

Elektrotehnički fakultet, Univerzitet u Beogradu

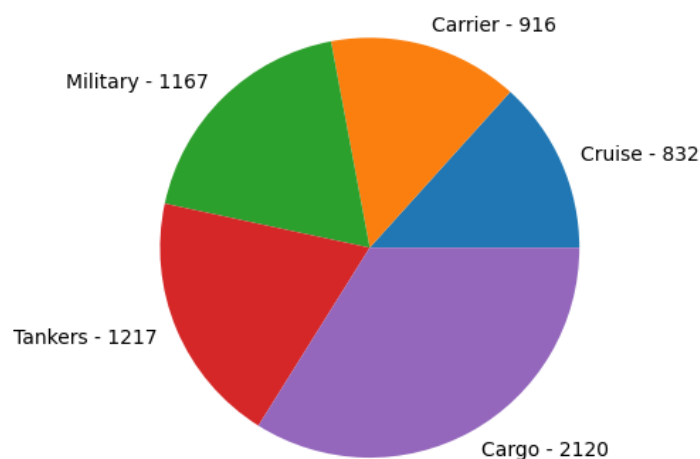
Projektni zadatak iz predmeta Neuralne mreže (13S053NM)

Klasifikacija brodova korišćenjem konvolucionih neuralnih mreža

Luka Radoičić 2020/0085
Dimitrije Jelisavčić 2019/0586

Postavka projekta

Cilj ovog projekta je kreiranje konvolucione neuralne mreže koja može da razlikuje tipove brodova na osnovu slika približnih dimenzija 210x140 px. Ulazni skup podataka čini 6252 slike podeljene u 5 klasa gde svakoj klasi odgovara jedan tip broda. Tipovi su brodovi za prevoz robe (*cargo*), vojni brodovi (*military*), brodovi za prevoz vozila (*carrier*), brodovi za krstarenje (*cruise*) i brodovi za prevoz nafte (*tanker*).



Raspodela slika po klasama

Za balansiranje klasa korišćena je tehnika klasnih težina. Težina klase računata je kao $\text{broj_slika} / (\text{broj_klasa} * \text{broj_slika_klase_i})$ za svaku klasu i .

Učitavanje i podela podataka

Originalni skup podataka ([link](#)) sadrži sve slike u jednom direktorijumu, a podaci o pravilnoj klasifikaciji trening skupa nalaze se u odvojenom fajlu .csv formata. Kako korišćena funkcija iz `keras.utils` biblioteke (`image_dataset_from_directory`) zahteva da su slike raspoređene u podfoldere po klasama, napravili smo dodatni python fajl (`distributor.py`) koji je najpre slike rasporedio u odgovarajuće podfoldere.

Pri učitavanju, slike se dele na trening skup (80%) i validacioni skup (20%) i njihova veličina se skalira na 160x160 px. Ova podela se radi kako bi mreža mogla da se testira tokom obučavanja slikama na kojima nije obučavana radi praćenja performansi modela tokom treniranja i sprečavanja preobučavanja.



Primeri klasa (s leva na desno): cargo, carrier, cruise, military, tanker



Nasumično izabrane slike nakon učitavanja i skaliranja

Proširenje skupa ulaznih podataka

Kako je broj slika kojima se trenira model relativno mali, neophodno je korišćenje tehnika proširivanja skupa podataka za treniranje. Augmentacije koje su izvršene nad slikama su horizontalno obrtanje (*flip*), zumiranje, rotacija, promena osvetljenja slike (*brightness*) i izoštravanje slike (*contrast*).

```

65 data_augmentation = Sequential([
66     layers.RandomFlip('horizontal', input_shape=(img_size[0], img_size[1], 3), seed=2),
67     layers.RandomZoom(0.1, seed=2),
68     layers.RandomRotation(0.1, seed=2),
69     layers.RandomBrightness(0.5, seed=2),
70     layers.RandomContrast(0.25, seed=2)
71 ])

```

Izvršene augmentacije nad ulaznim skupom podataka



Rezultati izvršavanja augmentacija

Treniranje mreže

Mreža se sastoji od naizmeničnih konvolucionih i *max-pooling* slojeva koji se smenjuju 4 puta. Konvolucionni slojevi imaju korak jednak 1 i dodat padding kako ne bi došlo do smanjenja dimenzija. Funkcija aktivacije im je ReLU (*rectified linear unit*), a koristi se zato što je jednostavna za računanje čime se dobijaju bolje performanse. Kriterijumska funkcija je *SparseCategoricalCrossentropy*. Za optimizaciju kriterijumske funkcije korišćen je Adamov optimizator sa konstantom obučavanje od 0,001.

Iza poslednjeg *max-pooling* sloja nalazi se *dropout* sloj radi sprečavanja preobučavanja mreže pa *flatten* sloj za pretvaranje višedimenzionalnog ulaza u jednodimenzionalni ulaz.

Zatim dva potpuno povezana (*fully connected, FC, dense*) sloja čije su aktivacione funkcije ReLU za prvi i *softmax* za drugi FC sloj. *Softmax* FC sloj ima 5 izlaza koji odgovaraju „uverenosti pripadnosti“ svakoj od klasa.

Ukupan broj parametara modela je 1736613 (približno 1,7 miliona parametara).

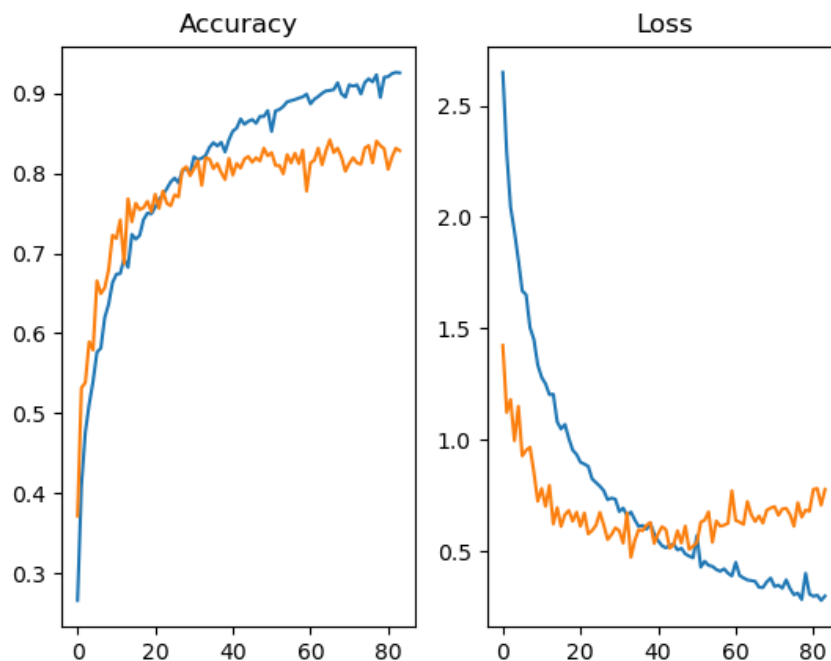
Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 160, 160, 3)	0
rescaling (Rescaling)	(None, 160, 160, 3)	0
conv2d (Conv2D)	(None, 160, 160, 16)	448
max_pooling2d (MaxPooling2D)	(None, 80, 80, 16)	0
conv2d_1 (Conv2D)	(None, 80, 80, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 40, 40, 32)	0
conv2d_2 (Conv2D)	(None, 40, 40, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 20, 20, 64)	0
conv2d_3 (Conv2D)	(None, 20, 20, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 10, 10, 128)	0
dropout (Dropout)	(None, 10, 10, 128)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 128)	1638528
dense_1 (Dense)	(None, 5)	645
Total params: 1736613 (6.62 MB)		
Trainable params: 1736613 (6.62 MB)		
Non-trainable params: 0 (0.00 Byte)		

Preobučavanje je pojava da je mreža previše prilagođena trening skupu, zbog čega se smanjuje sposobnost generalizacije. U izradi ovog projekta korišćene su dve tehnike za sprečavanje preobučavanja, *dropout* i rano zaustavljanje.

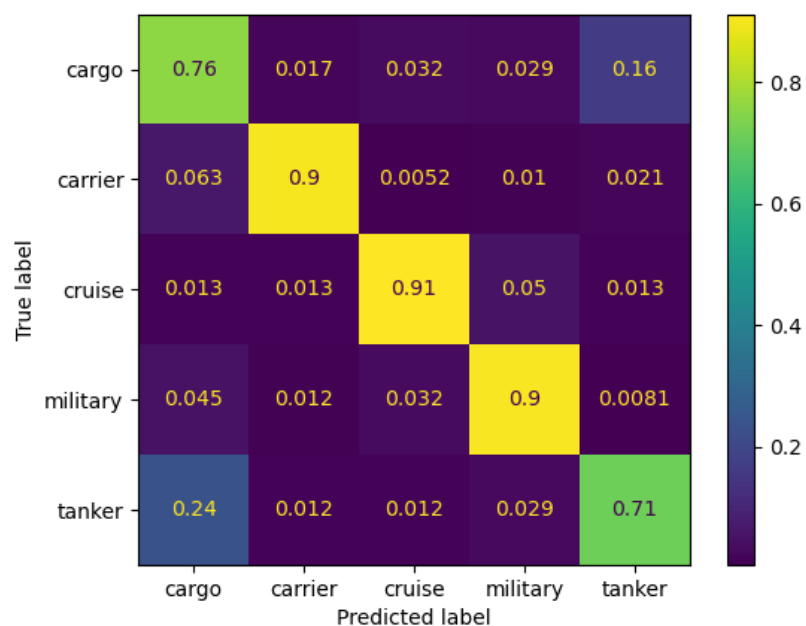
Analiza rezultata

Performanse i rezultat obučavanja mreže prate se kroz *accuracy* i *loss* funkcije po epohama. Plava boja predstavlja performanse skupa za obučavanje, a narandžasta performanse validacionog skupa.

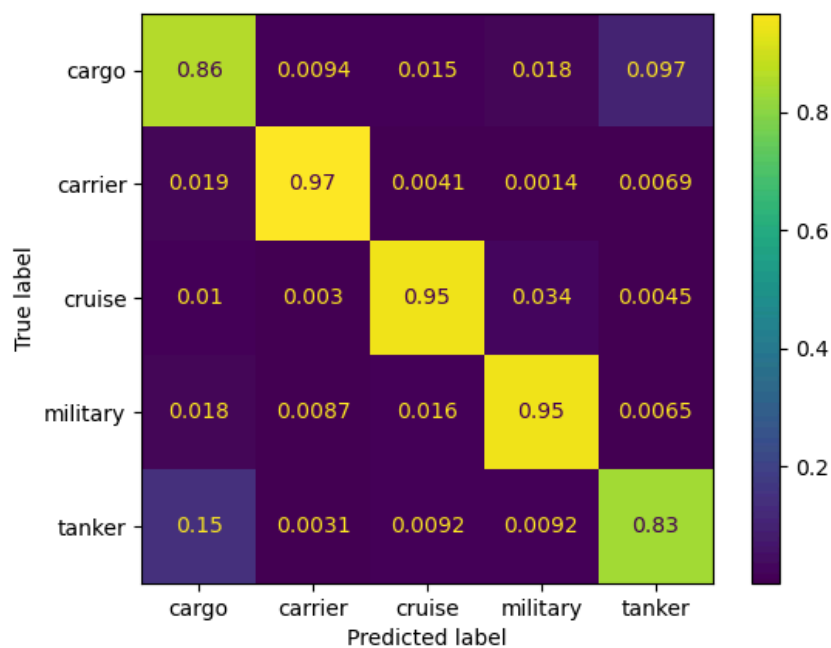
```
79/79 [=====] - 26s 324ms/step - loss: 0.2994 - accuracy: 0.9252 - val_loss: 0.7779 - val_accuracy: 0.8280  
Epoch 84: early stopping  
Accuracy: 81.92%
```



Grafik performanse neuralne mreže kroz epohe obučavanja nad trening i validacionom skupom



Matrica konfuzije nad validacionim skupom



Matrica konfuzije nad trening skupom

Iz matrica konfuzije može se videti da najviše greši u raspoznavanju *tanker* i *cargo* brodova.



Cargo



Tanker

Ovakva greška nije neočekivana jer su ove vrste brodova veoma slične, pogotovo kada su na slici prazni *cargo* brodovi (bez kontejnera) ili kada se kontejneri ne vide.

Kroz veći broj treniranja modela može se videti da su uglavnom najbolje performanse kod prepoznavanja vojnih brodova, što je posledica specifičnih oblika tih brodova kao i karakteristične vojne opreme koja se ne može naći kod drugih klasa. U prikazanom primeru to nije slučaj jer je ovaj model dobro istreniran i za *carrier* i *cruise* brodove.



Primeri vojnih (military) brodova