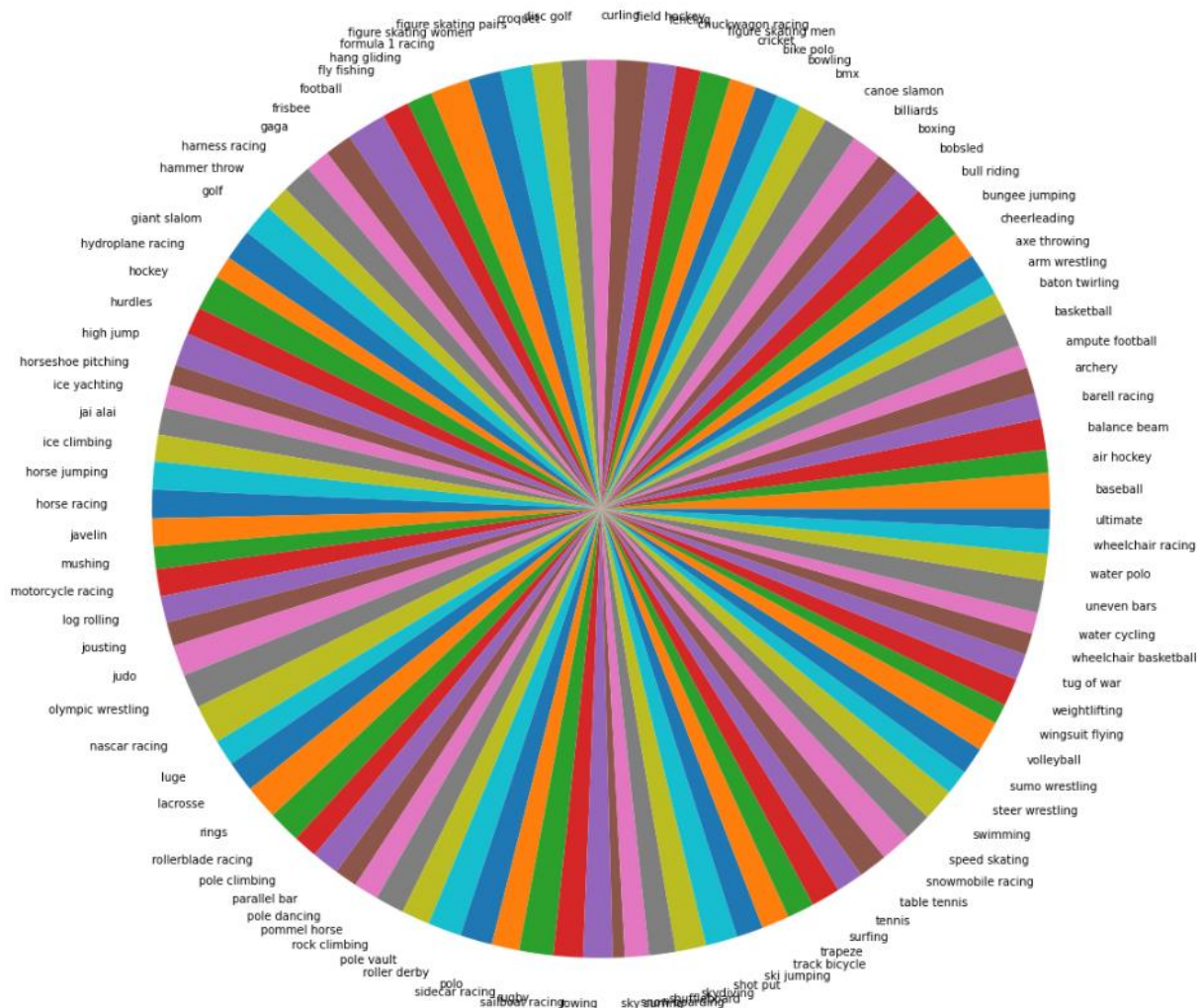


Zadanie 3

Načítal som si dáta z testovacieho súboru pomocou ImageDataGenerator. Dáta obsahovali 100 rôznych druhov športov. V tréningovej množine bolo 13582 obrázkov a vo validačnej a testovacej po 500 obrázkov. Na grafe môžeme vidieť početnosti všetkých druhov športu.



Najviac 191 obrázkov: **formula 1 racing**.



Najmenej 59 obrázkov: **sky surfing**



157: **golf**



172: **hockey**



Predpovede z modelu **VGG16** natrénovaného na ImageNet.

Náhodne som vybral obrázok z každého druhu športu a pomocou funkcie predict a decode_predictions som dostal pravdepodobnosť a názov triedy do ktorej patrí daný šport. V tabuľke je uvedených prvých 25 športov.

	Sport	Top predict	Probability
0	air hockey	turnstile	0.427113
1	ampute football	crutch	0.427832
2	archery	bow	0.991861
3	arm wrestling	mortar	0.716979
4	axe throwing	pedestal	0.404086
5	balance beam	balance_beam	0.76724
6	barell racing	dogsled	0.818608
7	baseball	ballplayer	0.469766
8	basketball	basketball	0.453518
9	baton twirling	tricycle	0.248813
10	bike polo	bicycle-built-for-two	0.776708
11	billiards	pool_table	0.99981
12	bmx	necklace	0.176218
13	bobsled	bobsled	0.569023
14	bowling	puck	0.940999
15	boxing	balance_beam	0.418113
16	bull riding	mountain_bike	0.174793
17	bungee jumping	scuba_diver	0.503864
18	canoe slamon	paddle	0.636479
19	cheerleading	sarong	0.175217
20	chuckwagon racing	dogsled	0.516743
21	cricket	dragonfly	0.487036
22	croquet	croquet_ball	0.796964
23	curling	Irish_wolfhound	0.279434
24	disc golf	unicycle	0.216963
25	fencing	balance_beam	0.883879

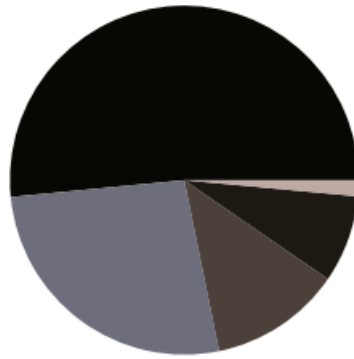
Analýza tried podľa farebnosti pixelov

Podobne som postupoval aj pri analýze farebnosti obrázkov. Kde som si vybral náhodné obrázky z tried a do koláčového grafu som zobrazil 5 farieb ktoré sa vyskytovali najviac na obrázku.

Air hockey



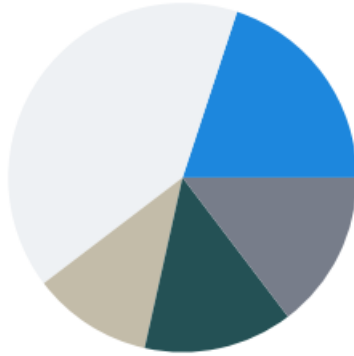
Bmx



axe throwing



Canoe slamon



Balance beam



Crocquet



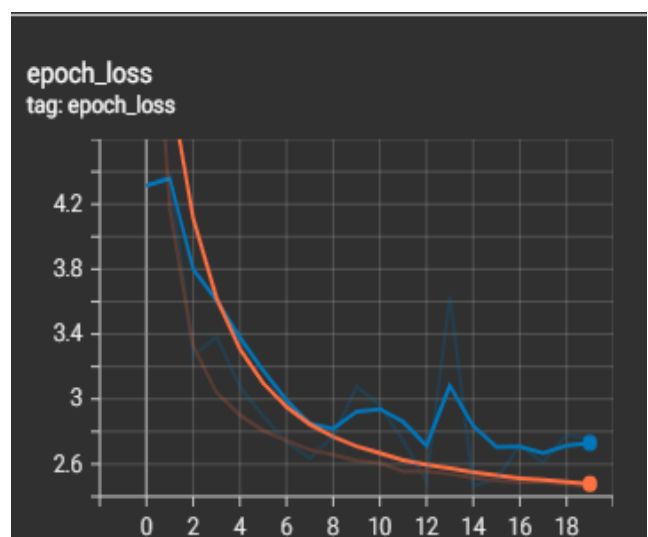
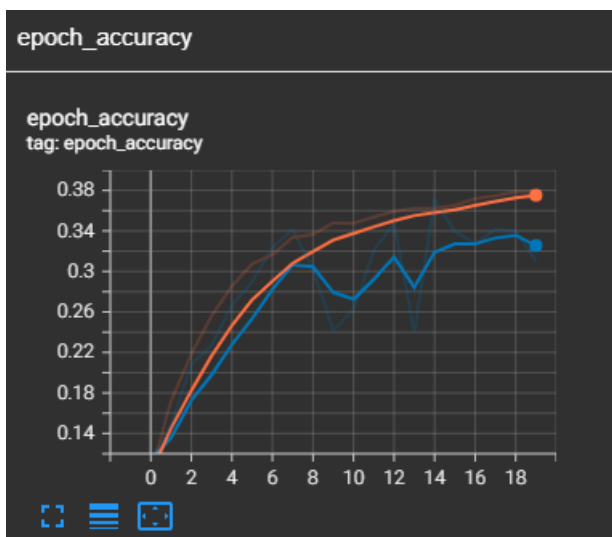
Spracovanie obrázkov pomocou ImageDataGenerator

Augumentoval som si množinu pomocou ImageDataGenerator, kde som nastavil `image_rotation` na 40, `width_shift_range` a `height_shift_range` na 0.2. Rescale na 1/255, pretože obrázky sú vo formáte rgb a chceme ich mať v intervale od 0 do 1. Zoom range na 0,2, Horizontal flip na true. Veľkosť obrázkov som zmenil na 64x64.

Knvolučná sieť

Poradie	Filters	Optimizer	Počet epoch	Počet vrstiev	Úspešnosť
1. tréovanie	32	rmsprop	20	2	41,8%
2. tréovanie	64	adam	20	2	34,2%
3. tréovanie	32	adam	15	3	44,9%

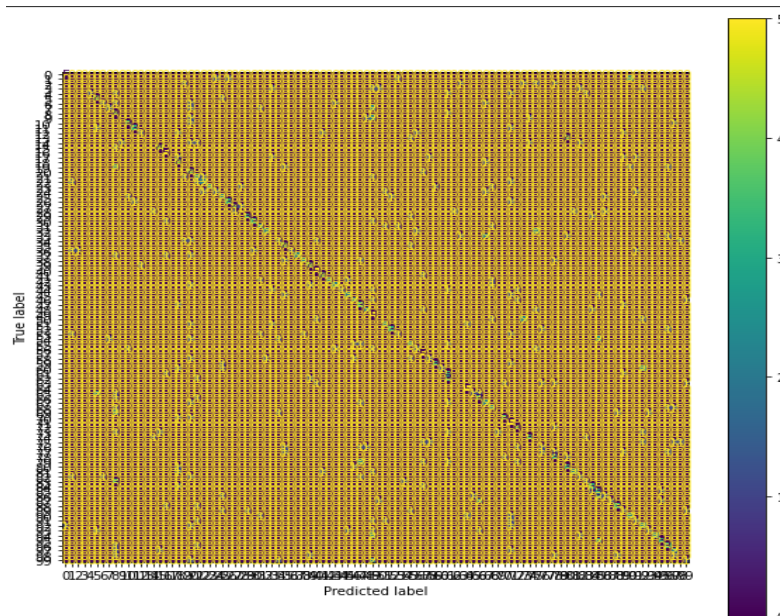
Vytvoril som sieť s dvoma konvolučnými vrstvami s aktivačnou funkciou relu. Filters som nastavil u oboch vrstiev na 32, kernel size na (3,3), `input_shape`(64,64,3). Na výstupnú vrstvu som dal 100 neurónov a aktivačnú funkciu softmax. Chybovú funkciu som nastavil na `categorical_crossentropy` a optimizer na rmsprop. Počet epoch 20. Vytvoril som si taktiež aj ModelCheckpoint aby som si ukladal najlepší výsledok tréovania a taktiež Early stopping som nastavil na 4 epochy. V callbackoch som taktiež použil aj tensorboard na zobrazenie priebehov tréovania.



Úspešnosť na testovacích dátach:

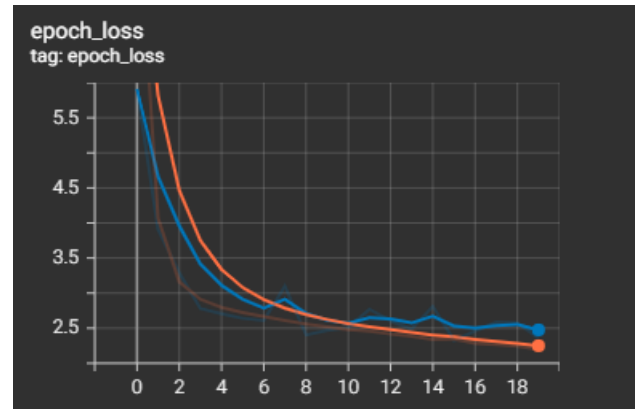
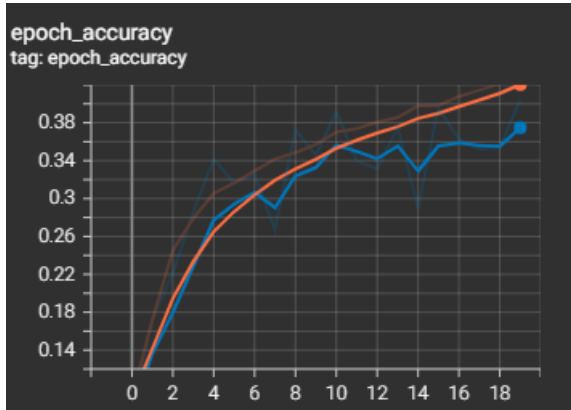
Test loss: 2.3495140075683594 Test Acc: 0.4180000126361847

Konvolučná matica:



Druhé tréovanie

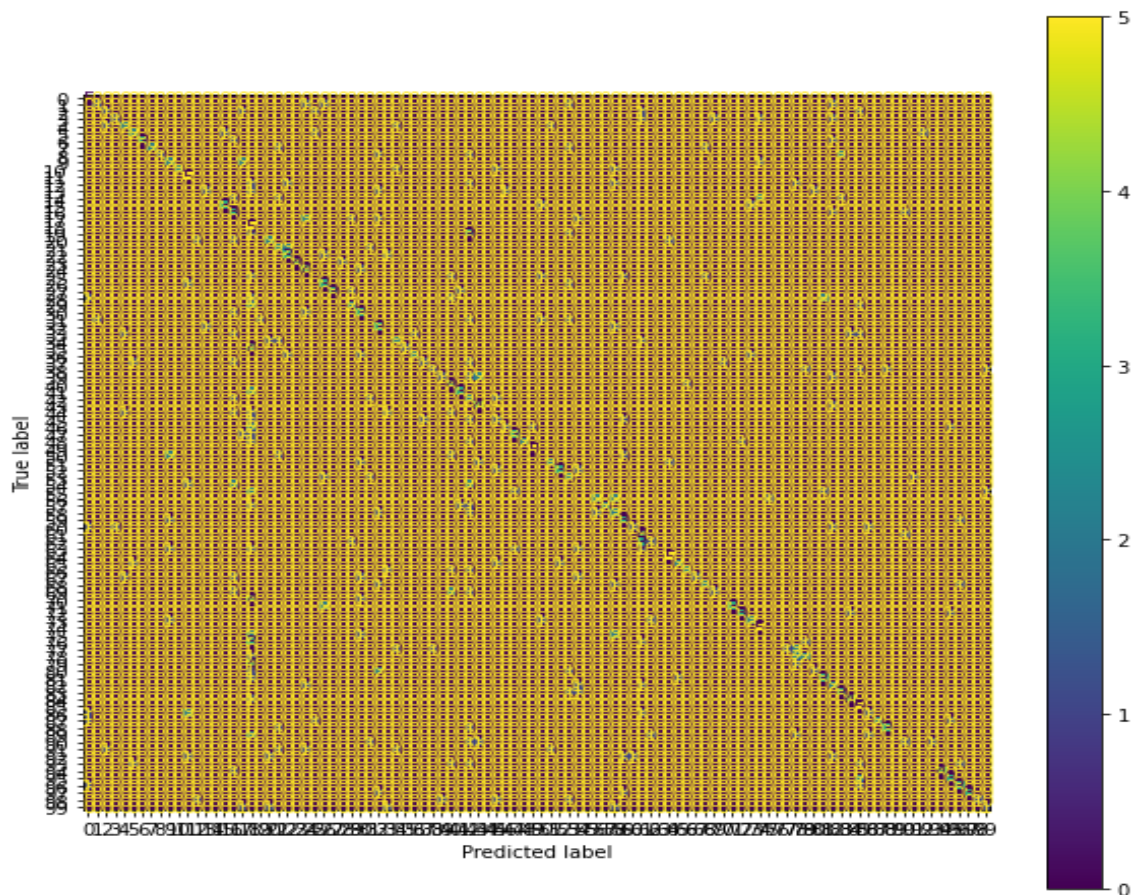
Zmenil som počet filtrov v druhej vrstve na 64. Optimalizačnú funkciu som nastavil na adam. A Early stopping som zmenšil len na 2 epochy.



Úspešnosť na testovacej množine:

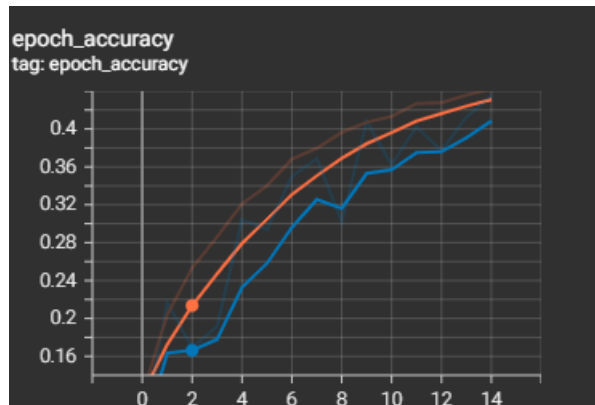
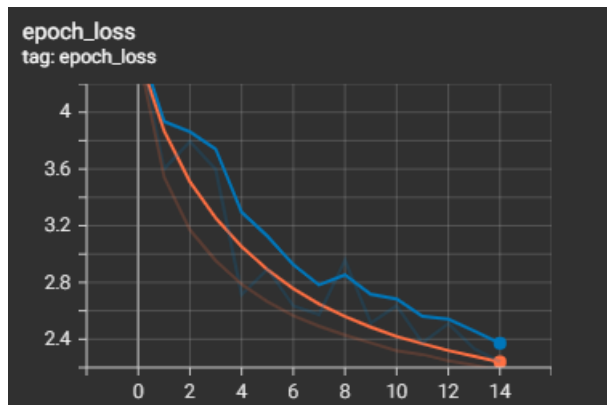
Test loss: 2.440206289291382 Test Acc: 0.34200000762939453

Konvolučná matica:



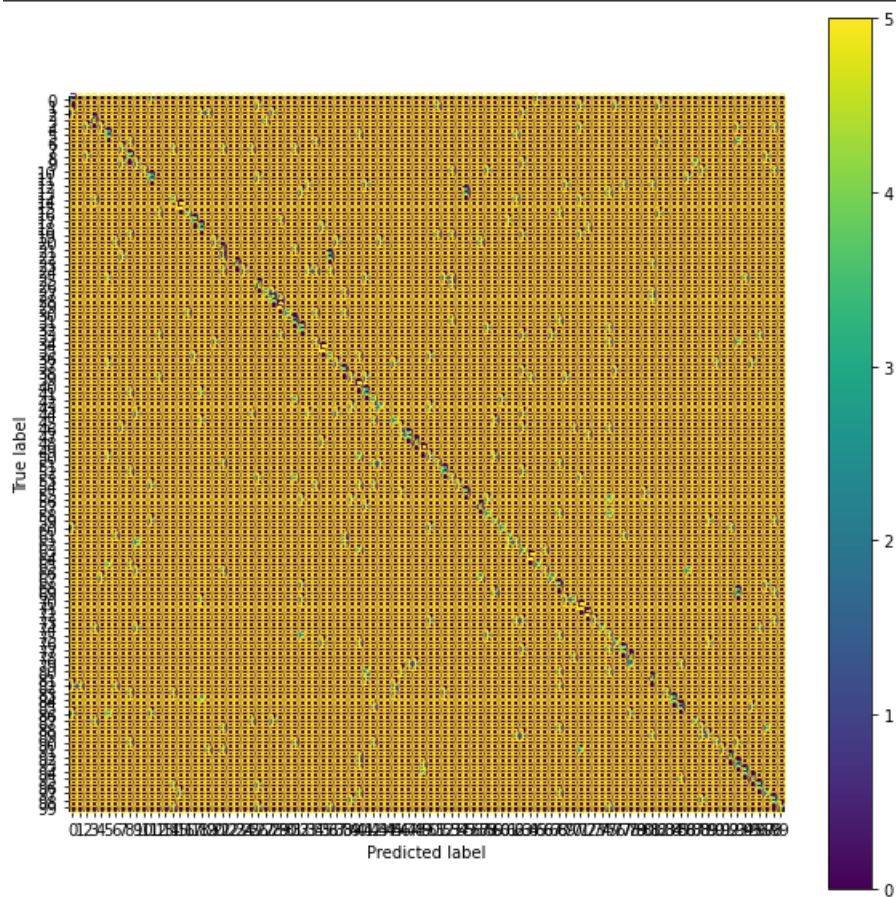
Tretie tréovanie

Pridal som ďalšiu konvolučnú vrstvu s tými istými parametrami ako mala predošlá sieť.



Úspešnosť: **Test loss: 2.1733791828155518 Test Acc: 0.44999998807907104**

Konfúzna matica:

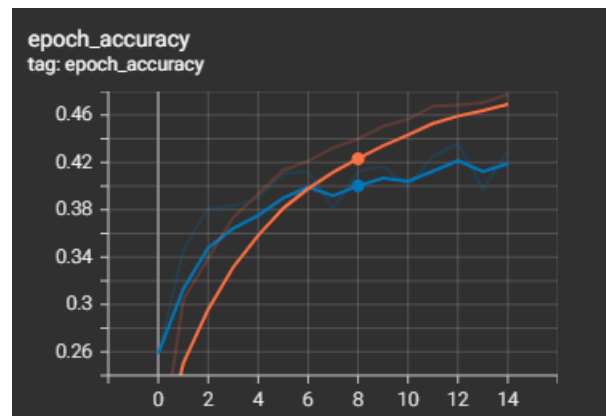
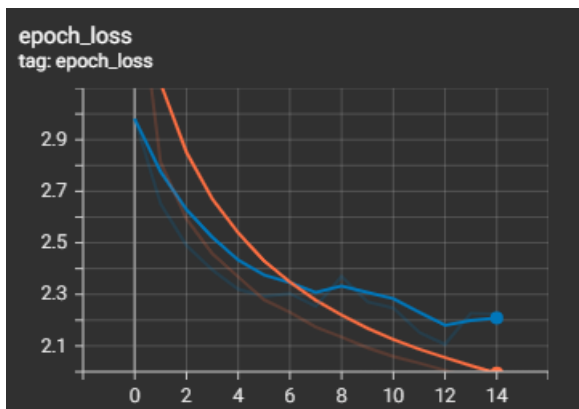


Generovanie príznakov z VGG16

Pomocou príkazu `include_top=False`, som odstránil poslednú vrstvu zo siete VGG16. Vygeneroval som príznaky pre každú množinu pomocou príkazu `predict`, príznaky som si následne uložil pomocou `np.save`.

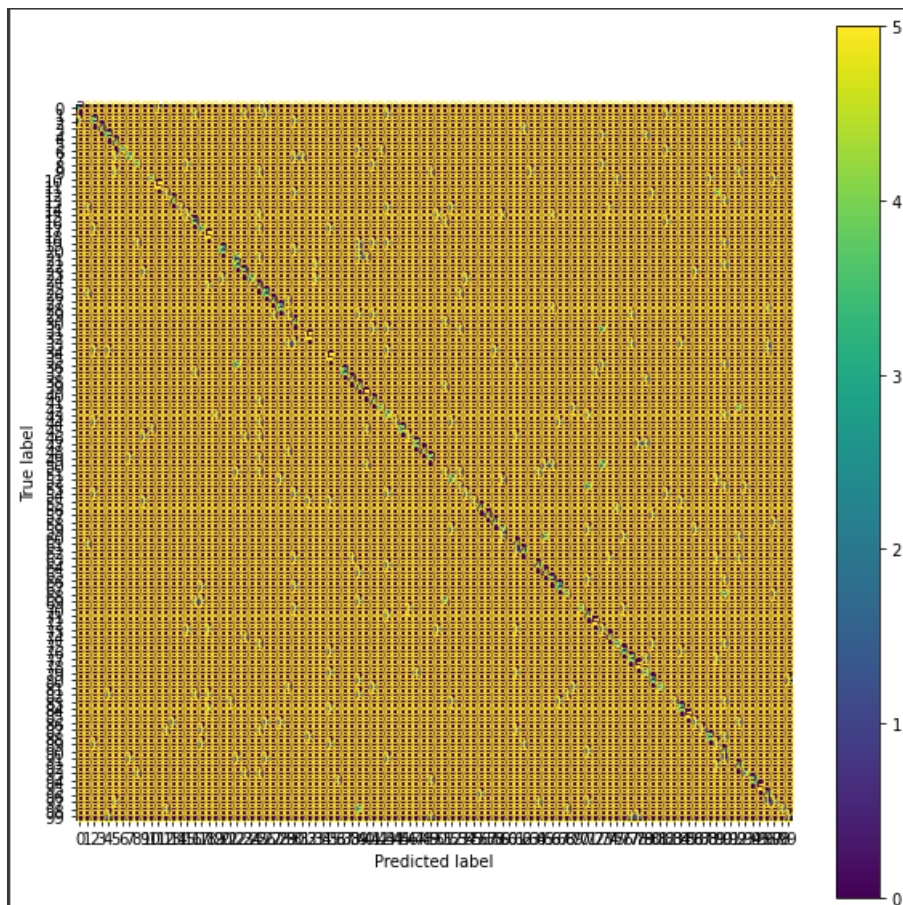
Vytvorenie výstupnej vrstvy a napojenie na VGG16

Vrstvy zo siete VGG16 som zamrazil tak, že som nastavil `trainable` `false` na všetky vrstvy VGG16. Vytvoril som ešte jednu skrytú vrstvu s 256 neurónmi a výstupnú vrstvu so 100 neurónmi. Optimizer `adam`, aktivačné funkcie `relu`.

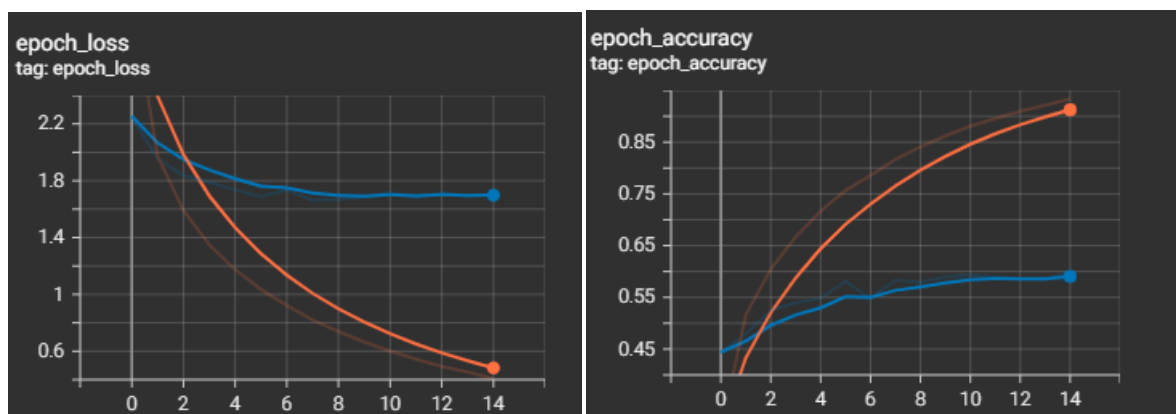


Úspešnosť: `Test loss: 1.9848839044570923 Test Acc: 0.5139999985694885`

Konfúzna matica:

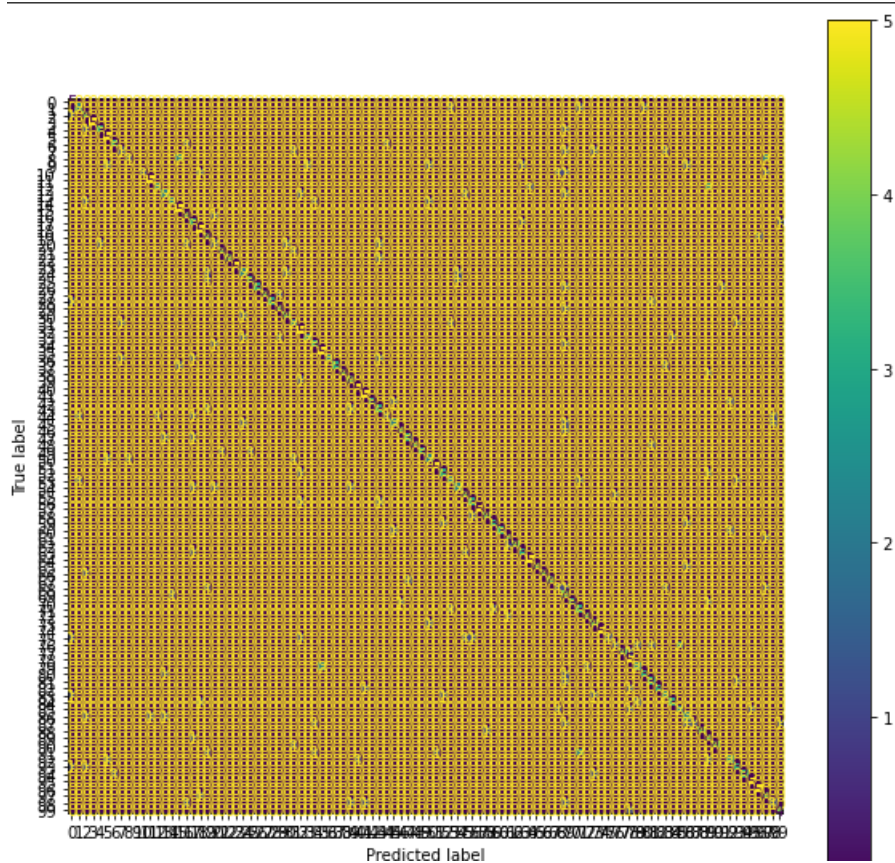


Druhé tréovanie – vymazal som vytvorenú skrytú vrstvu a nechal som len výstupnú.



Úspešnosť na testovacích dátach: `Test loss: 1.508981704711914 Test Acc: 0.6420000195503235`

Konfúzna matica:



Príznamy ktoré som dostal z VGG16 som pomocou PCA zredukoval na 3 dimenzie. S KMeans som vytvoril 6 skupín, ktoré som následne zobrazil v grafe.

Zlta skupina = hockey, field hockey, tenis, croquet, curling, axe throwing, golf

Cervena skupina = cricket, bull riding, harness racing, horse jumping, polo, jousting

Fialova skupina = arm wrestling, sumo wrestling, shot put, Olympic wrestling, steer wrestling

Oranzova skupina = rugby, football, golf, lacrosse, frisbee

Modra skupina = snowboarding, sky surfing, snowmobile racing, ice yachting, bobsled

Ruzova skupina = surfing, rowing, hydroplan racing, water cycling, fly fishing

