, vozilo, drvo, kuća i mnogi drugi. Detekcijski modeli rade na principu učenja neophodnih karakteristika, kako bi znali odrediti da li je nešto na slici traženi objekat ili nije. Određivanje objekata na slici je aktivnost koja je jednostavna za ljudski mozak, ali za sam računar predstavlja znatan izazov. Za detekciju objekata algoritmu se prvo moraju „dostaviti“ slike koje sadržavaju traženi objekat, odnosno da već imaju unaprijed određene lokacije (koordinate) objekta. Detekcijom je potrebno utvrditi da detektovani objekat najviše odgovara traženom objektu. Pošto na slici može biti više različitih objekata, algoritmi se trude da odrede veliki broj karakteristika koje su specifične za jedan traženi objekat, kao što su boja, oblik, veličina. Efikasnost detekcije se poboljšava kada se koristi veliki broj slika na kojima se traženi objekat nalazi na različitim pozicijama, te da se kvalitet slika znatno razlikuje (od slika visoke kvalitete do slika vrlo niske kvalitete). Također, mogu se uključiti i razni parametri koji mogu utjecati na rezultate detekcije kao što su: ugao slikanja, vrijeme i doba dana, blizina (daljina) objekta na slici, broj objekata i drugi. Napredni zadaci detekcije se mogu iskoristiti u aplikacijama za praćenje objekata, odnosno primjena detekcije na videu. Posebno su aktuelni primjeri detekcije pješaka, vozila, lica, teksta, poze.



Slika 1

# KLASIFIKACIJA

Klasifikacija je proces označavanja objekata na slici. Ovim pristupom se očekuje da su objekti na slici prethodno detektovani, te ih je sada potrebno pridružiti određenoj klasi. Pošto klasifikacija dolazi nakon detekcije, to znači da se detekcijom samo izuzimaju objekti, a klasifikacijom se određuje koji je objekat u pitanju. Klasifikacija daje rezultat koji govori da li je objekat prisutan na slici ili nije, odnosno kada algoritam primi sliku on izračunava vjerovatnoću da li je to objekat A ili objekat B (za primjer s dvije klase). Najjednostavniji je upravo ovaj binarni klasifikator, dok su napredniji modeli multiklasni klasifikatori (više od dvije klase). Na primjeru binarne klasifikacije za dvije klase A i B, te učitanu sliku, klasifikator radi na način da prvo mora izdvojiti neke karakteristike (obilježja) koje međusobno razlikuju ove dvije klase. Izdvajanjem karakteristika se određuje kojoj klasi objekat više pripada, odnosno s kojom ima najviše sličnosti. Kada se kreće s procesom treniranja klasifikatora, neophodno je da se koristi označeni skup podataka, koji pored podataka (slika) sadrži i informacije o klasi kojoj pripada objekat na slici. Istrenirani model nakon toga će biti u stanju neku novu sliku biti u stanju dodijeliti unaprijed definisanoj klasi. Problemi kod klasifikacije mogu nastati kada se objekat koji pripada klasi A greškom dodijeli klasi B, što ukazuje na nedostatke istreniranog modela. Također, važno je napomenuti da se objekat može klasifikovati kao objekat A ili objekat B, ali ne i jedno i drugo istovremeno. Pored toga, najveći problem jeste dodijeljivanje kategorije A ili B objektu koji ne pripada ni jednoj od kategorija.



Slika 2

# PRAKTIČNI DIO

Kroz praktični dio implementirat će se algoritmi (modeli) za detekciju i klasifikaciju saobraćajnih znakova. Odabrani domen koji se odnosi na saobraćajne znakove, može se iskoristiti kod autonomnih ili samovozećih vozila. Proces implementacije može se primijeniti u raznim domenima, na vrlo sličan način, ali je prvo potrebno odrediti područje primjene i nabaviti odgovarajući skup podataka.

## Alati, biblioteke i radno okruženje

Praktični dio je implementiran pomoću Python programskog jezika (verzija 3.10.0). Za zadatke koji obuhvataju analizu podataka i treniranje raznih modela mašinskog učenja i vještačke inteligencije Python je idealan kandidat. Napredne mogućnosti koje pruža Python uključuju vrlo jednostavno i efikasno rukovanje podacima i njihovu vizualizaciju. Razlog zbog kojeg je Python posebno postao popularan u ovakvim projektima jeste njegova fleksibilnost, široka podrška zajednice, platformska nezavisnot, te veliki ekosistem biblioteka koje pružaju mnoge gotove metode za rad s velikim skupovima podataka i njihovo procesiranje. Također, korištenje algoritama je vrlo jednostavno s unaprijed definisanim parametrima, kako bi se u samo nekoliko linija koda mogli postići očekivani rezultati. Biblioteke koje su korištene u projektu su:

* numpy
* pandas
* mathplotlib
* scikit-learn
* scikit-image
* os
* tensorflow/keras
* cv2
* ultralytics

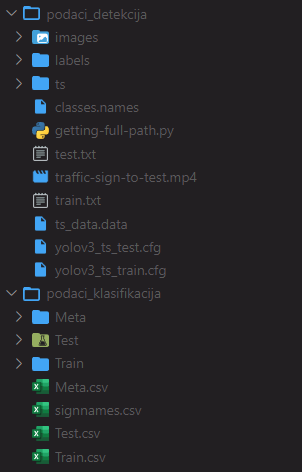
Kod je pisan unutar Visual Studio Code okruženja, uz korištenje posebnog formata (ekstenzije) Jupyter Notebook (.ipynb), koja je prilagođena pisanju Python skripti. Jupyter Notebook pruža podršku za kreiranje i dijeljenje dokumenata koji sadrže „živi“ (izvršeni) kod, zajedno s vizualnim prikazom (grafovi, slike), te formatirani opisni dio teksta. Implementacija se sastoji od analize skupa podataka, detekcije korištenjem YOLO algoritma, klasifikacije korištenjem CNN i SVM modela, te testiranje modela na novim slikama i videu, uz pregled performansi. Što se tiče same strukture projekta, organizovan je u nekoliko foldera i fajlova. Projekat se sastoji od 6 Jupyter Notebook (.ipynb) fajlova s implementiranim algoritmima. Također, postoje odvojeni folderi za podatke za detekciju i klasifikaciju, kao i folderi za output (izlazne slike nakon detekcije i klasifikacije), te modeli, za spremanje istreniranih modela. O samim podacima za treniranje i testiranje će biti više govora u nastavku.

## Skup podataka (dataset)

Za implementaciju praktičnog dijela korištena su dva nezavisna skupa podataka, jedan za detekciju, a drugi za klasifikaciju:

* GTSDB (German Traffic Sign Detection Benchmark)
* GTSRB (German Traffic Sign Recognition Benchmark)

Skupovi podataka su korišteni na takmičenju održanom na Međunarodnoj zajedničkoj konferenciji o neuronskim mrežama (IJCNN) 2011. godine, a kreirani su od strane samog instituta (<https://benchmark.ini.rub.de/>). Nakon održanog takmičenja, objavljeni su podaci o najboljim rezultatima, te su nakon toga skupovi postali dostupni svima na korištenje, radi vlastitog istraživanja. Skupovi su preuzeti s web stranice Kaggle (<https://www.kaggle.com>), koja omogućava korisnicima da pronađu različite skupove podataka, te da vrše svoja istraživanja u implementaciji i izgradnji modela vještačke inteligencije. Ideja koja stoji iza ove stranice je kolaboracija s drugim naučnicima, te dijeljenje vlastitih projekata i rezultata. Također, redovno se organizuju razna takmičenja i izazovi koji se zasnivaju na rješavanju probleme “nauke o podacima” (data science), a sama stranica nudi i mogućnost pokretanja kodova na njihovim serverima. Prvi skup podataka GTSDB se koristi za detekciju saobraćajnih znakova i sastoji se od 741 slike, rezolucije 1360 x 800 piksela u .jpg formatu. Na slikama se nalaze različite scene na kojima su prikazani saobraćajni znakovi, zajedno s drugim objektima. Uz svaku sliku, pridružen je prateći .txt fajl koji sadrži informacije o x i y koordinatama znaka na slici (za treniranje), širini i visini, te broju znakova na slici. Skup podataka je prilagođen YOLO algoritmu (koji će biti objašnjen u nastavku). Također, u skupu se nalaze i gotove konfiguracije za YOLOv3 algoritam (spremljene u .cfg fajlovima), koje će poslužiti za kreiranje vlastitog YOLO modela. Pošto se ovaj algoritam može koristiti i za klasifikaciju, radi jednostavnosti sadrži 4 osnovne klase saobraćajnih znakova (generalizovano): zabrana (prohibitory), opasnost (danger), obavezni (mandatory), ostali (other). Ove klase služe samo radi izdvajanja saobraćajnih znakova od ostalih objekata, a za detaljniju klasifikaciju će se koristiti drugi skup podataka, odnosno GTSRB. Drugi skup podataka GTSRB služi za klasifikaciju, te sadrži oko 50.000 slika podijeljenih u 43 klase. Klase su opisane unutar signnames.csv fajla koji sadrži pune nazive svake klase (na engleskom jeziku). Pored toga, slike su podijeljene u 2 foldera Train i Test (otprilike podijeljenih na oko 40.000 trening slika i 10.000 test slika). Train folder se sastoji od 43 podfoldera (folder za svaku od klasa). Test folder sadrži slike koje će se koristiti za procjenu preciznosti modela nakon treniranja. Također, u posebnom Meta folderu su izvorne slike za svaku od 43 klase (koje služe za detaljnije razumijevanje klasa). Za svaki folder, postoji i prateći .csv fajl Train.csv, Test.csv, Meta.csv, koji sadrži sljedeće kolone: width (širina), height (visina), ROI.x1, ROI.y1, ROI.x2, ROI.y2, ClassId (jedinstveni identifikator klase), Path (putanja do slke). ROI ili Region of Interest su x i y koordinate koje određuju znak na slici, odnosno ROI (područje interesovanja) predstavlja lokaciju na kojoj se nalazi znak u odnosu na cijelu sliku. Sve slike su u .png formatu.



Slika 3

## Priprema i analiza podataka

Nakon preuzimanja skupova podataka, neophodno ih je pripremiti za aktivnosti detekcije i klasifikacije. Kada je u pitanju skup za detekciju, on dolazi s 741 slikom, te ih je prvo potrebno podijeliti u tri odvojena foldera (Train, Test, Valid). Kada su folderi kreirani, potrebno je odrediti omjer kojim će podaci (slike) biti podijeljeni po folderima. Odabrani omjer je 70% trening, 20 % validacija i 10 % test. Da bi se postigli što bolji rezultati slike se prije podjele nasumično „miješaju“, korištenjem biblioteke random i metode shuffle. Rezultati koji se nakon toga dobijaju su: 518 trening slika, 148 validacijskih slika i 75 test slika spremljenih u zasebne foldere. 

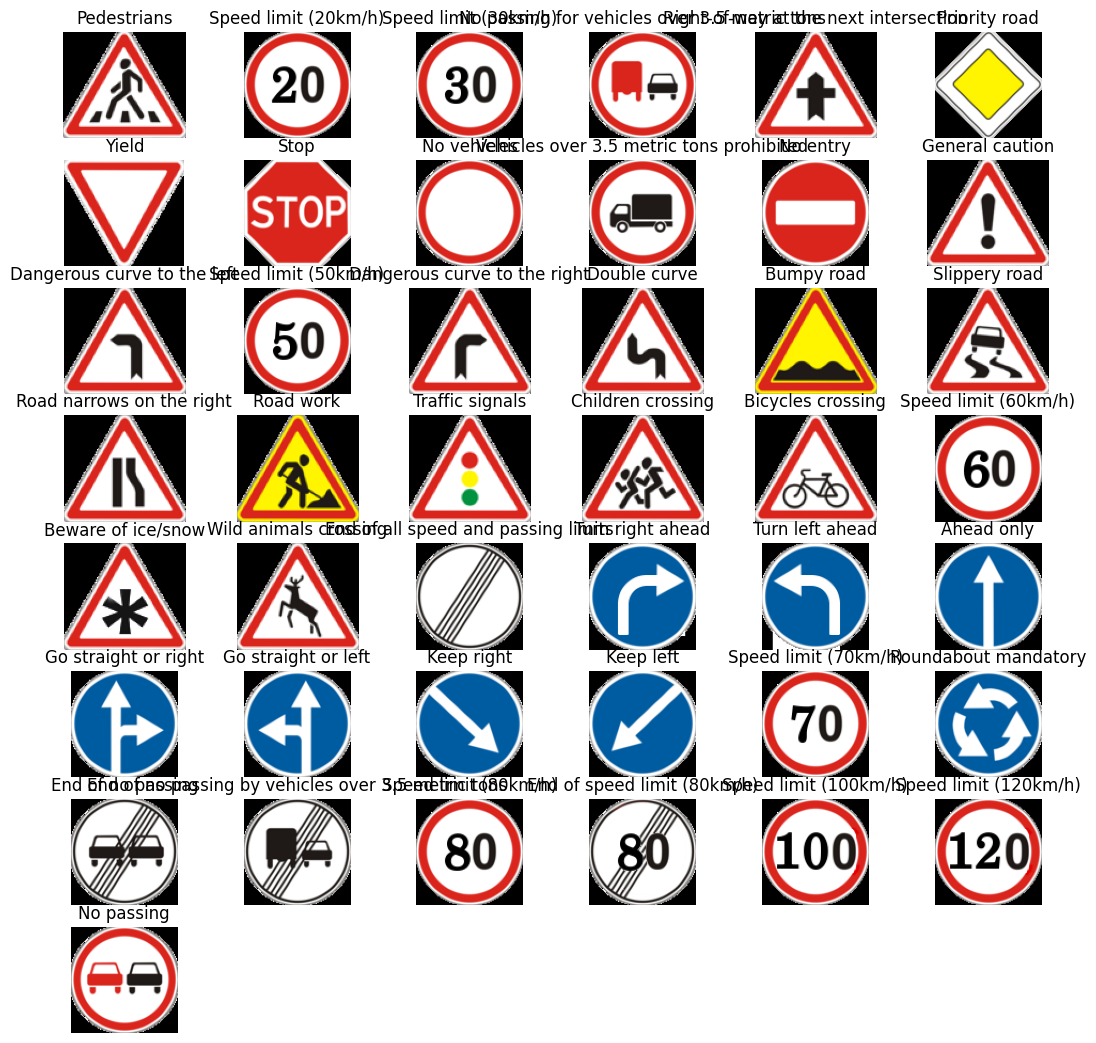
Slika 4

Istraživanje skupa za klasifikaciju sastoji se od određivanja ključnih karakteristika za svaku sliku kao što su minimalna i maksimalna visina slika, čime je utvrđeno da sve slike u prosjeku imaju širinu između 25 – 243, te visinu između 25 – 225. Može se vidjeti da su slike prilično malih dimenzija, jer za razliku od slika za detekciju sadrže samo znak na slici, bez bilo kakvih drugih objekata ili pozadine. Sljedeći korak je vizualizacija nekoliko slučajno odabranih slika radi utvrđivanja kvalitete slika. Ovim korakom je utvrđeno da većina slika ima vrlo nisku kvalitetu, pa je na nekim slikama teško čak i ljudskom oku procijenititi pravu klasu slike. Također, slike su uslikane u različitim dobima dana, imaju različito osvjetljenje i zamućenje. Na slici ispod se mogu vidjeti kako izgledaju 3 različite klase znakova iz trening, test i meta foldera (s lijeva na desno).



Slika 5

Kada je u pitanju razumijevanje klasa znakova i njihova distribucija na trening i test skupu podataka, utvrđeno je da neke klase dominiraju nad ostalim po broju slika. Većina klasa ima oko 500 slika, dok nekoliko klasa ima i do 2000 slika. Kao što se može vidjeti neke od klasa su ograničenja brzine, dozvoljeni smjerovi, STOP i ostali.



Slika 6

# IMPLEMENTACIJA

Kada su podaci (slike) spremni za daljnju obradu može se krenuti s konkretnom implementacijom algoritama za detekciju i klasifikaciju. Nakon detaljne analize po pitanju kvalitete, veličine i formata slika ove slike se mogu koristiti za treniranje modela, a zatim i za evaluaciju njegovih performansi, odnosno tačnosti i preciznosti. Cilj je prvo izvršiti neophodno pretprocesiranje slike, te slike kao takve proslijediti modelu, koji će na kraju biti u stanju detektovati ili klasifikovati novu sliku koja mu se proslijedi.

## YOLO algoritam

YOLO (You Look Only Once) je jedan od najpopularnijih algoritama za detekciju objekata, koji radi u realnom vremenu, te se koristi u mnogim komercijalnim proizvodima od strane najvećih tehnoloških kompanije, koje koriste kompjuterski vid. YOLO algoritam je prvi put originalno objavljen 2016. godine i već tada je ostvario znatno veće brzine od ostalih algoritama. Nakon toga objavljeno je nekoliko unaprijeđenih verzija i varijanti YOLO algoritma, gdje svaka nova verzija pruža bolje performanse i efikasnost. Verzija koja je korištena u ovom projektu je trenutno najnoviji YOLOv8 iz 2023. godine, koji sa sobom donosi inovacije kao što su poboljšanje mozaika podataka, samo-suparnička obuka i normalizacija mini serija. YOLOv8 razvio je Ultralytics (<https://docs.ultralytics.com/>), te je dizajniran da bude brz, precizan i jednostavan za korištenje. YOLO algoritam koristi karakteristike naučene od strane duboke konvolucionalne neuronske mreže (CNN). YOLO radi tako što model za detekciju pregleda cijelu sliku u vremenu testiranja, što znači da globalni kontekst na slici daje informacije o predviđanjima. Algoritam prvo dijeli sliku u mrežu (grid). Svaka ćelija mreže služi za predviđanje određenog broja graničnih okvira (frame-ova) oko objekata koji imaju visoke ocjene s unaprijed definisanim klasama. U svakom graničnom okviru identifikuje samo jedan objekat. Nakon toga se generišu granične „kutije“ (boxes) tako što se grupiraju dimenzije osnovnih okvira kako bi se pronašli najčešći oblici i veličine. Granične kutije mogu imati različite omjere širine i visine, koji se određuju prije treninga korištenjem nekog od algoritama grupisanja, kao naprimjer K-means algoritam. Naredni korak jeste poređenje koliko se odabrana kutija poklapa s osnovnom kutijom (odnosno unaprijed definisanom kutijom koja predstavlja izdvojeni objekat – ROI). Na kraju algoritam vraća oznake kao što su x i y koordinate, klasu i sigurnost predviđanja (confidence score). Ono što posebno karakteriše YOLO jeste što može napraviti duple detekcije za isti objekat, pa se iz rog razloga primjenjuje nemaksimalno potiskivanje, kako bi se uklonilo dupliciranje s nižim sigurnostima (povjerenjem), a zadržava se samo jedna s najvećim rezultatom.

Za konkretni primjer prvo je instalirana biblioteka „ultralytics“ kako bi se mogle koristiti pogodnosti najnovijeg YOLOv8 algoritma. Pošto su slike za detekciju već podijeljenje u 3 foldera (Train, Valid, Test) slijedi korak kreiranja .yaml konfiguracijskog fajla, kako bi YOLO algoritam mogao znati kako će odrediti putanje do slike, te koje su specifične klase. Poslije toga potrebno je kreirati YOLO model, koji će automatski preuzeti istrenirane težinske vrijednosti samog modela (yolov8x.pt), koji je već osposobljen za detekciju nekih vrsta objekata (naprimjer ljudi i vozila). Sada je cilj koristiti već naučena znanja, kako bi se model prilagodio posebnoj vrsti podataka (slika), odnosno slika na kojima se nalaze saobraćajni znakovi. Proces treniranja modela je jednostavan pozivom metode train na samom modelu, uz prosljeđivanje parametara, kao što su broj epoha, preferiranu veličinu slike koja se prosljeđuje algoritmu, te prethodno generisani .yaml konfiguracijski fajl. Treniranje za 10 epoha je trajalo približno 4 i pol sata, što može znatno varirati u odnosu na specifikacije računara. Istrenirani model se može sačuvati u posebnom .onnx formatu, kako bi se mogao koristiti u nastavku za daljnja unaprijeđenja i testiranja.



Slika 7

## Analiza rezultata detekcije

Pri završetku procesa treniranja, automatski se kreira poseban folder runs u kojem se čuvaju podaci o detekciji i treningu, koji su grafički predstavljeni i spremljeni u formatu slike (.png i .jpg). Neki od najvažnijih grafičkih prikaza uključuju: konfuzijsku matricu, F1 confidence krivulju, oznake, korelogram oznaka, P krivu (Precision-Confidence), PR krivu (Precision-Recall), R krivu (Recall-Confidence). Istrenirani model se onda može iskoristiti za kreiranje predikcija, tako što se metodi predict proslijedi putanja do slike (ili u ovom slučaju cijeli folder testnih slika, koje nisu korištenje prilikom treniranja). Neki od dodatnih parametara koji se mogu uključiti su da li će se slika s detektovanim saobraćajnim znakovima spremiti na disk ili ne, te koji će biti prag sigurnosti predviđanja (decimalni broj između 0-1). Proces detekcije na testnim slikama u ovom slučaju sprema nove slike na disk, a također i daje informacije o broju detektovanih znakova po slici. Još jedna od prednosti korištenja YOLO algoritma je to što se na tim slikama oko objekata (znakova) automatski iscrtava pravougaonik koji označava ROI, zajedno s nazivom klase i povjerenju predikcije. Kada je u pitanju evaluacija modela, vrlo jednostavno se određuje pozivom metode val na samom modelu, koji daje preciznost za trening i validacijske slike, a ako mu se proslijedi parametar „test“, dobijaju se rezultati preciznosti za testne slike. Ova metrika je izražena kao mAP (mean average precision). U konkretnom primjeru dobijaju se vrijednosti 0.66 za validacijski skup i 0.70 sa testni skup (obe zaokružene na dvije decimale).



Slika 8

## CNN model

## SVM algoritam

## Poređenje rezultata testiranja klasifikacije

## Detekcija i klasifikacija na videu

# ZAKLJUČAK

# LITERATURA