

Klasifikacija povrća

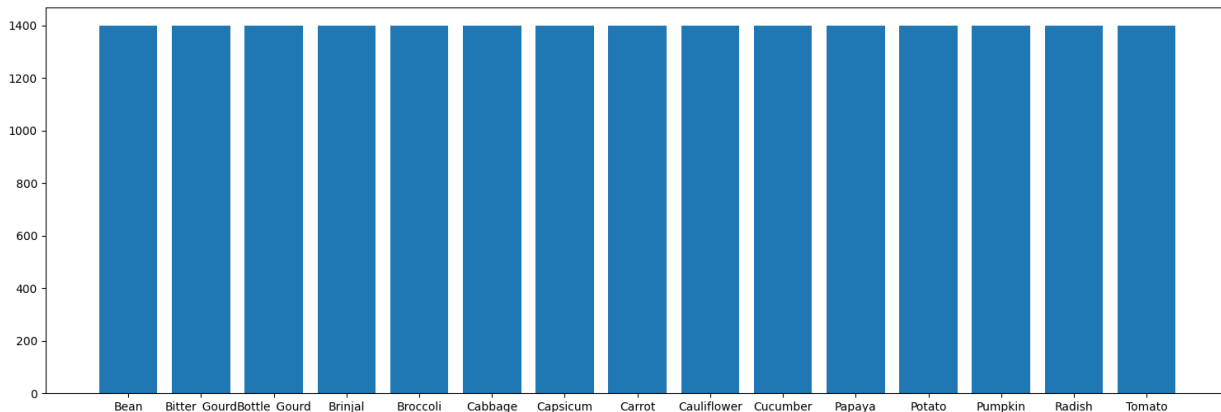
-Projekat iz neuralnih mreža-

Problem:

Klasifikacija 15 vrsta povrća koje se nalaze u [javnom datasetu](#) koji smo koristili pri rešavanju problema. Dataset uključuje 15 vrsta povrća (klasa za klasifikaciju): *Bean, Bitter Gourd, Bottle Gourd, Brinjal Broccoli, Cabbage, Capsicum, Carrot, Cauliflower, Cucumber, Papaya, Potato, Pumpkin, Radish i Tomato*.

Ulazni podaci za rešavanje problema uključuju ukupno 21000 slika raspoređenih u 15 kategorija, svaka sa 1400 slika rezolucije 224x224 u *.jpg formatu.

Grafik odabiraka klasa:



Podaci nad kojim smo radili su potpuno balansirani, jer svaka klasa ima tačno onoliko odabiraka koliko i bilo koja druga klasa. Pošto su sve klase podjednako zastupljene, nismo ni jednu odbacili.

Primer podataka iz svake klase:



Podela podataka na skupove:

Dataset je već podeljen na 15000 slika za trening, 3000 slika za testiranje i 3000 slika za validaciju. To predstavlja 70%, 15% i 15% celokupnog dataseta, respektivno.

Podela na distinktnne skupove je bitna, jer time izbegavamo preobučavanje i omogućavamo direktan vid provere (validacijom) u uspešnost modela.

Predprocesiranje podataka:

Stavke koje smo sprovedi za predprocesiranje podataka uključuju dodavanje pseudo-nasumičnih varijacija u slikama dataseta u vidu:

- Rotacije ili prevrtanja
- Zumiranja
- Dodatnog pozitivnog/negativnog kontrasta i

Takođe je izvršena normalizacija RGB vrednosti svih piksela slika (skaliranje na vrednosti između 0 i 1).

Diskusija o arhitekturi mreže, kriterijumima i funkcijama aktivacije:

Korišćena kriterijumska funkcija (funkcija gubitka) je **Sparse Categorical Crossentropy** koja daje najbolje rezultate za slučajeve kada treniramo multi-klasni model čiji su izlazi intidžerske vrednosti koji predstavljaju pripadnost određenoj klasi (ovim takođe izbegavamo dodatni one-hot encoding korak u postprocesuiranju). Izabrali smo ovaj način penalizovanja jer odlično odgovara našim potrebama. Optimizovali smo po preciznosti jer želimo da maksimizujemo broj korektnih predikcija.

Za metodu optimizacije kriterijumske funkcije smo koristili Adam zbog toga što:

- Adam prilagođava stope učenja za svaki parametar pojedinačno na osnovu njihovih istorijskih gradijenata. Ova prilagodljivost može biti korisna za obuku na podacima sa različitim karakteristikama i može dovesti do brže konvergencije.
- Adam često mnogo brže konvergira od tradicionalnih optimizacionih algoritama, posebno prilikom rada sa ogromnim skupovima podataka i visoko-dimenzionim parametarskim prostorima.

Za aktivacione funkcije smo svugde birali **Rectified Linear Unit (ReLU)**. U svim skrivenim slojevima smo je koristili jer je najbolja u našem slučaju iz više razloga:

1. Jednostavnost - jednostavnost funkcije čini je računarski efikasnom za izračunavanje i lakom za implementaciju u arhitekturama neuralnih mreža.
2. Nelinearnost - ReLU uvodi nelinearnost u model, što je ključno za sposobnost učenja neuralnih mreža.

3. Izbegavanje problema nestajanja gradijenta (Vanishing Gradient Problem)

- ReLU pomaže u ublažavanju problema nestajanja gradijenta koji se može pojaviti sa funkcijama aktivacije kao što su sigmoid ili tanh.

4. Efikasno treniranje - kombinovanjem jednostavnosti i nelinearnosti ReLU-a često dovodi do bržeg konvergiranja rezultata tokom treniranja u poređenju sa drugim funkcijama.

U izlaznom sloju smo koristili **Softmax** funkciju aktivacije jer:

CNN često koriste softmax funkciju aktivacije u izlaznom sloju za višeklasnu klasifikaciju. Funkcija softmax pretvara sirove rezultate u distribucije verovatnoće po klasama.

Arhitektura modela i broj parametara:

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 224, 224, 3)	0
rescaling (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 16)	448
max_pooling2d (MaxPooling2D)	(None, 112, 112, 16)	0
conv2d_1 (Conv2D)	(None, 112, 112, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 32)	0
conv2d_2 (Conv2D)	(None, 56, 56, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_4 (Conv2D)	(None, 14, 14, 256)	295168
dropout (Dropout)	(None, 14, 14, 256)	0
flatten (Flatten)	(None, 50176)	0
dense (Dense)	(None, 512)	25698624
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 15)	3855

```
=====
Total params: 26218415 (100.02 MB)
Trainable params: 26218415 (100.02 MB)
Non-trainable params: 0 (0.00 Byte)
=====
```

Preobučavanje:

Preobučavanje se dešava kada model ne može da generalizuje i umesto toga se previše uklapa u skup podataka za obuku. Preobučavanje se dešava iz nekoliko razloga, kao što su:

- Veličina podataka za obuku je premala i ne sadrži dovoljno uzoraka podataka da bi tačno predstavili sve moguće vrednosti ulaznih podataka
 - Podaci o obuci sadrže velike količine irelevantnih informacija
 - Model se obučava predugo na jednom skupu uzoraka podataka
- (Preobučavanje se može primetiti tako što se training loss smanjuje, a validation loss se povećava)

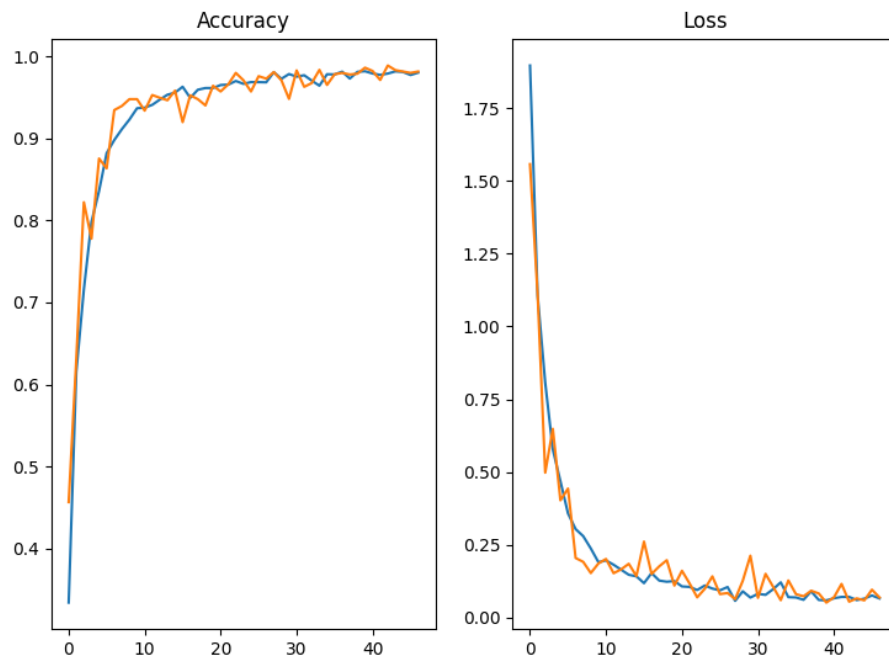
Tehnika koju smo mi primenili za zaštitu od preobučavanja je Early Stopping.

Hiperparametar:

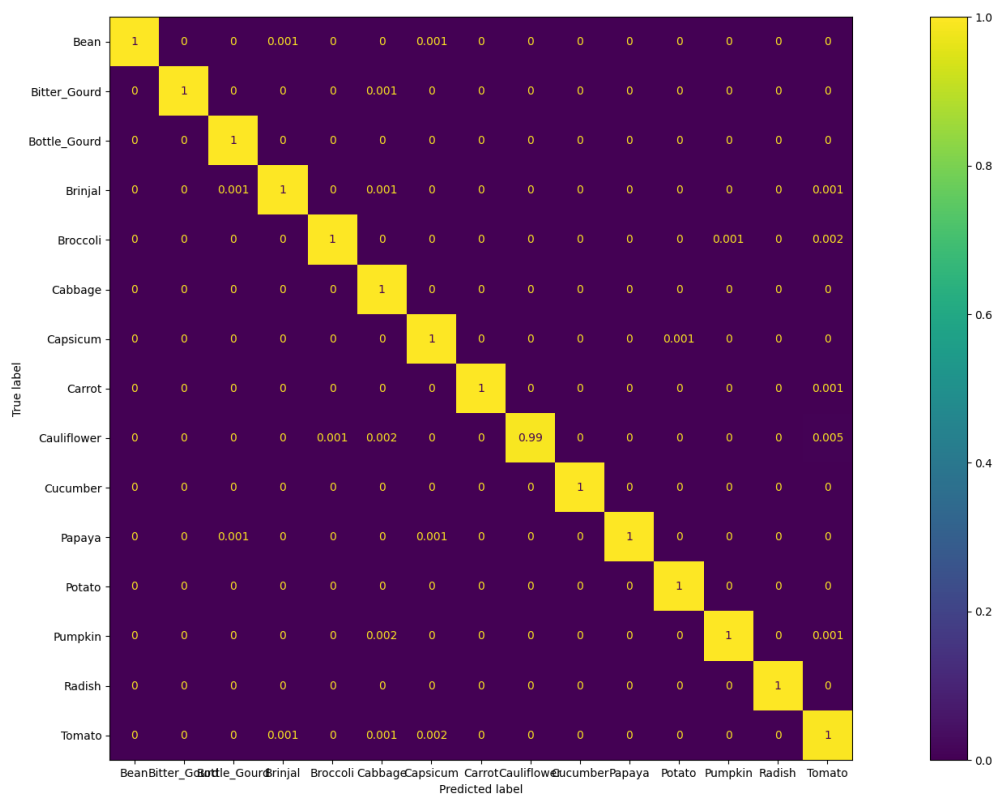
Izabrali smo konstantu obučavanja i našli njenu optimalnu vrednost.

Finalno obučen model:

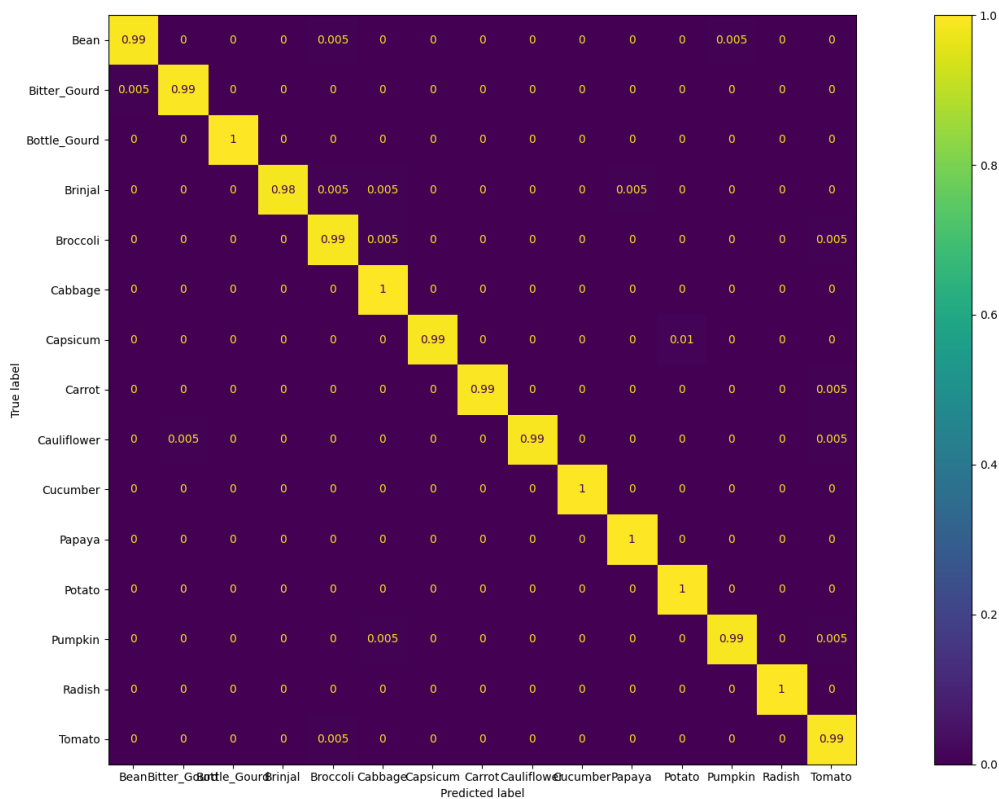
- Grafik performanse



- Matrica konfuzije na trening skupu:



- Matrica konfuzije na test skupu:



- Primeri dobro klasifikovanih primera dataset-a.



- Primeri loše klasifikovanih primera dataset-a.

P - predicted

R - Real

P: Broccoli
R: Tomato



P: Tomato
R: Broccoli



P: Tomato
R: Pumpkin



P: Potato
R: Capsicum



P: Tomato
R: Carrot



P: Papaya
R: Brinjal



P: Tomato
R: Cauliflower



P: Broccoli
R: Bean



P: Bean
R: Bitter_Gourd



- Tačnost modela:

Nad trening skupom je: 99.81%

Nad test skupom je: 99.47%