Detekcija i prepoznavanje saobraćajnih znakova sa slike

Uroš Petrić

Milica Škipina

UVOD

U ovom projektu bavili smo se temom detekcije i prepoznavanja saobraćajnih znakova na osnovu njihove slike. Za izradu projekta korišćen je Python u Jupyter Notebook okruženju, a od tehnologija je korišćeno Duboko učenje (**Deep learning**) zajedno sa Konvolucionim neuronskim mrežama (**Convolutional neural network - CNN**).

Duboko učenje smo koristili zbog rada sa velikom količinom podataka koji su predstavljeni putem slika a **CNN** smo koristili kako bi prepoznali znak na slici i klasifikovali ga**. Keras** je korišćen za čuvanje istreniranog modela, **Matplotlib** za iscrtavanje grafa preciznosti i gubitka, **Pandas** za čitanje csv file-a i povezivanje slika sa opisom i **PIL biblioteku** za pretvaranje slike u niz.

DATASET

Prilikom izrade projekta je korišćen dataset koji se sastoji od **50000 slika** podeljenih u **42 različite klase** gde svaka klasa predstavlja jednu vrstu znaka.

Dataset je na samom početku podeljen u 2 grupe:

- **Trening skup**

- **Testni skup**

KREIRANJE MODELA CNN

Sam projekat je podeljen na **4 dela:**

1. Istraživanje dataset-a

2. Kreiranje CNN modela

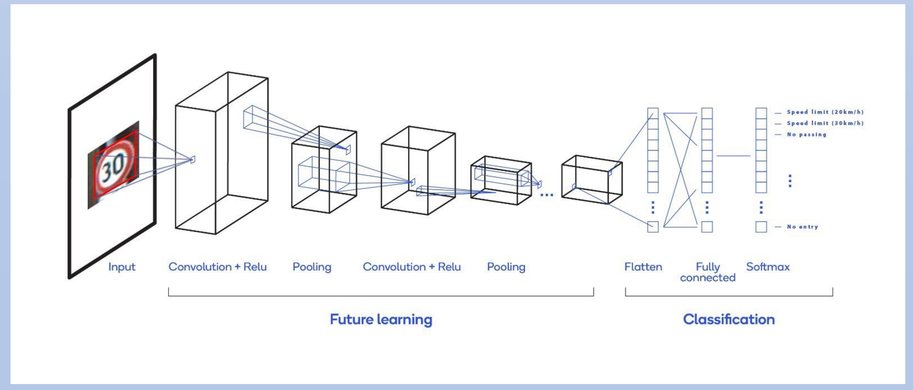
3. Trening i validacija modela

4. Testiranje modela testnim podacima

1. ISTRAŽIVANJE DATASETA

Iz trening foldera prolazimo kroz svih **43 klasa** i iteracijom kroz slike resizujemo ih na veličinu **30x30** i ubacujemo ih u 2 liste odnosno 2 niza. Podaci su predstavljeni kao (**39209,30,30,3**) gde prvi broj predstavlja broj slika u skupu, druga dva broja su širina i visina a četvrti broj označava da su slike u RGB formatu.

Nakon toga preko **train_test_split()** metode delimo dataset na trening skup i testni skup koji će se posle koristiti za trening i validaciju modela.



2. KREIRANJE CNN MODELA

Konvoluciona neuronska mreža se koristi za samu klasifikaciju jer predstavlja najbolje rešenje za to. Sama arhitektura modela sadrži više slojeva kao što su:

- 2 Conv2D

- MaxPool2D

- Dropout

- Flatten

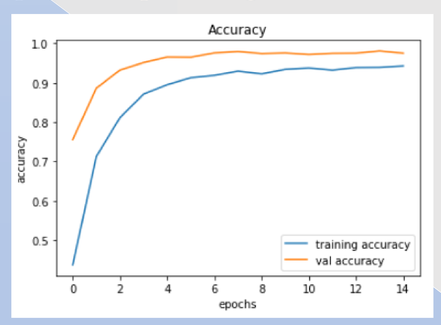
- Dense Fully connected

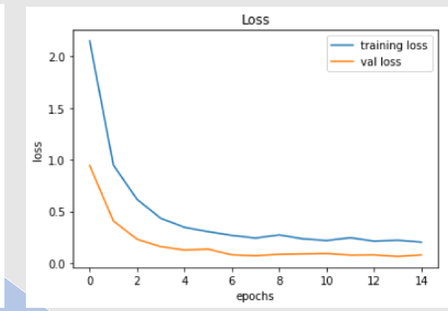
- Dense

Model se kompajlira pomoću **Adam optimajzera**, zatim koristimo **categorical_crossentropy** za prikaz gubitka jer imamo više klasa a od metrika za evaluaciju problema koristimo **accuracy** kao i kerasovu metriku **MeanIoU** za prikaz srednje vrednosti Intersection over Union.

3. TRENING I VALIDACIJA MODELA

Model treniramo pomoću **model.fit()** funkcije batch veličine 64 i 15 epoha nakon čega dobijamo preciznost od **94%** u proseku što je i prikazano pomoću grafa.





4. TESTIRANJE MODELA POMOĆU TESTNOG DATASETA

Za proveru tačnosti modela potrebno je ponoviti proceduru za pribavljanje dataseta samo što ovaj put to radimo iz testnog foldera. Tačnost modela proveravamo pomoću **accuracy_score** gde dobijamo da naš model ima **95%** tačnosti.