**Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Računarstvo usluga i analiza podataka

**Seminarski rad**

„Klasifikacija zastava država Balkana“

Sven Suk

Zvonimir Medak

Ivan Štajcer

Osijek, 2021.

Sadržaj

[1. UVOD 1](#_Toc64315790)

[2. POSTUPAK I USPOREDBA NAČINA RJEŠAVANJA ZADATKA 2](#_Toc64315791)

[2.1. Odabir podatka 2](#_Toc64315792)

[2.2. Odabir značajki 3](#_Toc64315793)

[2.2.1. Prewitt 3](#_Toc64315794)

[2.2.2. Fourier 4](#_Toc64315795)

[2.3. Max Pooling 6](#_Toc64315796)

[2.4. Klasifikatori i Azure modeli 6](#_Toc64315797)

[2.4.1. Fourier skup podataka i model 7](#_Toc64315798)

[2.4.2. Prewitt vertical skup podataka i model 8](#_Toc64315799)

[2.4.3. Prewitt horizontal skup podataka i model 10](#_Toc64315800)

[3. Programsko rješenje 12](#_Toc64315801)

[3.1 Azure ML 12](#_Toc64315802)

[3.2 Python 13](#_Toc64315803)

[3.3 React 15](#_Toc64315804)

[4. Zaključak 17](#_Toc64315805)

LITERERATURA

# UVOD

Tema ovoga seminarskog rada je prepoznavanje zastava država Balkana putem machine learninga i usporedba raznih klasifikatora. Kako bi se ovakvo nešto postiglo potrebno je pronaći skup podataka kojim će se moći obučiti i napraviti model za prepoznavanje. Skup podataka treba dobro predstavljati pojmove koji se trebaju prepoznati kako bi model što bolje funkcionirao. Iz podataka je prvotno trebalo izvući značajke (*feature extraction*) te spremiti ih u csv (*comma separated values*) datoteku kako bi se mogla učitati na Azure. Korišteni jezik za skripte je Python, a napravljena je i web aplikacija koja koristi napravljeni model.

Web aplikacija omogućava korisnicima da učitaju sliku zastave države Balkana te da ju model pokuša prepoznati. Omogućuje osobama brzo prepoznavanje tražene zastave.

# POSTUPAK I USPOREDBA NAČINA RJEŠAVANJA ZADATKA

U ovom poglavlju opisat će se pronalazak i odabir skupa podataka zastava, postupci odabira značajki iz slike te klasifikatori i Azure modeli. Također, usporedit će se korišteni postupci odabira značajki podataka te klasifikatori.

## Odabir podatka

Kako bi se što bolje napravio model prepoznavanja, potrebno je pronaći skup podataka koji će predstavljati pojedine klase koje je potrebno prepoznati. U našem slučaju potrebno je pronaći deset slika za svaku državu (u slučaju modela to su klase). Kako bi se model istrenirao tako da uspije prepoznati i zastave koje nisu u potpunosti ravne, dodane su slike na kojima se zastava vijori. Slike su pronađene na internetu, a neke slike bilo je potrebno isjeći kako bi se prikazivala samo zastava, a ne i pozadina.

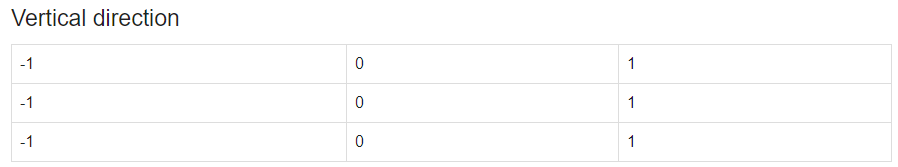


Slika 2.1, prikaz korištenog skupa podataka

## Odabir značajki

### Prewitt

Prewitt operator koristi se za odabir rubnih značajki slike, a postoje dva načina odabira, horizontalni i vertikalni. Rubovi se računaju koristeći razliku između odgovarajućih intenziteta pixela slike. Koriste se *derivative masks* jer je slika signal, a promjene na signalu se mogu izračunati pomoću diferencijacije.



Slika 2.2, maska za prewitt vertical

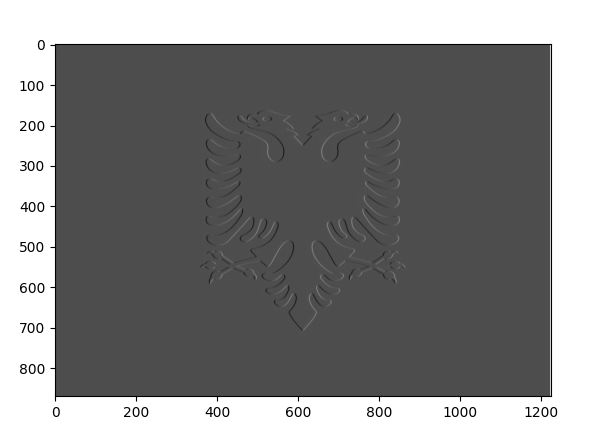
Radi na način kao derivacija prvog reda i računa razliku intenziteta pixela na rubnim dijelovima. Kako je srednji stupac popunjen nulama, on ne sadrži vrijednosti slike nego ih računa kao razliku lijevog i desnog pixela oko tog ruba.



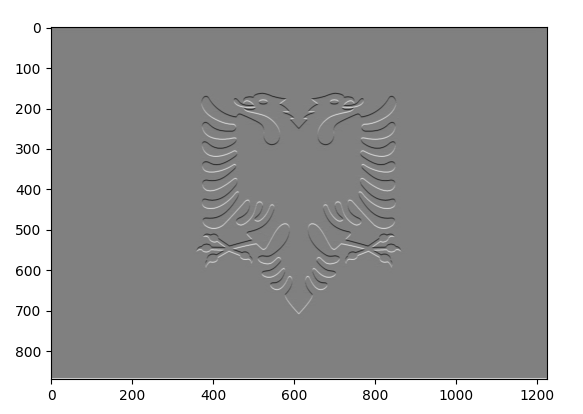
Slika 2.3, maska za prewitt horizontal

Radi na istom principu kao i maska za vertikalne značajke. Kako je srednji red ispunjen nulama, on nema vrijednosti slike nego ih računa kao razliku gornjeg i donjeg intenziteta pixela za određeni rub.

Primjer slika Albanije kada se provuče kroz prewitt vertical i horizontal.



Slika 2.4, slika Albanije provučena kroz prewitt vertical

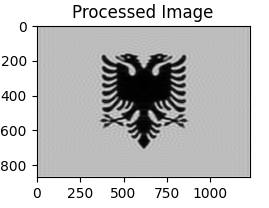


Slika 2.5, slika Albanije provučena kroz prewitt horizontal

### Fourier

Fast Fourirer Transformation se može iskoristiti pri klasifikaciji slika na način da se pixeli pretvore u frekvencije. Kako su pixeli stalni, a ne promjenjivi moguće je koristiti Discrete Fourier Transformation, ali ona je većinom prespora kako bi se koristila pa je zato bolja opcija Fast Fourier Transformation. Njom je moguće manipulirati spektar izlaznih vrijednosti za procesiranje slika. To se odvija dodavanjem filtera koji mogu propuštati niske frekvencije, visoke, itd.

Primjer slike Albanije nakon FFT (Fast Fourier Transformation) s filterom koji propušta niske frekvencije.

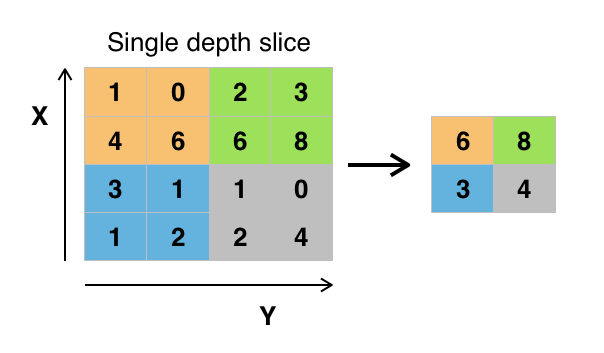


Slika 2.6, Low pass filter FFT slika Albanije

## Max Pooling

Kako su slike u izvornom formatu prevelike jer bi imali previše značajki, prvo ih je potrebno smanjiti. Slike su smanjene na format 20x16, a zatim i tada može doći do problema prevelikog broja značajki kod većih skupova podataka pa se izvodi polling.

Pooling služi kako bi smanjio dimenzije danih podataka. Njime se šalje kernel proizvoljne veličine koji ide po originalnoj matrici slike i odabire najveću značajku u tom području. U našem slučaju da nismo proveli pooling na slici od 20x16, imali bi 320 značajki, a nakon što smo proveli pooling s kernelom (2,2) smanjilo se na 80 značajki po slici. Time smo sačuvali najznačajnije podatke iz svake regije matrice.



Slika 2.7, prikaz smanjenja dimenzije matrice poolingom

Nakon što se izvede pooling nad podacima tada je potrebno smanjiti matricu na jednu dimenziju. Nakon što se slika smanji na samo jedan vektor značajki moguće je napraviti csv file koji će se moći uploadati na Azure za kreiranje modela.

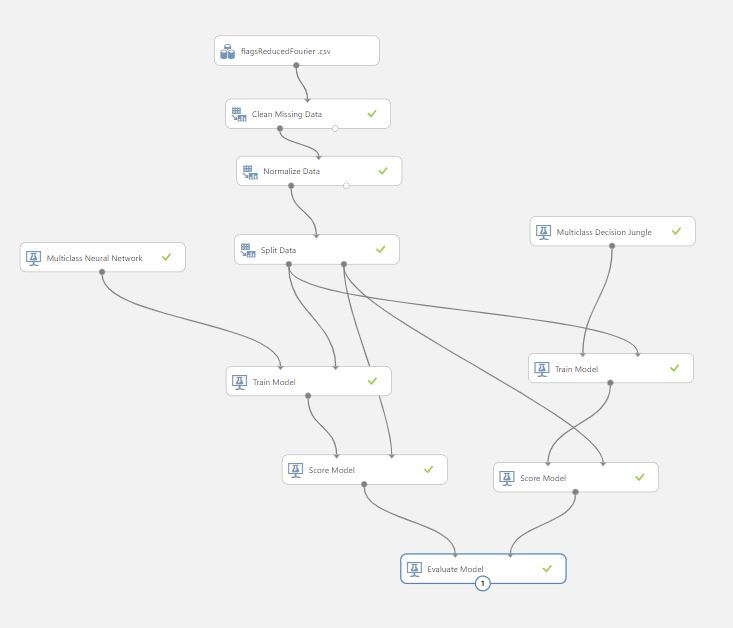
## Klasifikatori i Azure modeli

Kako postoji mnoštvo dostupnih klasifikatora, potrebno je vidjeti koji će davati najbolje rezultate za odabrani skup podataka. U našem slučaju smo odabrali dva klasifikatora. Multiclass Neural Network i Multiclass Decision Jungle. Svaki od klasifikatora testiran je na svakom od skupova podataka na načine koji su objašnjeni u prethodnom potpoglavlju.

Klasifikatori služe kako bi predviđali naziv klase iz danih podataka. Postoje Binarni, Multi-Class, Multi-Label i Imbalanced, a u našem slučaju bili su potrebni Multi-Class. Svaki skup podataka podijeljen je 0.75:0.25 na trening:test.

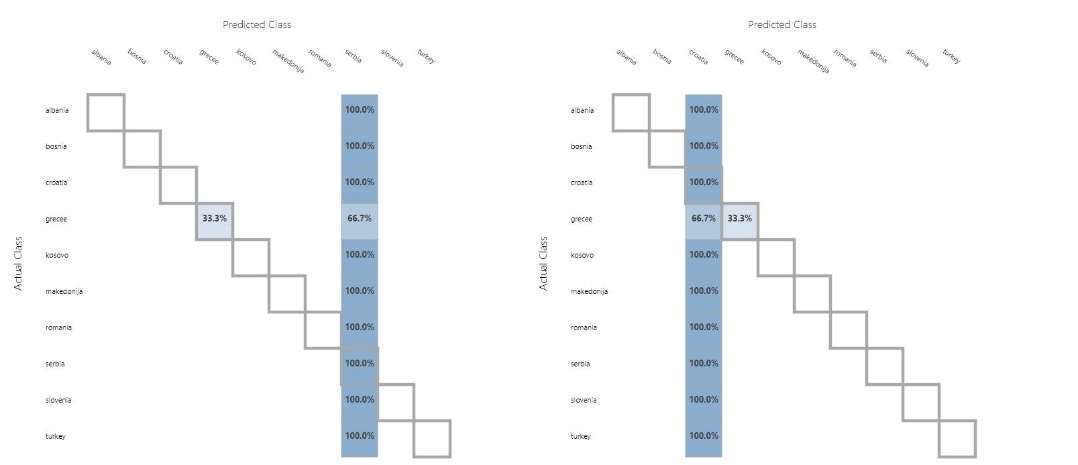
### Fourier skup podataka i model

Nakon što se napravio csv i uploadao na Azure moguće je kreirati novi eksperiment s kojim se može provjeriti točnost algoritma za predviđanje klasa s obzirom na skup podataka koji je generiran FFT-om.



Slika 2.8, prikaz Azure modela s podacima dobivenim FFT-om

Kao što je vidljivo na slici iznad, korištena su dva klasifikatora. Dodan je Clean Missing Data u slučaju da se dogodi neka pogreška pri kreiranju podataka. Nakon što se pokrenuo model moguće je usporediti dva klasifikatora za zadani skup podataka.



Slika 2.9, evaluacija predviđanja podataka dobivenih FFT-om

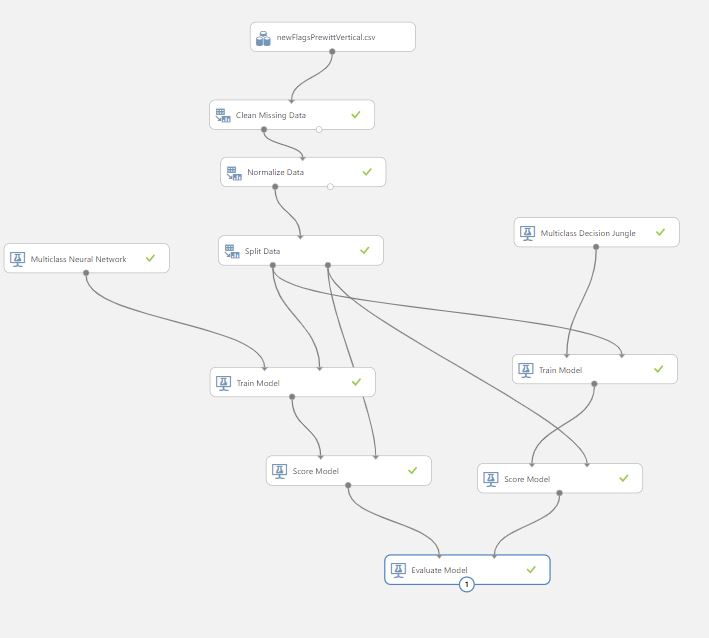
Kao što je vidljivo iz slike iznad da je rijetkost gdje je nešto pogođeno. Što znači da nije dobar postupak za dobivanje rubnih značajki u ovome skupu podataka.



Slika 2.10, numerički prikaz točnosti predviđanja podataka dobivenih FFT-om

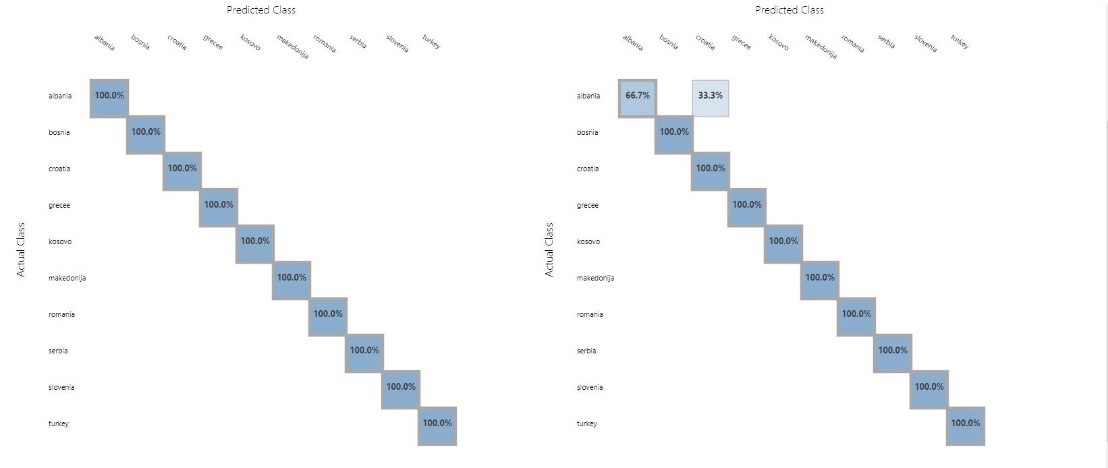
### Prewitt vertical skup podataka i model

Kao što je opisano i za Fourierov skup podataka, isto je potrebno napraviti i za ostale.



Slika 2.11, prikaz Azure modela s podacima dobivenim Prewitt verticalom

Korišteni su isti klasifikatori kao i za Fourier samo se razlikuje skup podataka koji je predan modelu.



Slika 2.12, evaluacija predviđanja podataka dobivenih Prewitt verticalom

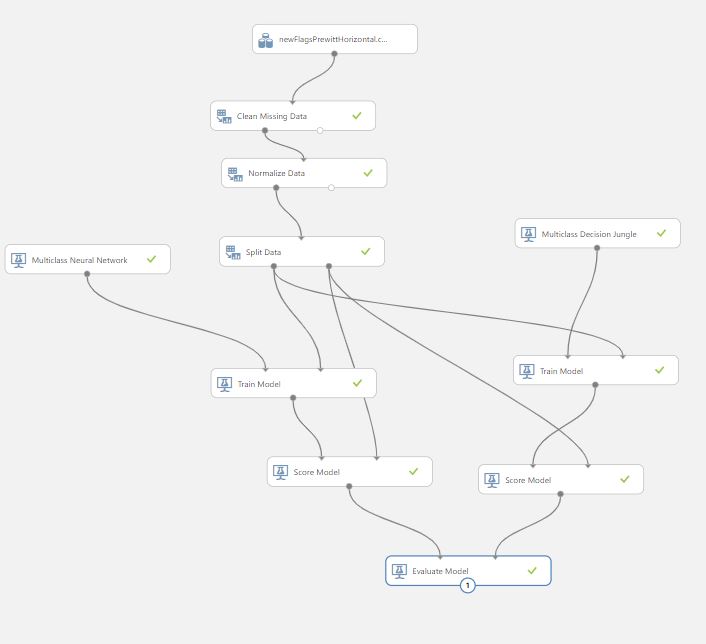


Slika 2.13, numerički prikaz točnosti predviđanja podataka dobivenih Prewitt verticalom

Na slikama iznad vidljivo je kako s ovim skupom podataka točnost predviđanja se drastično povećala. Zbog ovakvih rezlultata predviđanja s neural networkom, ovaj način je odabran kao glavni model koji će koristiti aplikacija ovoga seminara.

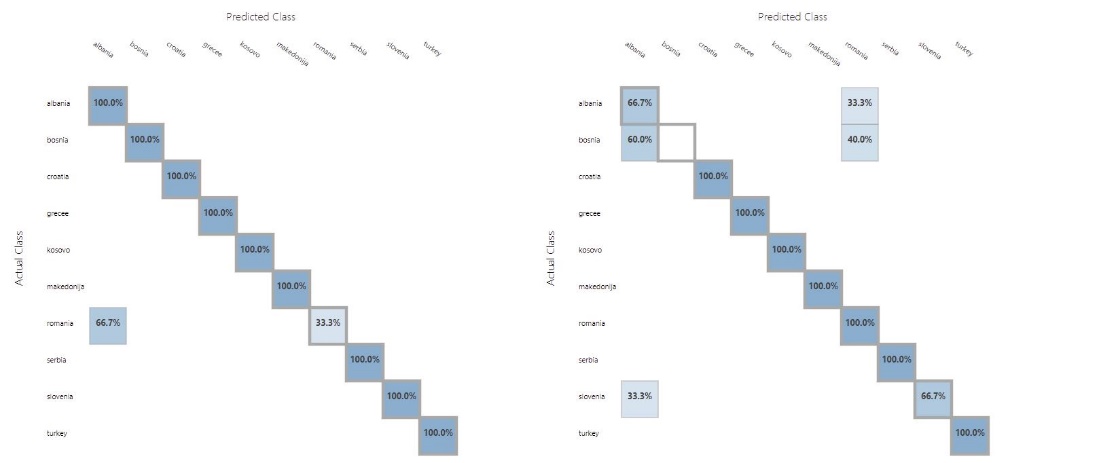
### Prewitt horizontal skup podataka i model

Kao što je objašnjeno i u prijašnjim potpoglavljima, samo je promijenjen skup podataka.



Slika 2.15, prikaz Azure modela s podacima dobivenim Prewitt horizontalom

Pokretanjem modela dobiju se rezultati predviđanja.



Slika 2.16, evaluacija predviđanja podataka dobivenih Prewitt horizontalom



Slika 2.17, numerički prikaz točnosti predviđanja podataka dobivenih Prewitt horizontalom

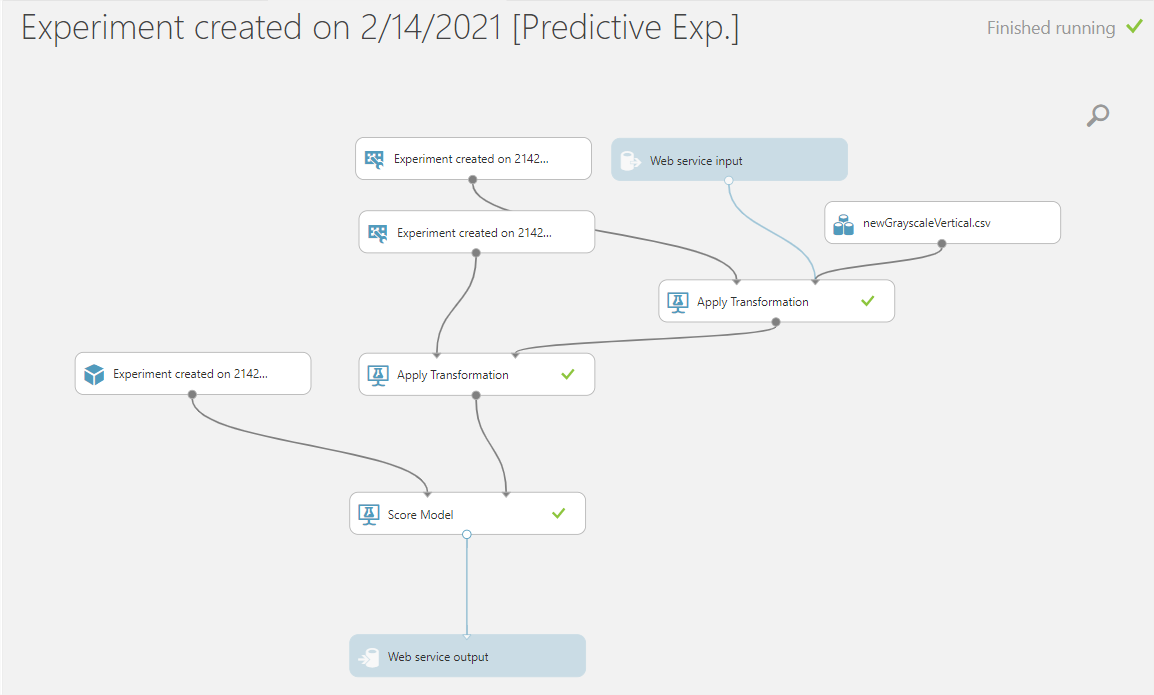
Vidljivo je kako je postupak dao bolje rezultate nego FFT, ali malo lošije nego Prewitt vertical. Što znači da je Prewitt vertical dao najbolje rezultate te je zato i odabran kao predstavnik.

# Programsko rješenje

U ovom poglavlju opisat će se programsko rješenje prepoznavanja zastava.

## 3.1 Azure ML

Nakon što se obradila detaljna analiza prethodno navedenih modela, izabrani model se implementira kao web usluga. Azure ML unutar sebe nudi tu opciju te je bilo potrebno samo uzeti predefinirane postavke.



*3.1 Implementiranje modela kao web usluge*

Nakon što je model implementiran kao web usluga dobivamo API key te url koji omogućuju komunikaciju s modelom.

## 3.2 Python

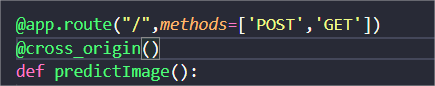
Kako je ranije spomenuto, za kreiranje skripti korišten je programski jezik Python. Python je korišten prilikom kreiranja datasata kako bi se slike smanjile i kako bi se izvukle značajke, no također je korišten kao server. Korišten je kao server, odnosno u ovome slučaju kao posrednik u komunikaciji između Azura i React-a. Razlog tomu je što web preglednici ne dozvoljavaju pokretanje python skripti te se zbog toga nije mogla izvršiti direktna komunikacija između Azure ML-a i React-a. Server je kreiran pomoću flaska koji je trenutno jedan od najpopularnijih mikrookvira.

Nakon instalacije paketa kreiran je jedan endpoint za POST zahtjev, a način na koji se kreira je prikazan slikom 3.2.



*3.2 Prikaz korištenog endpointa*

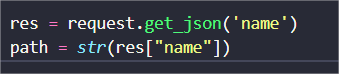
Nakon toga se definira funkcija koja se izvršava pod tim endpointom.



*3.3 Definiranje funkcije*

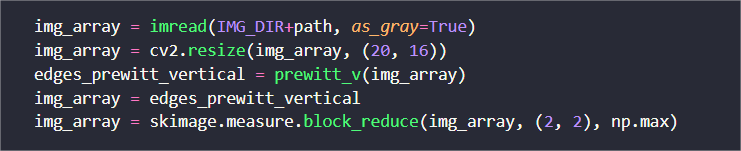
POST zahtjev se kreira jer klijentska strana šalje putanju slike za koju je potrebno izvršiti predikciju.

Obrada tijela POST zahtjeva izvršava se na način prikazan slikom 3.4.



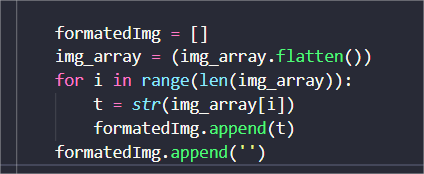
*3.4 Obrada tijela POST zahtjeva.*

Klijentska aplikacija i server su postavljeni lokalno te je zbog toga moguće proslijediti putanju slike na navedeni način. Klijentska aplikacija šalje putanju slike nad kojom se nisu provodile nikakve metode. Zbog toga se nad njom provode četiri metode unutar POST zahtjeva. Prvo se slika pretvara u sive tonove te se zatim skalira, a na kraju se nad njom izvršava prewit vertical i block reduce koji su objašnjeni u poglavlju 2.2.



*3.5 Metode koje se provode nad slikom*

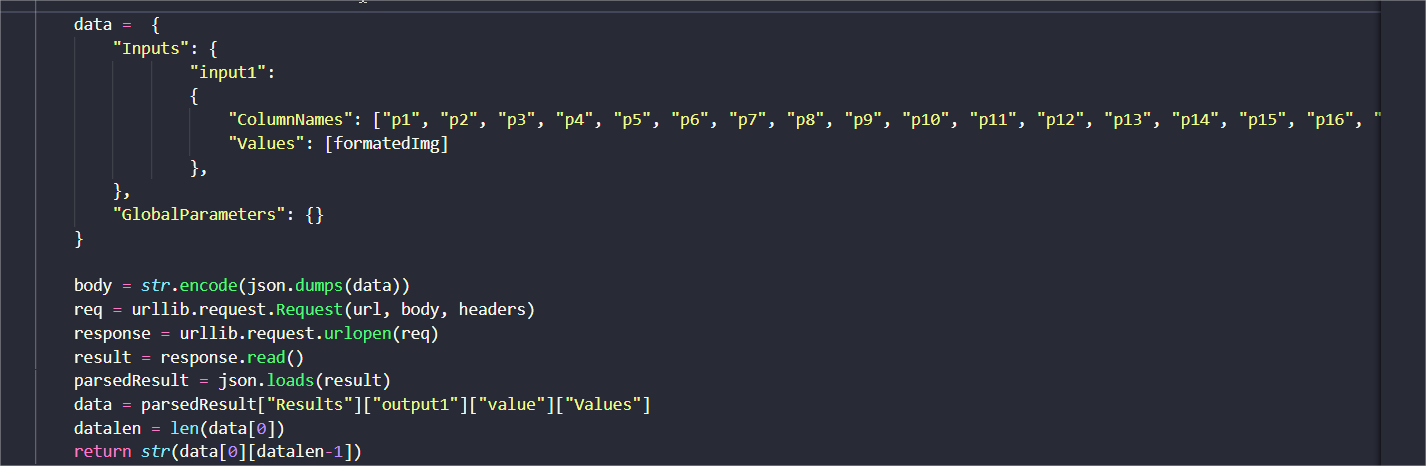
Nakon izvršavanja tih metoda dobiveno je 1-d polje brojeva koj se pretvara u polje stringova kako bi Azure ML mogao obraditi zahtjev.



*3.6 Formatiranje podataka*

Unutar “formatedImg“ se nalazi 81 podatak od kojih 80 čine značajke slike, a posljednji je prazan string jer predstavlja ime države koje još ne znamo.

Nakon toga je kopiran kod sa Azura kojim se web servisu šalje zahtjev te se dobiva odgovor. Odgovor se zatim parsira u JSON format te se vraća zadnji podatak koji predstavlja predviđenu državu.



*3.7 Slanje zahtjeva za predviđanje države*

## 3.3 React

React, biblioteka koja se koristi za razvoj web aplikacija predstavlja klijentsku stranu problemskog rješenja. Aplikacija koristi input pomoću kojega se učitava file te gumb pomoću kojega se šalje POST zahtjev. Na slici 3.8 prikazan je izgled aplikacije nakon učitavanja zastave.



*3.8 Izgled web aplikacije*

Komunikacija se izvršava pomoću axios biblioteke jer omogućuje jednostavnije rukovanje podacima i greškama.



*3.9 Slanje POST zahtjeva te dobivanje odgovora*

Odgovor se sprema u stanje koje se zatim prikazuje korisniku, a to vidimo na slici 3.10.



*3.10 Izgled aplikacije nakon predviđanja zastave*

**Aplikacija s korištenim datasetovima nalazi se na githubu: https://github.com/Sven2219/Flag-Recognizer.**

# 4. Zaključak

Ovim radom stvoren je model koji predviđa zastave Balkana. Model predviđa koja bi to klasa (u ovom slučaju zastava) mogla biti na temelju ulaznih podataka, a u pozadini koristi klasifikaciju pomoću machine learninga. Kako je prikazano u drugom poglavlju, postoji mnoštvo načina za odabir značajki na slikama, a tako i postupaka klasifikacije. Od tri provedena načina odabira značajki, Fourier se istaknuo kao najlošiji za ovaj skup podataka. S druge strane, Prewitt vertical je briljirao gdje je u slučaju klasifikacije s Neural Networkom imao čak 100%. Prewitt horizontal je bio za nijansu lošiji od verticala.Time se došlo do zaključka da je na ovom skupu podataka bilo najbolje dohvaćanje rubnih značajki slike preko Prewitt verticala.

Tijekom klasifikacije raznih podatkovnih skupova iskazao se Neural Network kao bolji klasifikator. Iz tog razloga se on koristio zajedno s Prewitt vertical podatkovnim skupom za izradu web aplikacije.

Machine learning omogućuje mnoštvo zanimljivih načina klasifikacije i rješavanja problema. Pomoću ML-a moguće je napraviti „pametne“ stvari koje bi na temelju svakog novog podatka učile i svaki puta bile sve bolje i bolje. Uz puno truda i napora moguće je stvoriti odlične aplikacije popraćene ML-om, ali potrebno je odrediti problem koji se rješava i pravilno mu pristupiti.LITERATURA

[1] Aishwarya Singh, 29.8.2019., *3 Beginner-Friendly Techniques to Extract Features from Image Data using Python,* https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/

[2] *Prewitt Operator, https://www.tutorialspoint.com/dip/prewitt\_operator.htm*

[3] Dan Nelson, *Image Recognition in Python with TensorFlow and Keras*, https://stackabuse.com/image-recognition-in-python-with-tensorflow-and-keras/

[4] Craig Chen, 17.2.2020., *Digital Image Processing using Fourier Transform in Python,* https://hicraigchen.medium.com/digital-image-processing-using-fourier-transform-in-python-bcb49424fd82