

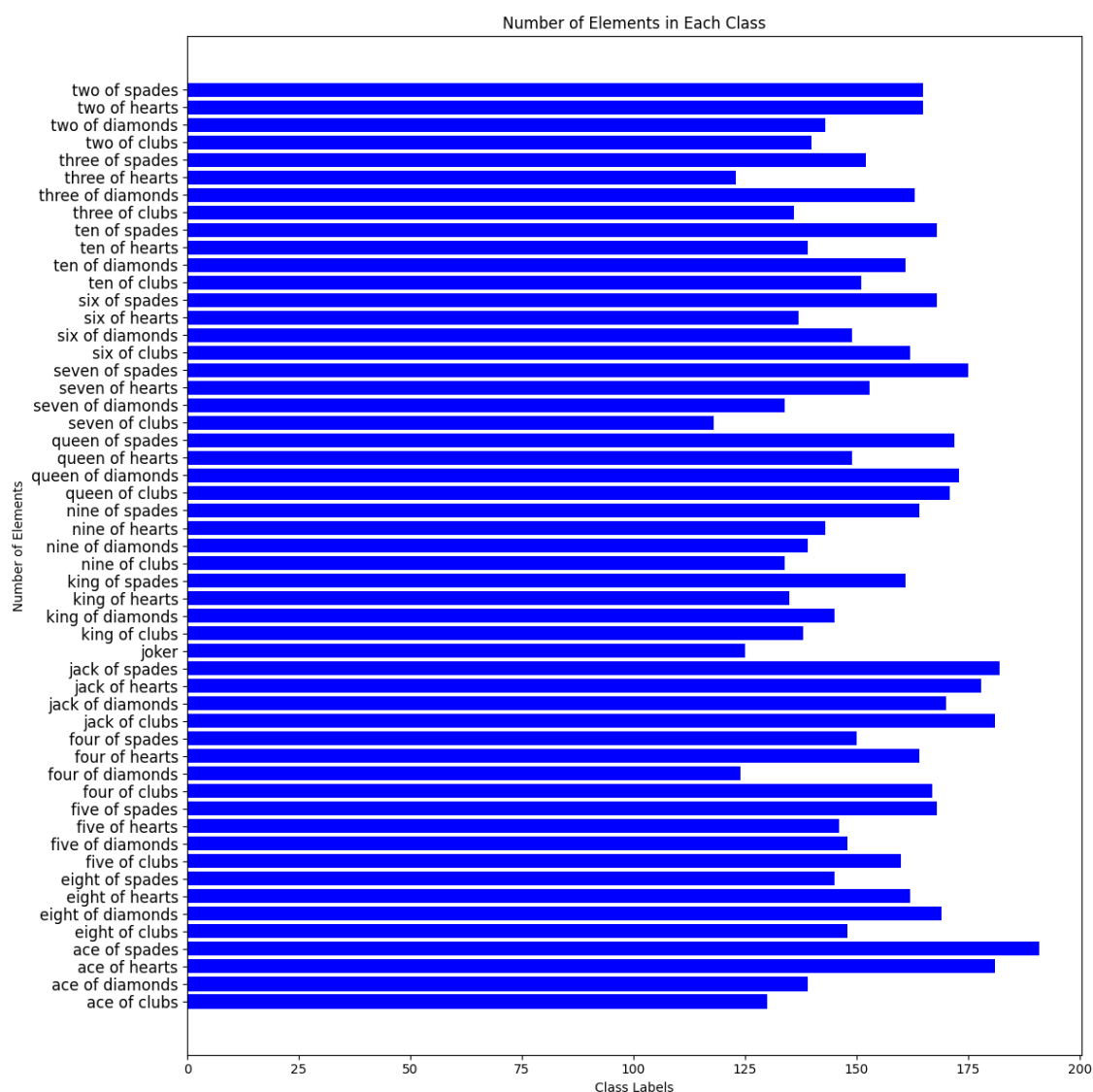
Projektni zadatak iz neuralnih mreža

Klasifikacija karata – izveštaj

Dataset koji je korišćen je: <https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification>

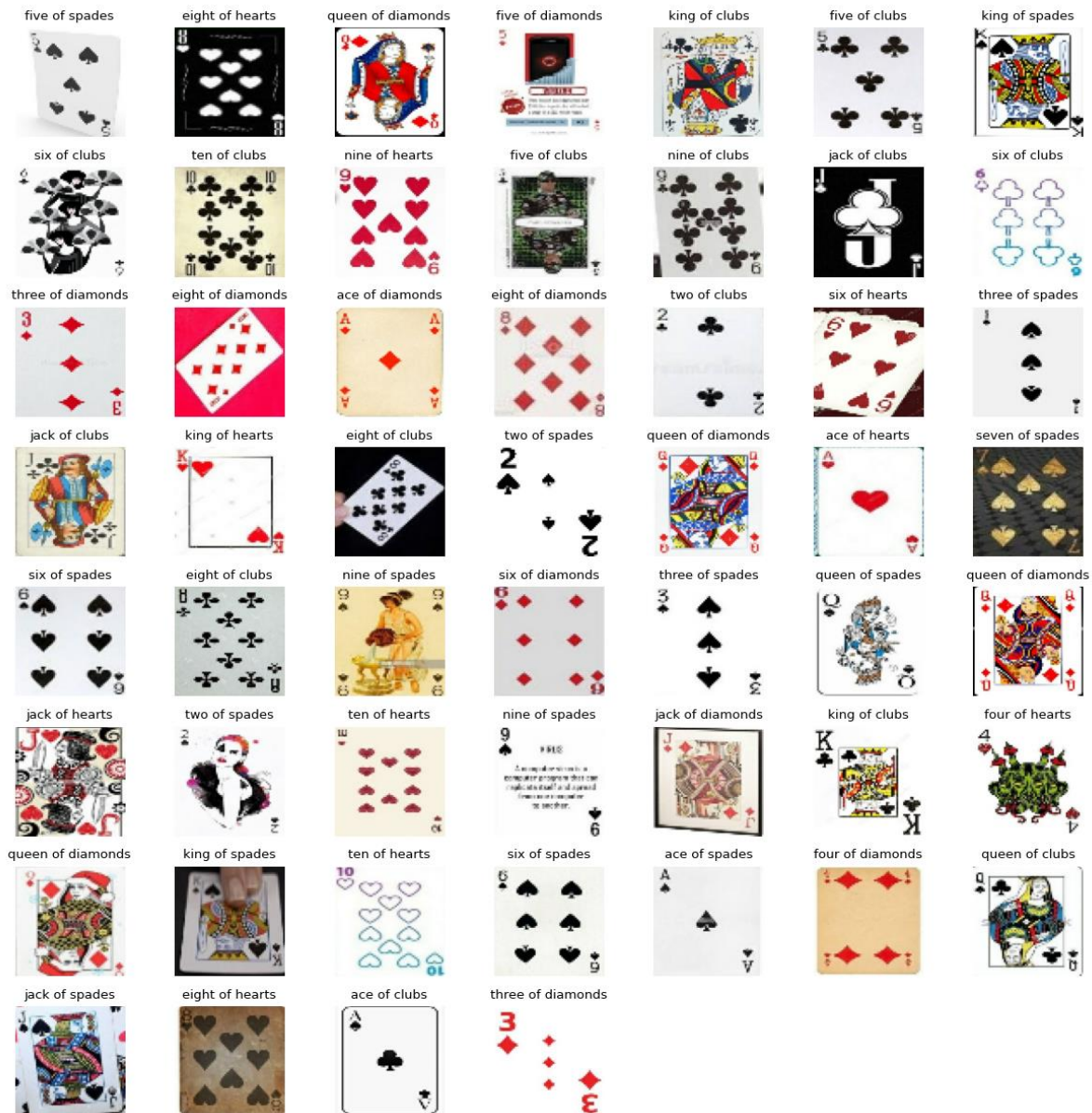
Dataset se sastoji iz 53 klasa. Treba klasifikovati svaku od karata u posebnu klasu.

Broj odbiraka za svaku klasu je prikazan na slici:



Podaci sa svaku klasu su relativno balansirani, ali u slučaju nebalansiranosti, primenili bismo class weights tehniku za razrešavanje.

Primeri podataka za svaku klasu su prikazani na slici:



U folderu je već bio izdelfen skup slika za trening, validaciju i testiranje, međutim, pošto skupovi za validaciju i testiranje nisu bili odgovarajućih veličina, prebacili smo sve podatke u jedan folder „Data“ i nakon toga smo u kodu adekvatno podelili podatke za trening, validaciju i testiranje u odnosu 60-20-20. Podela podataka je bitna da bismo imali dovoljno veliki skup slika za validaciju i testiranje kako ne bi došlo do preobučavanja mreže na trening skupu.

Za predprocesiranje podataka smo koristili augmentaciju podataka tako što smo primenili rotaciju, uvećanje, i translaciju na nasumično odabranim podacima iz dataseta. Skalirali smo sliku sa 224x224x3 na 128x128x3.

Kriterijumska funkcija primenjena u projektu je SparseCategoricalCrossentropy(). Ovu kriterijumsku funkciju smo odabrali zato što je najčešće korišćena u konvolucionim neuralnim mrežama i optimizovana je za veći broj klasa, što One-Hot encoding radi vrlo neefikasno.

Izbor funkcije aktivacije neurona je „ReLU“ zato što se omogućava da mreža uči višestruko brže od mreža koje koriste na primer tanh i sigmoid funkcije.

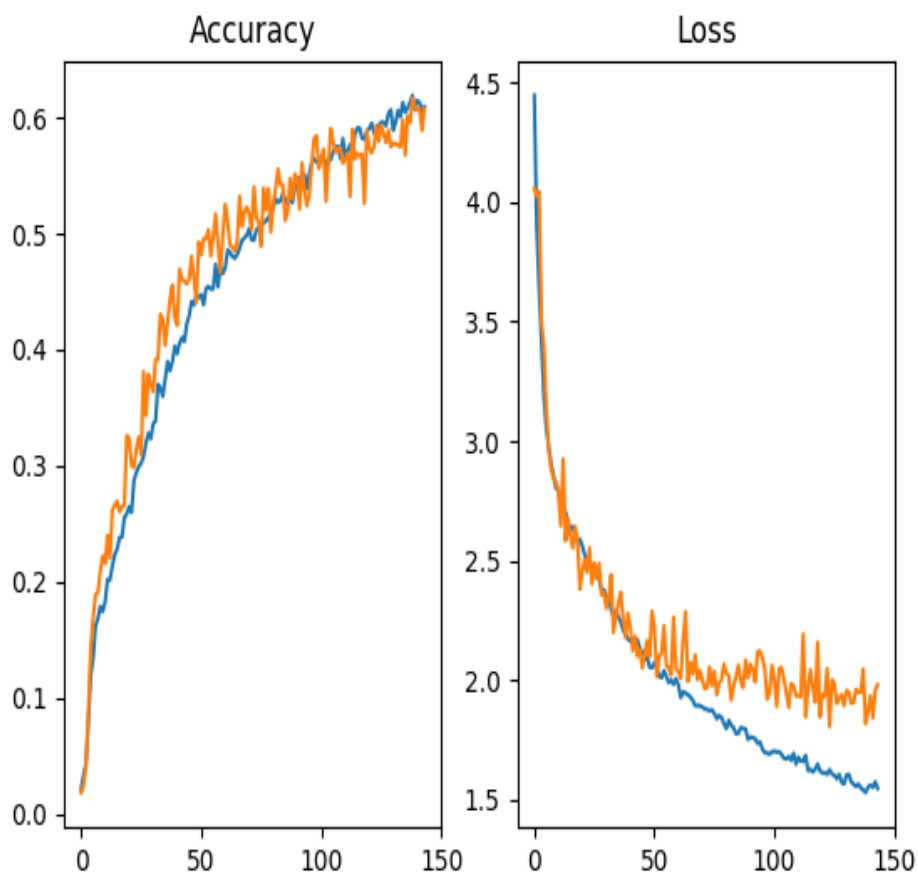
Izbor metode optimizacije kriterijumske funkcije je „Adam“ kome je glavna prednost da dinamički prilagođava konstante obučavanja za svaki parametar mreže. Ovaj algoritam takođe doprinosi bržoj konvergenciji u odnosu na tradicionalne gradient descent algoritme.

U mreži postoji **5 konvolucionih slojeva** sa veličinom filtera 3x3, i sa brojem filtera 16, 32, 64, 128, 256 respektivno. Između svakog konvolucionog sloja, **postoji po jedan MaxPooling2D sloj**. Kako se dimenzije slike ne bi smanjivale prilikom nanošenja konvolucionog sloja, korišćen je padding. Nakon nanošenja svih konvolucionih slojeva iskorišćen je **dropout** radi sprečavanja od preobučavanja. Na samom kraju smo iskoristili flatten da ispravimo 3D izlaz u 1D i povezali ga na **fully connected sloj** sa 256 neurona koji koristi l2 regularizer, i taj sloj je povezan na izlaz mreže čija je aktivaciona funkcija softmax.

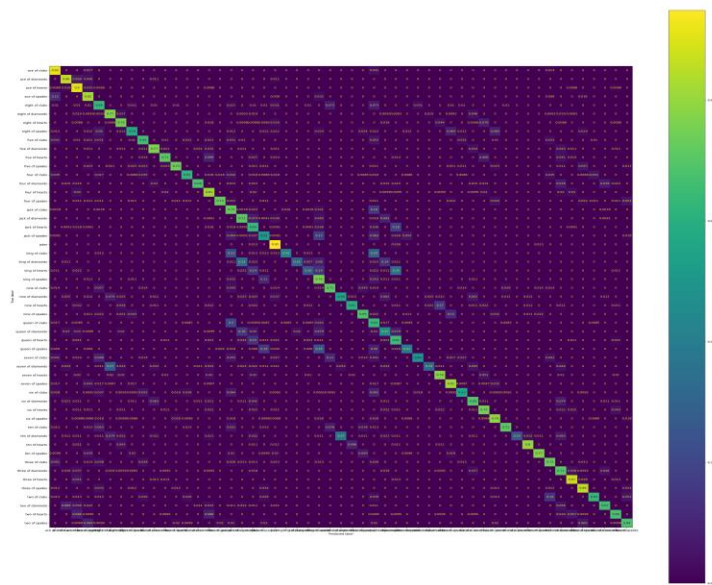
```
model = Sequential([
    data_augmentation,
    layers.Rescaling(1./255, input_shape=(128, 128, 3)),
    layers.Conv2D(filters=16, kernel_size=3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(filters=32, kernel_size=3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(filters=64, kernel_size=3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(filters=128, kernel_size=3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(filters=256, kernel_size=3, padding='same', activation='relu'),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(units=256, activation='relu', kernel_regularizer=l2(regularizer)),
    layers.Dense(num_classes, activation='softmax', kernel_regularizer=l2(regularizer))
])
```

Preobučavanje neuralne mreže nastaje kada su nekom skupu podataka za treniranje odlične, ali na skupu podataka za validaciju lose performanse. Tehnike zaštite od preobučavanja korišćene u ovoj mreži su dropout i l2 regularizacija pokazane na prethodnom primeru i early stopping koji prati validation loss funkciju sa strpljenjem 20.

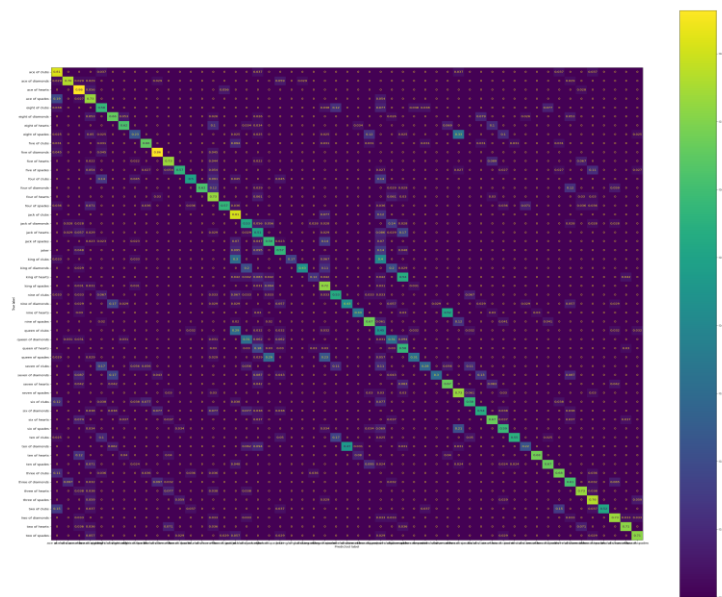
Grafik performanse:



Matrica konfuzije na trening skupu:



Matrica konfuzije na test skupu:



Prikaz nekoliko *dobro* klasifikovanih odbiraka:

king of spades



three of hearts



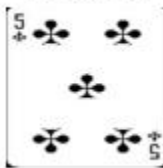
three of clubs



jack of hearts

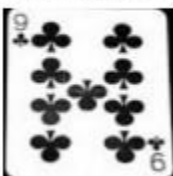


five of clubs



Prikaz nekoliko *lose* klasifikovanih odbiraka:

nine of clubs



jack of spades



seven of diamonds



king of clubs



jack of hearts

