

I-SUNS: Zadanie č. 3

Richard Körösi (111313)

Deadline: 12.12.2023

1 Prvá časť - Spoznávanie dát

V tejto časti zadania sme sa venovali analýze dát.

1.1 Analýza početnosti v triedach a zobrazenie reprezentantov z tried

```
Train data:  
Found 4860 files belonging to 90 classes.  
Using 4374 files for training.  
  
Validation data:  
Found 4860 files belonging to 90 classes.  
Using 486 files for validation.  
  
Test data:  
Found 540 files belonging to 90 classes.  
  
Class names:  
['antelope', 'badger', 'bat', 'bee', 'beetle', 'bison', 'boar', 'butterfly', 'cat', 'caterpillar', 'chimpanzee', 'cockroach',  
'cow', 'coyote', 'crab', 'crow', 'deer', 'dog', 'dolphin', 'donkey', 'dragonfly', 'duck', 'eagle', 'elephant', 'flamingo', 'fly', 'fox',  
'goat', 'goldfish', 'goose', 'gorilla', 'grasshopper', 'hamster', 'hare', 'hedgehog', 'hippopotamus', 'hornbill', 'horse', 'hummingbird',  
'hyena', 'jellyfish', 'kangaroo', 'koala', 'ladybugs', 'leopard', 'lion', 'lizard', 'lobster', 'mosquito', 'moth', 'mouse', 'octopus',  
'okapi', 'orangutan', 'otter', 'owl', 'ox', 'oyster', 'panda', 'parrot', 'pelecaniformes', 'penguin', 'pig', 'pigeon', 'porcupine',  
'possum', 'raccoon', 'rat', 'reindeer', 'rhinoceros', 'sandpiper', 'seahorse', 'seal', 'shark', 'sheep', 'snake', 'sparrow', 'squid',  
'squirrel', 'starfish', 'swan', 'tiger', 'turkey', 'whale', 'wolf', 'wombat', 'woodpecker', 'zebra']
```

Figure 1: Početnosť dát a názvy tried

Obrázok č.1 zobrazuje informácie o počte dát a ich rozdelenie do trénovacej, testovacej a validačnej množiny. Taktiež zobrazuje aj názvy všetkých 90 tried, ktoré nám boli poskytnuté. Každá trieda je zastúpená 54 obrázkami v časti pre trénovanie a validáciu a 6 obrázkami pre testovanie.

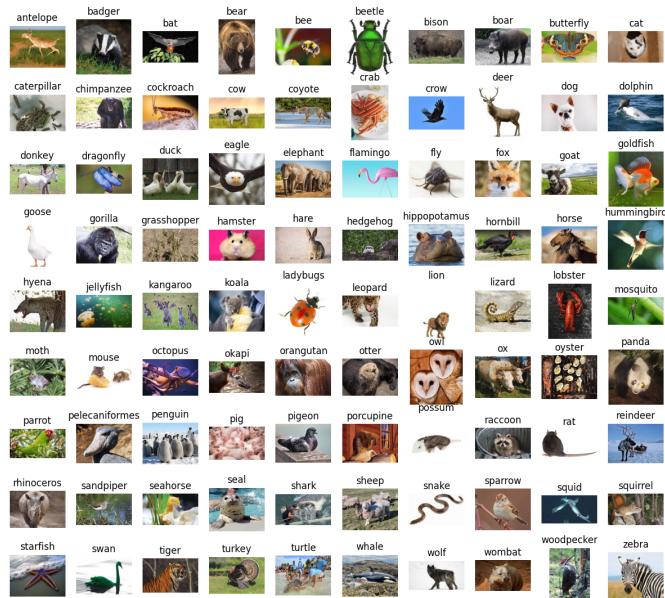


Figure 2: Reprezentanti z každej triedy

Obrázok č.2 zobrazuje všetkých 90 tried a taktiež aj reprezentanta z každej triedy.

1.2 Analýza tried pomocou modelu natrénovaného na probléme Imagenet

```
1 antelope: [('gazelle', 0.5), ('impala', 0.33), ('hartebeest', 0.12)]
2 badger: [('badger', 0.96), ('brown_bear', 0.02), ('skunk', 0.02)]
3 bat: [('weasel', 0.1), ('howler_monkey', 0.06), ('walking_stick', 0.06)]
4 bear: [('brown_bear', 0.69), ('American_black_bear', 0.27), ('red_wolf', 0.02)]
5 bee: [('bee', 0.9), ('honeycomb', 0.08), ('cardoon', 0.02)]
6 ...
7 cat: [('Egyptian_cat', 0.46), ('tabby', 0.25), ('tiger_cat', 0.08)]
8 caterpillar: [('centipede', 0.25), ('lacewing', 0.12), ('leaf_beetle', 0.06)]
9 chimpanzee: [('chimpanzee', 0.98), ('orangutan', 0.02)]
10 cockroach: [('cockroach', 1.0)]
11 cow: [('ox', 0.82), ('Great_Dane', 0.08), ('Ibizan_hound', 0.04)]
12 coyote: [('coyote', 0.88), ('red_wolf', 0.04), ('dingo', 0.02)]
13 ...
14 whale: [('grey_whale', 0.67), ('killer_whale', 0.1), ('dugong', 0.1)]
15 wolf: [('timber_wolf', 0.77), ('white_wolf', 0.11), ('red_wolf', 0.06)]
16 wombat: [('wombat', 0.98), ('koala', 0.02)]
17 woodpecker: [('black_grouse', 0.23), ('bee_eater', 0.15), ('jacamar', 0.13)]
18 zebra: [('zebra', 0.98), ('hartebeest', 0.02)]
```

Listing 1: Predikcie modelu EfficientNetB4

V tejto časti sme sa zamreli na analýzu tried pomocou modelu natrénovaného na probléme Imagenet, konkrétnie išlo o model EfficientNetB4, ktorý sme sa rozhodili použiť aj v neskôrších častiach zadania. Model dokázal vo väčšine prípadov správne predikovať dané zviera (niekedy však namiesto druhu predpovedal konkrétny druh ako napríklad na riadku 7 v Listings 1). Nastali však prípady ako napríklad na radku č.5, kde model namiesto včely predpovedal včelí vosk. Môže to byť aj vďaka tomu, že včelu ako samotnú triedu daný model nepozná (nie tento prípad, ako tento príklad si môžeme uviesť riadok č.3), alebo aj tým, že pozadie na niektorých fotkách bol včelí vosk a na základe toho sa model rozhodol určiť tomu vyššiu váhu.

```
1 pigeon:[('black_grouse', 0.26), ('partridge', 0.23), ('Eu_gallinule', 0.17)]
```

Riadok nad týmto textom zobrazuje prípad, kde si bol model nedokázal správne zaraďovať obrázky do kategórie (najčastejšie zaraďuje holuba do kategórie "Tetrov hoňiak").

```
1 cockroach: [('cockroach', 1.0)]
```

Riadok nad týmto textom zobrazuje prípad, kde si model zaradil všetky obrázky správne.

```
1 cow:[('ox', 0.82), ('Great_Dane', 0.08), ('Ibizan_hound', 0.04)]
2 ox:[('ox', 0.71), ('oxcart', 0.09), ('triceratops', 0.07)]
```

Riadky nad týmto textom zobrazujú dve podobné triedy (vôl[tur domáci] a krava), model v oboch prípadoch väčšinu obrázkov zaradil do rovnakej kategórie. Daná predikcia dáva zmysel keďže vôl a krava patria viac menej do toho istého druhu zvieratá.

2 Druhá časť - trénovanie konvolučných sietí

V tejto časti sme mali za úlohu vytvoriť a natrénovať konvolučnú neurónovú sieť na klasifikáciu zvierat. Cieľom bolo presiahnuť náhodnú úspešnosť a dosiahnuť viditeľnú diagonálu na konfúznej matici. Z poskytnutých dát sme si vytvorili trénovaciu, validačnú a testovaciu množinu, veľkosť obrázkov sme v generátoroch upravili na takú veľkosť aby s nimi mohol EfficientNetB4 pracovať (380,380). Trénovaciu a validačnú množinu sme rozdelili v pomere 9:1.

2.1 Vytvorenie siete - Augmentácia, Zmena veľkosti, Normalizácia

Pre zrýchlenie učenia našej siete sme si zadefinovali menšiu veľkosť obrázkov a taktiež sme znormalizovali hodnoty pixelov z $<0;255>$ na $<0;1>$

```
1 keras.layers.Resizing(150, 150)
2 keras.layers.Rescaling(1. / 255)
```

Okrem iného sme dáta aj augmentovali (náhodne prevraciame obrázky horizontálne a vertikálne a tiež náhodne otáčame obrázky o 25 percent).

```
1 keras.layers.RandomFlip("horizontal_and_vertical")
2 keras.layers.RandomRotation(0.2)
```

2.2 Trénovanie a porovnanie troch konvolučných sietí

Pri všetkých troch pokusoch sme využili rovnaké upravené dáta z kapitoly 2.1 (pri všetkých pokusoch sme mali zachovanú rovnakú veľkosť a aj vlastnosti augmentácie). Ako povinnú regularizačnú techniku sme si vybrali Dropout.

(Kon.vrstva)Info	Pokus 1	Pokus 2	Pokus 3
(1)Conv2D	filters=16;kernel_size=3	filters=32;kernel_size=3	filters=16, kernel_size=3
(1)MaxPooling2D	✓	✓	✓
(1)Dropout	0.2	0.4	0.4
(2)Conv2D	filters=16;kernel_size=3	filters=32;kernel_size=3	filters=16, kernel_size=3
(2)MaxPooling2D	✓	✓	✓
(2)Dropout	0.3	0.4	0.3
Flatten	✓	✓	✓
Dense	90	90	90
Optimizer, Learning rate	Adam,0.0001	Adam,0.00005	Adam,0.00002
EarlyStopping patience	1	2	3
Epochs	40	40	70
Train Accuracy	0.2032	0.1740	0.1710
Test Accuracy	0.1704	0.1426	0.1407

Table 1: Porovnanie troch pokusov

Confusion matrix on train set

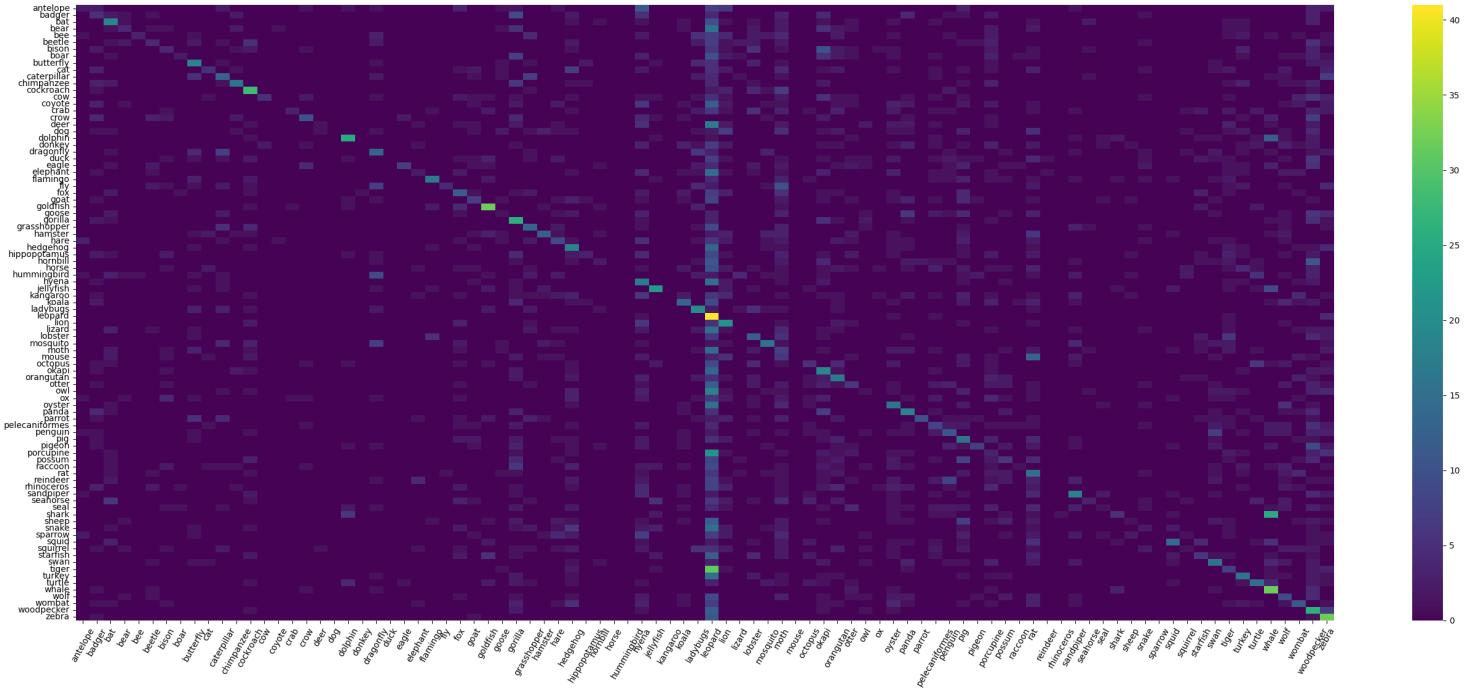


Figure 3: Konfúzna matica trénovacích dát

Confusion matrix on test set

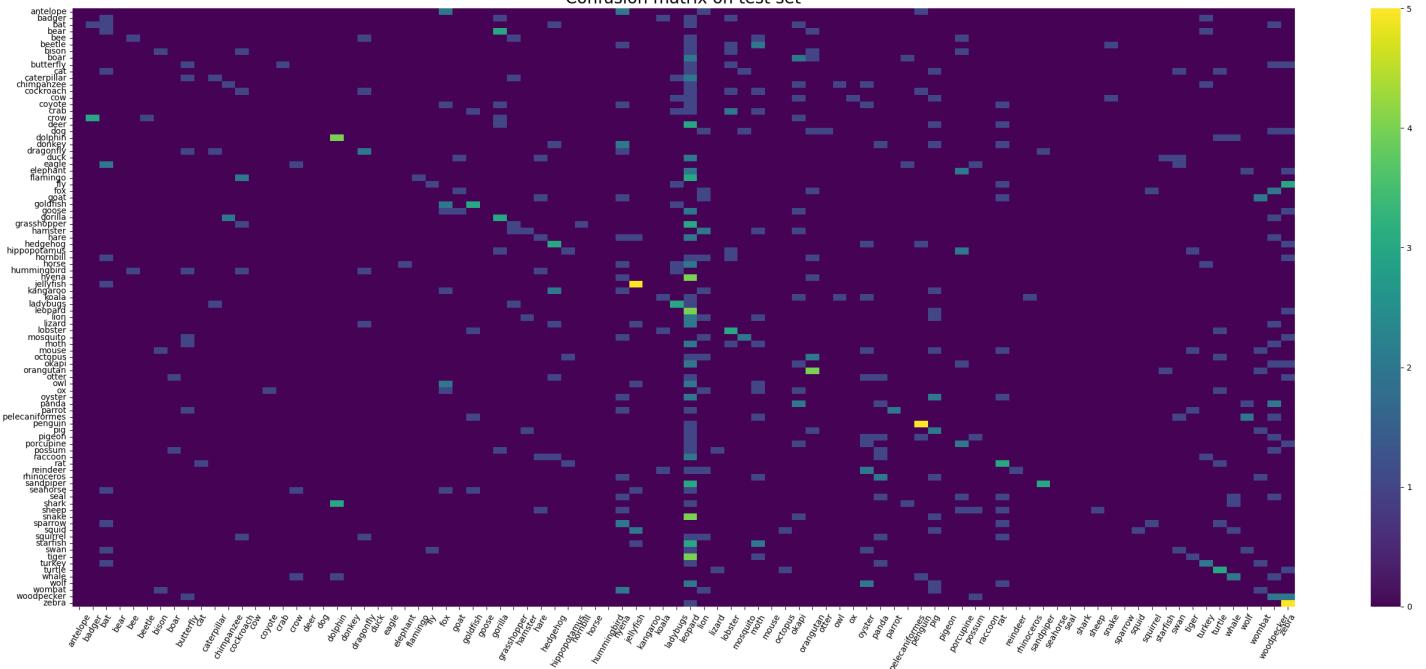


Figure 4: Konfúzna matica testovacích dát

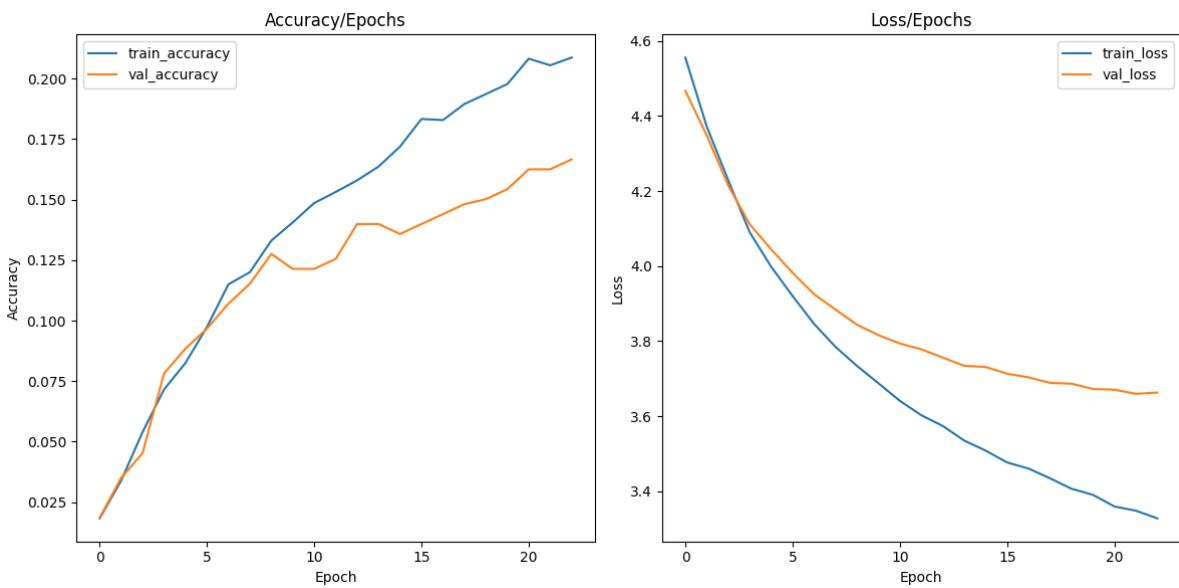


Figure 5: Priebeh trénovania

Obrázky v tejto kapitole zobrazujú výsledky a priebeh trénovania najúspešnejšieho pokusu. Môžeme konštatovať, že naša sieť netrpela pretrénovaním ani podtrénovaním, na konfúznej matici testovacích dát je viditeľná diagonálna a podarilo sa nám prekonať náhodnú úspešnosť.

3 Tretia časť - zhľukovanie príznakov

3.1 Generovanie príznakov

Na tvorbu príznakov sme použili už vyššie spomenutú natrénovanú sieť na probléme ImageNet EfficientNetB4 (nastavili sme mu parameter include_top na false). Sieť nám vytvorila dataset s 1792 príznakmi, okrem toho sme do datasetu ešte vložili cestu k obrázkom a ich pravdivú kategóriu.

	C1	C2	C3	C4	C5	C6
1	pathOfImage	actualClass	feature_0	feature_1	feature_2	feature_3
2	data/train/antelope/03d7fc0888.jpg	antelope	-0.24481949	-0.20981291	-0.12946916	0.29728785
3	data/train/antelope/0a37838e99.jpg	antelope	-0.24031968	-0.27846178	-0.26123607	0.03061515
4	data/train/antelope/0b1a3af197.jpg	antelope	-0.19581698	-0.24434336	-0.25733095	-0.1140776
5	data/train/antelope/0b688923b0.jpg	antelope	-0.22328664	-0.25195587	-0.27339756	-0.18107347
6	data/train/antelope/0c16ef86c0.jpg	antelope	-0.2588496	-0.27652144	-0.20003018	0.25407556
7	data/train/antelope/0e17715606.jpg	antelope	-0.21735094	-0.23605807	-0.26355734	-0.07700461
8	data/train/antelope/0ee903ea13.jpg	antelope	-0.2772582	-0.21730958	0.036169812	-0.23587455
9	data/train/antelope/0fb2e9aa81.jpg	antelope	-0.269753	-0.22858317	-0.27502143	0.814737
10	data/train/antelope/0fe7076f06.jpg	antelope	-0.14194408	-0.24633105	-0.11003427	0.14162177

Figure 6: Časť datasetu

3.2 Zhlukovanie príznakov

V tejto časti sme sa venovali zhlukovaniu príznakov, pre vytváranie zhlukov sme využili zhlukovací algoritmus (Centroid model) KMeans, kde sme nastavili, že chceme 35 zhlukov. Pôvodne sme v datasete mali 1792 rôznych príznakov, preto sme taktiež využili redukciu dimenzie za pomoci PCA (PCA(n_components=5)).



Figure 7: Enter Caption

Obrázok č.7 zobrazuje priemerné obrázky všetkých 35 zhlukov. Medzi zaujímavé zhluky podľa priemerných obrázkov sme zaradili nasledovné: Cluster20, Cluster34, Cluster10, Cluster27, Cluster30. Podľa samotných obrázkov daných zhlukov k súmě k zaujímavým zhlukom pridali ešte aj Cluster8.

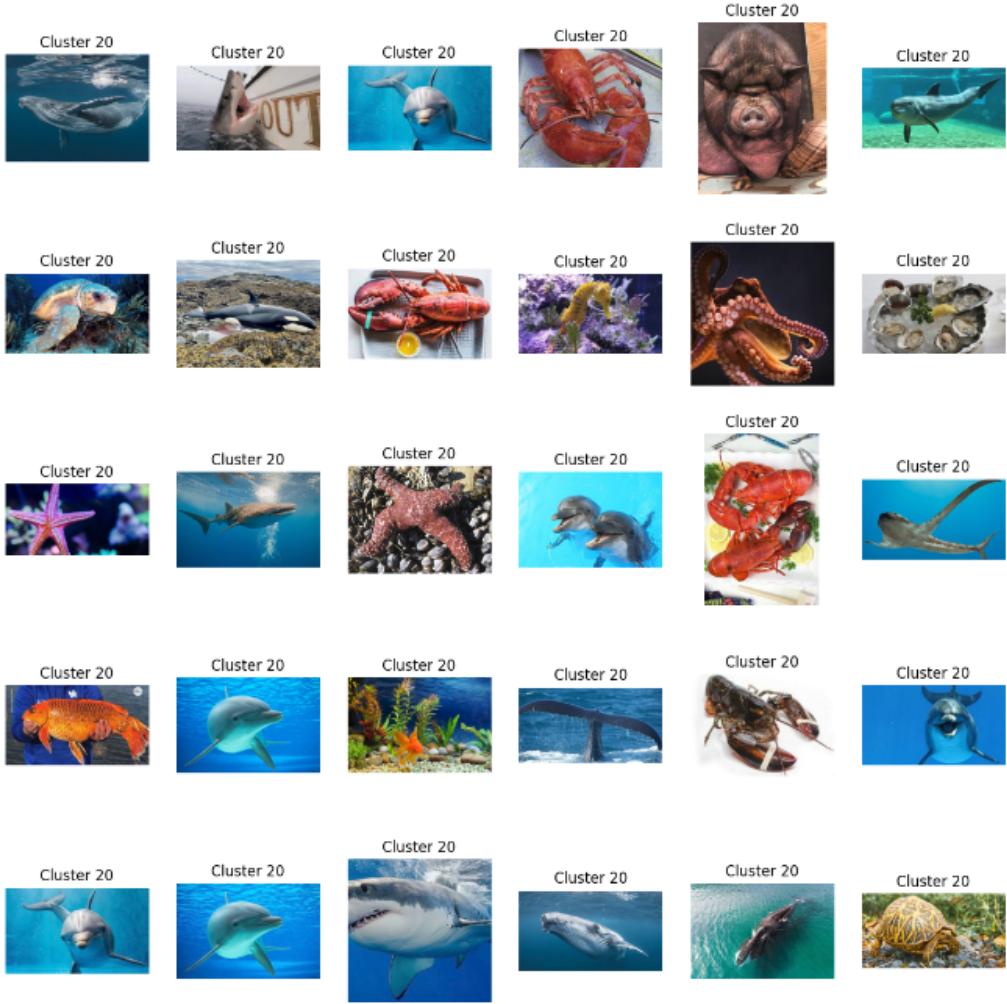


Figure 8: Podmnožina obrázkov zhluku 20

Tento zhluk obsahoval obrázky prevažne s modrým pozadím a morskými živočíchmi, dokazuje to aj obrázok č.7, kde si môžeme všimnúť pomerne modrý priemerný obrázok zhluku 20. Väčšinu obrázkov v danej podmnožine tvoria delfíny, žraloky a veľryby.

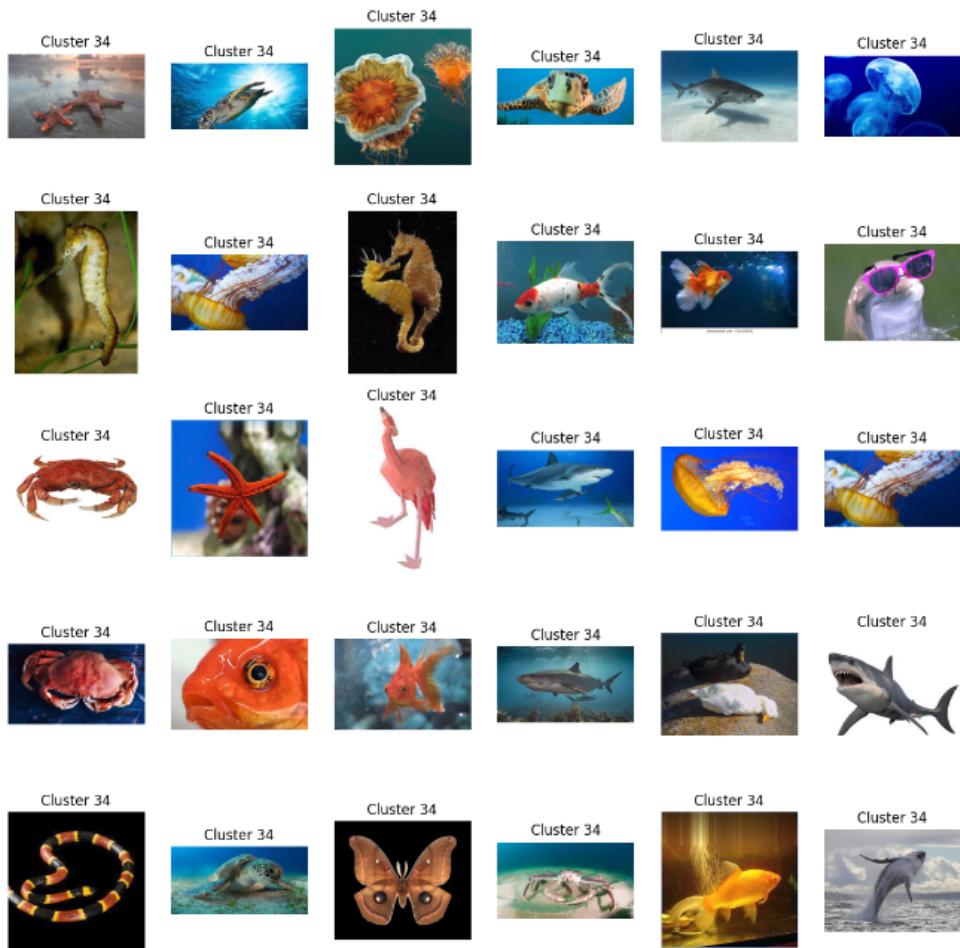


Figure 9: Podmnožina obrázkov zhluku 34

Tento zhluk obsahoval obrázky prevažne s modrým pozadím a morskými živočíchmi, dokazuje to aj obrázok č.7, kde si môžeme všimnúť pomerne modrý priemerný obrázok zhluku 34. Oproti predošlému zhluku tento zhluk obsahuje po väčšine iné živočíchy ako delfíny, žraloky a veľryby (farebnejšie druhy).

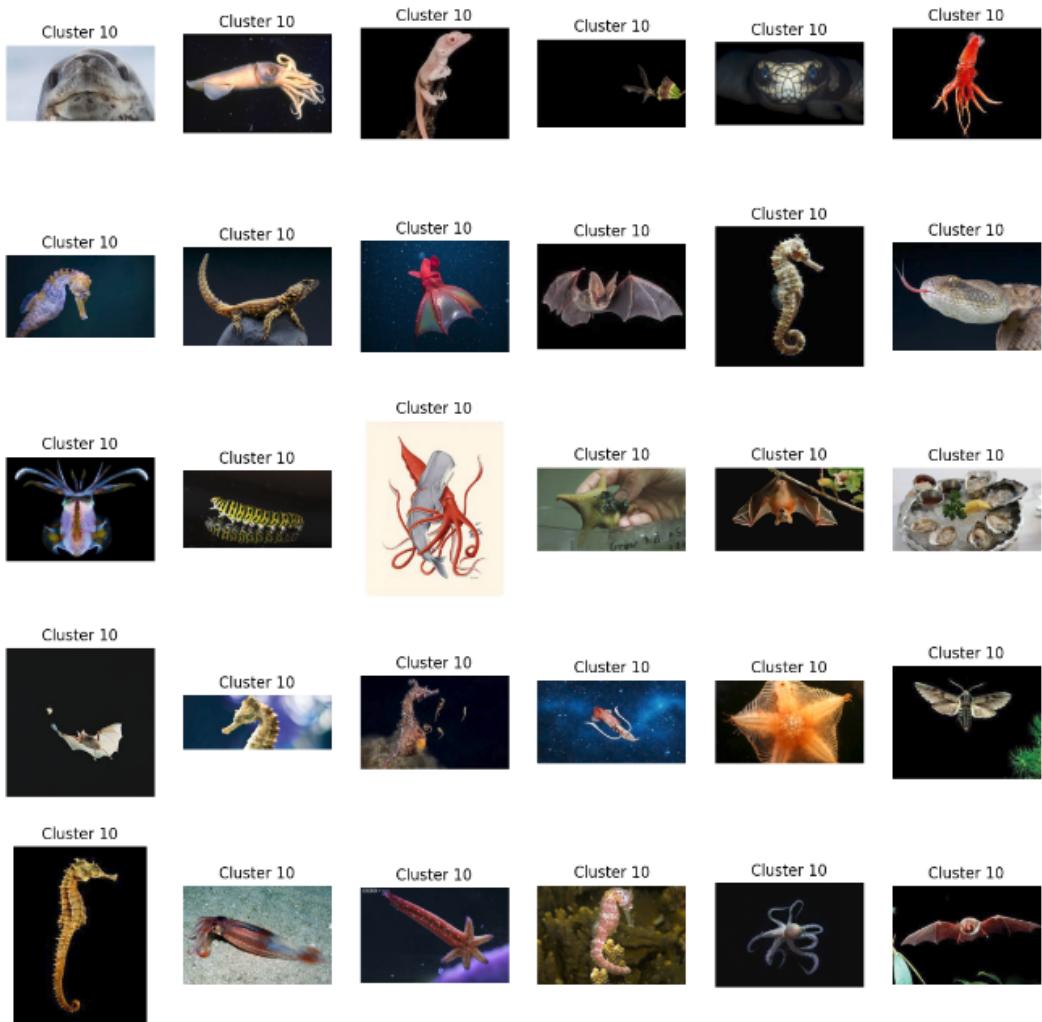


Figure 10: Podmnožina obrázkov zhluku 10

Tento zhluk obsahoval obrázky prevažne s čiernym pozadím, dokazuje to aj obrázok č.7, kde si môžeme všimnúť pomerne tmavý priemerný obrázok zhluku 10. Môžeme si teda všimnúť, že v zhluku sa nachádzajú buď nočné živočíchy alebo, podmorské živočíchy, ktoré žijú na miestach s nedostatkom denného svetla.

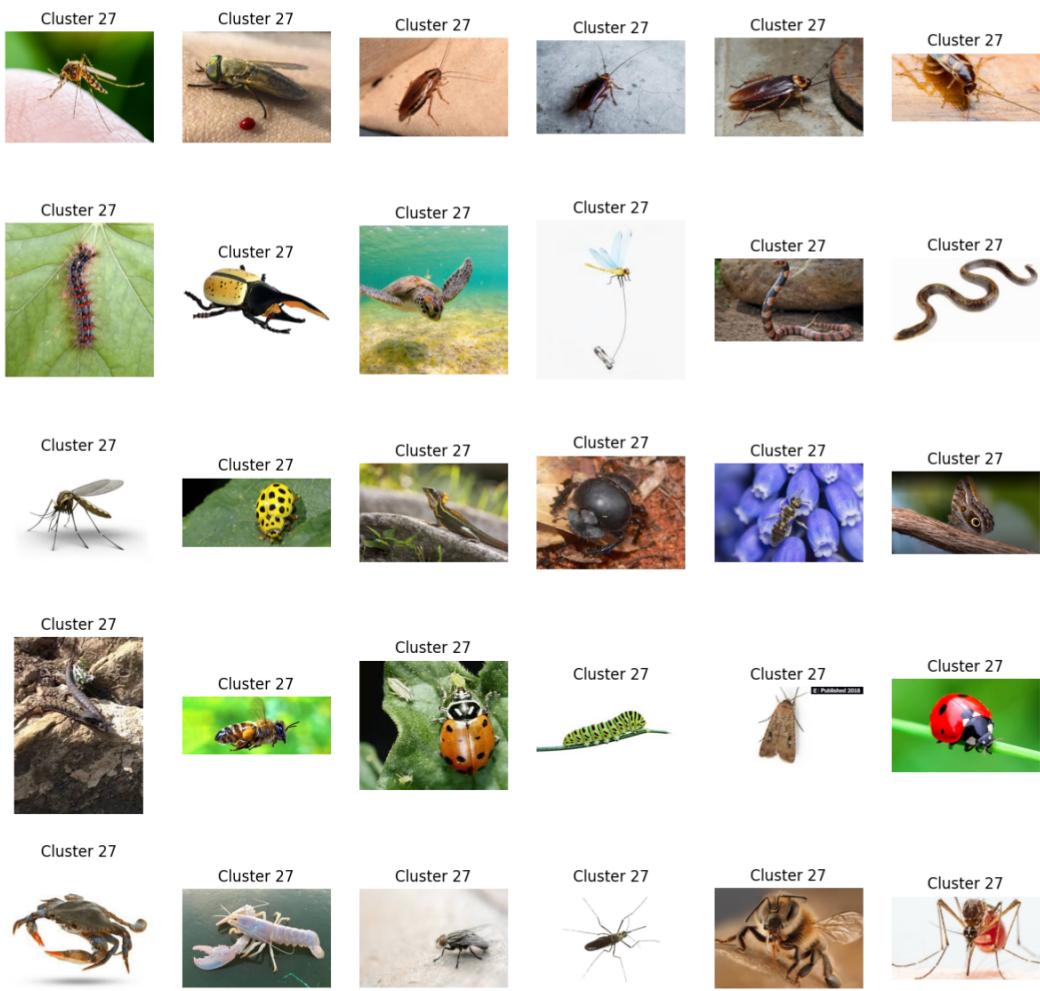


Figure 11: Podmnožina obrázkov zhluku 27

Tento zhluk obsahoval obrázky prevažne s bielym pozadím, dokazuje to aj obrázok č.7, kde si môžeme všimnúť pomerne bledý priemerný obrázok s tmavším stredom zhluku 27. Môžeme si teda všimnúť, že v zhluku sa nachádza najmä hmyz.

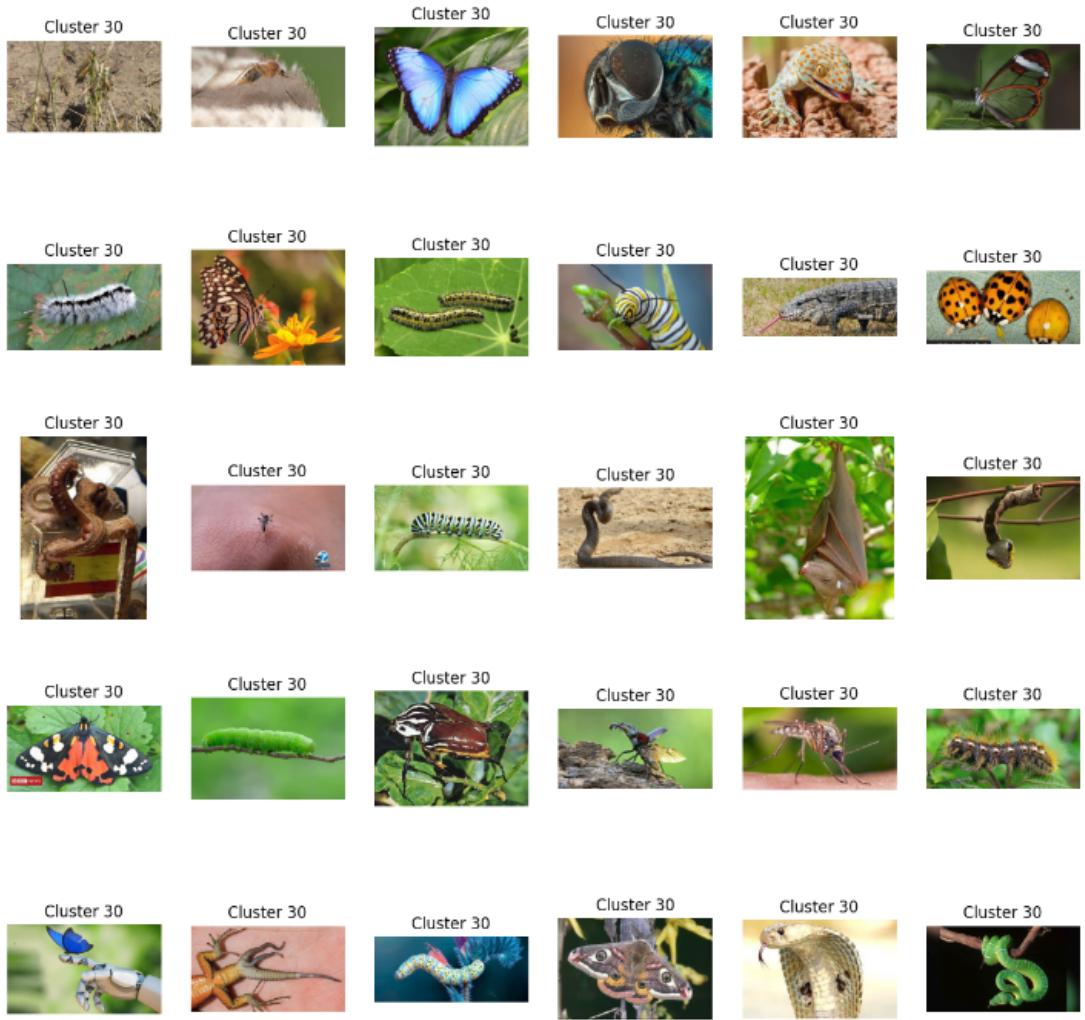


Figure 12: Podmnožina obrázkov zhluku 30

Tento zhluk obsahuje podobne ako predošlý zhluk najmä hmyz, na rozdiel od predošlého zhluku sú však obrázky zafarbené poväčšine do zelenej farby.

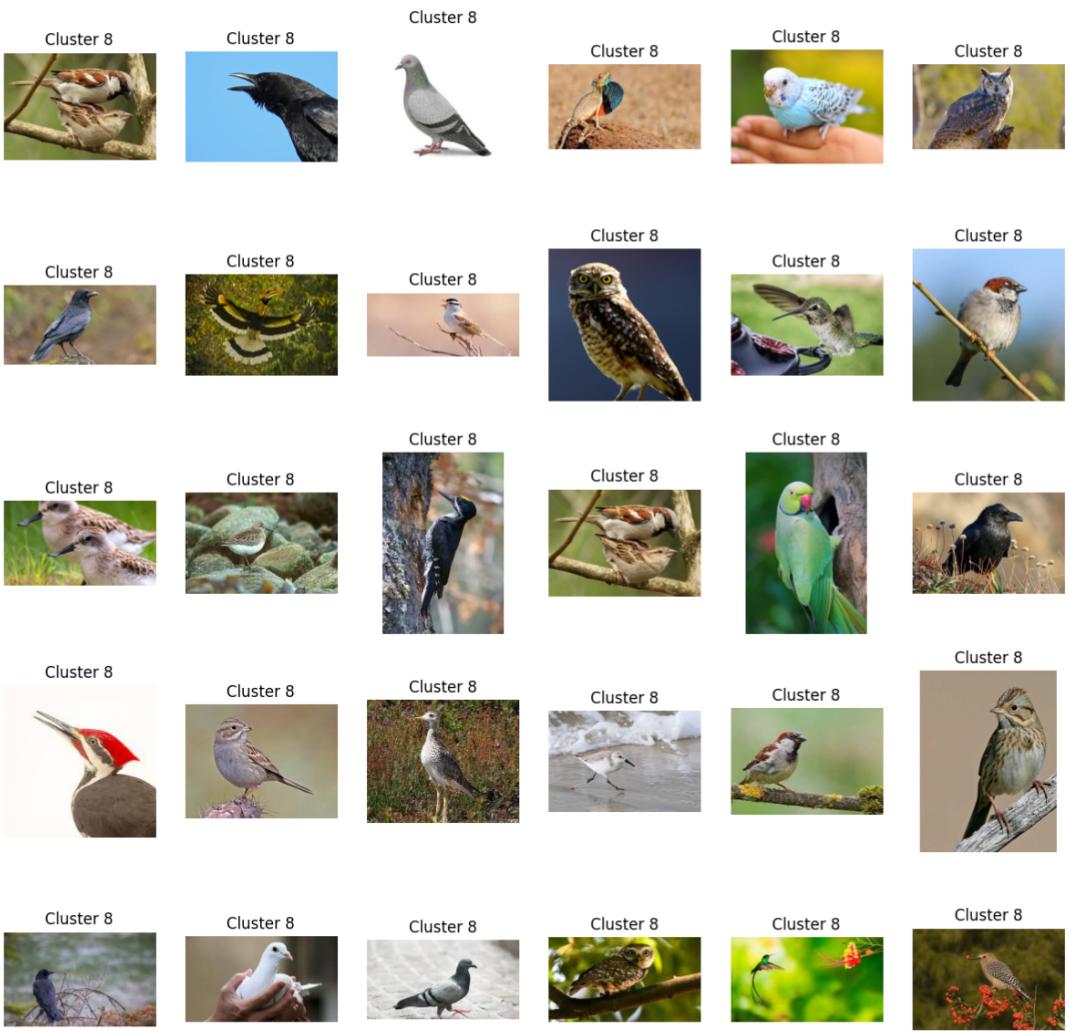


Figure 13: Podmnožina obrázkov zhľuku 8

Tento zhľuk v priemernom obrázku nijako nevynikal avšak pri kontrole podmnožín obrázkov sme si všimli, že obsahuje iba obrázky vtákov.

4 Transferlearning

4.1 Dotrénovanie posledných vrstiev ImageNet siete

V tejto časti sme mali za úlohu dotrénovať posledné vrstvy ImageNet siete a tak získať lepšiu úspešnosť ako v našich predošlých troch pokusoch. Postupovali sme podľa návodu, ktorý nám bol poskytnutý a odkaz naň sa nachádza v kóde zadania. Museli sme napríklad zamraziť konvolučný základ (convolutional base), aby sa váhy tohto modelu počas trénovania neupravovali.

```
1 Model: "efficientnetb4"
2 -----
3 | Total params: 17673823 (67.42 MB) |
4 | Trainable params: 0 (0.00 Byte) |
5 | Non-trainable params: 17673823 (67.42 MB) |
6 -----
```

Listing 2: EfficientNetB4 summary

Train Accuracy	0.9040
Test Accuracy	0.8537

Table 2: Úspešnosť pokusu na trénovacích a testovacích dátach

Na základe tabuľky č.3 môžeme konštatovať, že v najnovšom pokuse sa nám podarilo zlepšiť presnosť modelu.

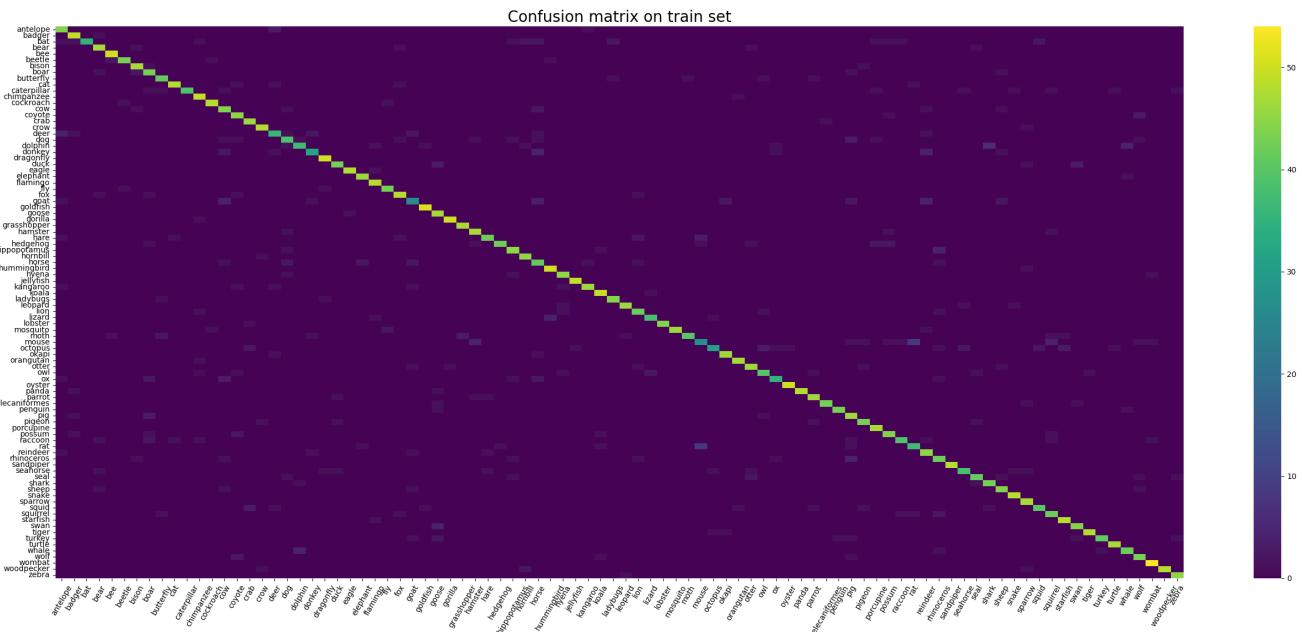


Figure 14: Konfúzna matica trénovacích dát

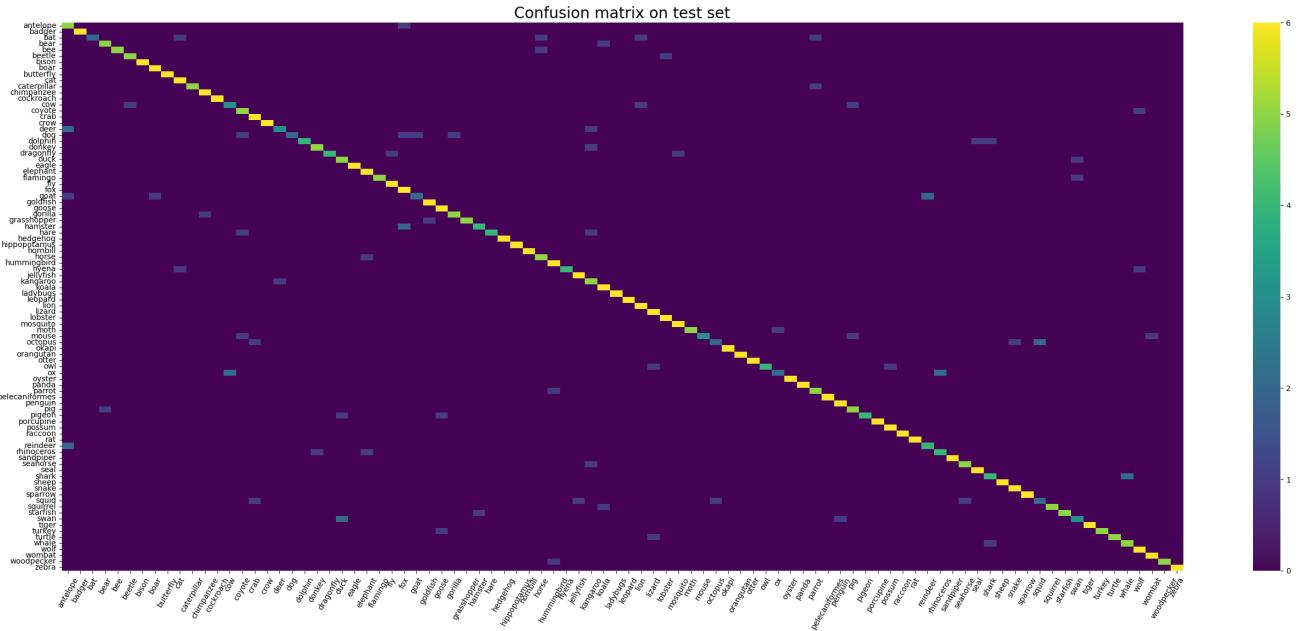


Figure 15: Konfúzna matica testovacích dát

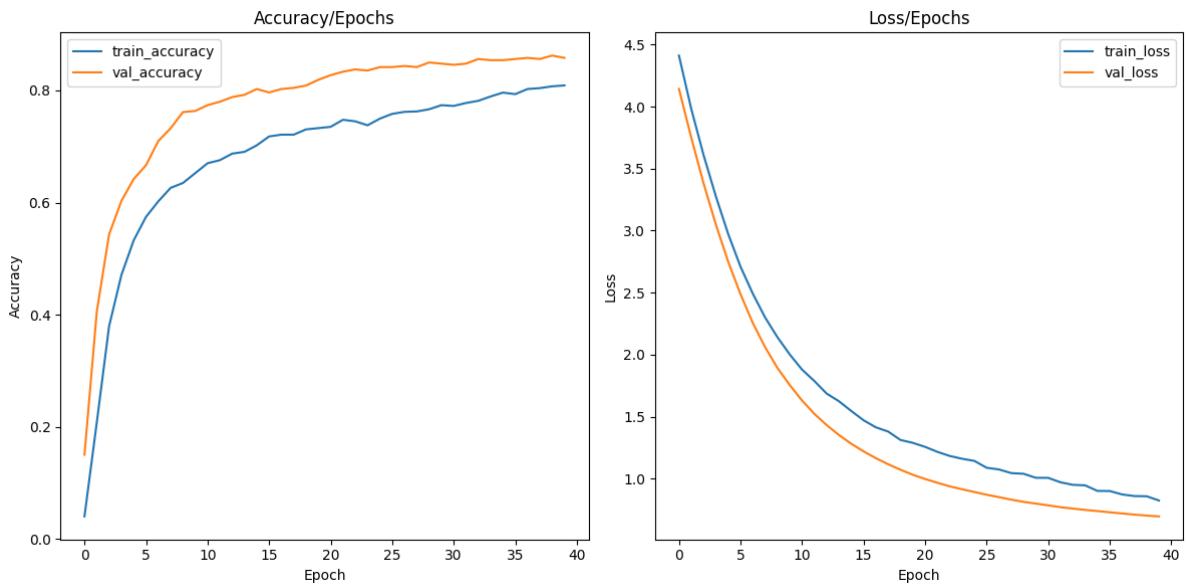


Figure 16: Priebeh trénovania

Obrázky v tejto kapitole zobrazujú výsledky a priebeh trénovania pokusu. Môžeme konštatovať, že naša sieť netrpela pretrénovaním ani podtrénovaním, na konfúznej matici testovacích dát je viditeľná diagonála a podarilo sa nám prekonať úspešnosť z predošlých testov.

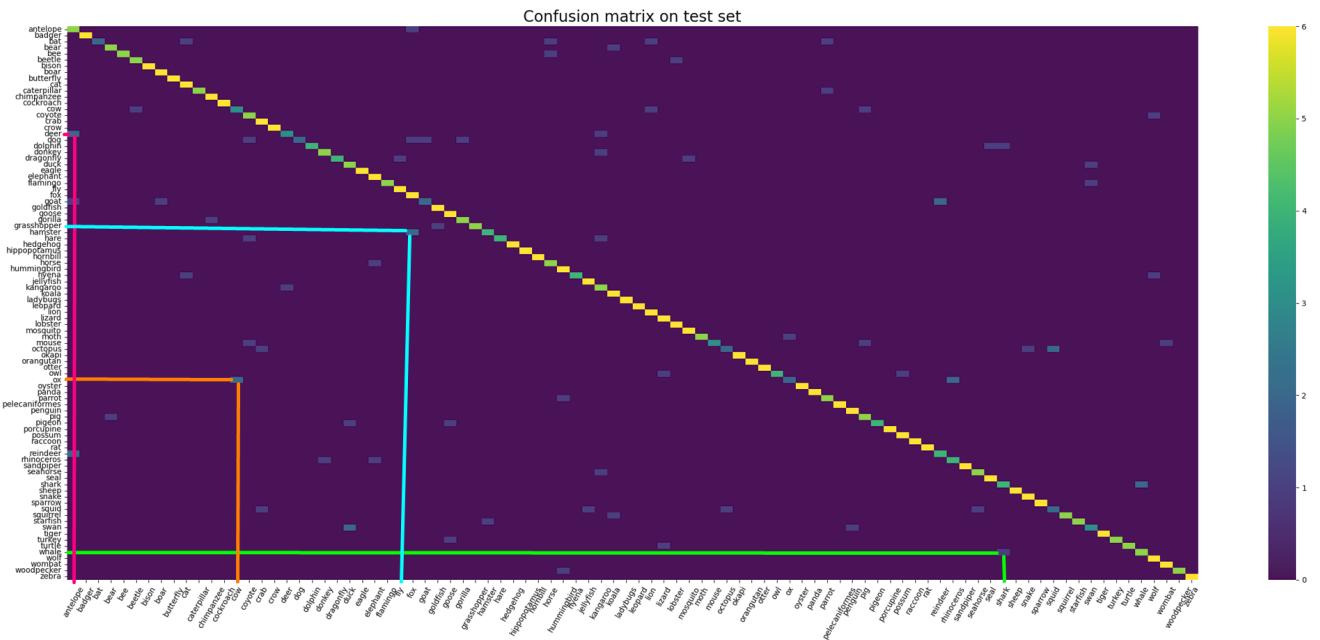


Figure 17: Analýza konfúznej matice testovacích dát

Následne sme si po krátkej analýze všimli, kde má náš model stále nedostatky. Väčšinou vždy išlo o podobný druh, ktorý si model nesprávne zaradil (morský živočích [veľryba-žralok], hmyz [kobylka-mucha], [vôl-krava] a podobne).

4.2 Trénovanie NS na pripravených príznakoch

Následne sme skúsili aj druhý spôsob a to trénovanie NS na pripravených príznakoch. Tento pokus však neboli až tak úspešný ako predošlý, preto si ukážeme len hlavné časti tohto pokusu.

```

1 model = Sequential()
2 model.add(Dense(20, input_dim=X_train.shape[1], activation='relu'))
3 model.add(Dense(20, activation='relu'))
4 model.add(Dense(20, activation='relu'))
5 model.add(Dense(90, activation='softmax'))
6 model.compile(loss='categorical_crossentropy',
7                 optimizer=Adam(learning_rate=0.000025),
8                 metrics=['accuracy'])
9 history = model.fit(x=X_train, y=y_train,
10                       validation_data=(X_valid, y_valid), epochs=200, batch_size=32,
11                       callbacks=[early_stopping])

```

Listing 3: Nastavenie NS pre trénovanie na pripravených príznakoch

Train Accuracy	0.3350
Test Accuracy	0.2222

Table 3: Úspešnosť pokusu na trénovacích a testovacích dátach

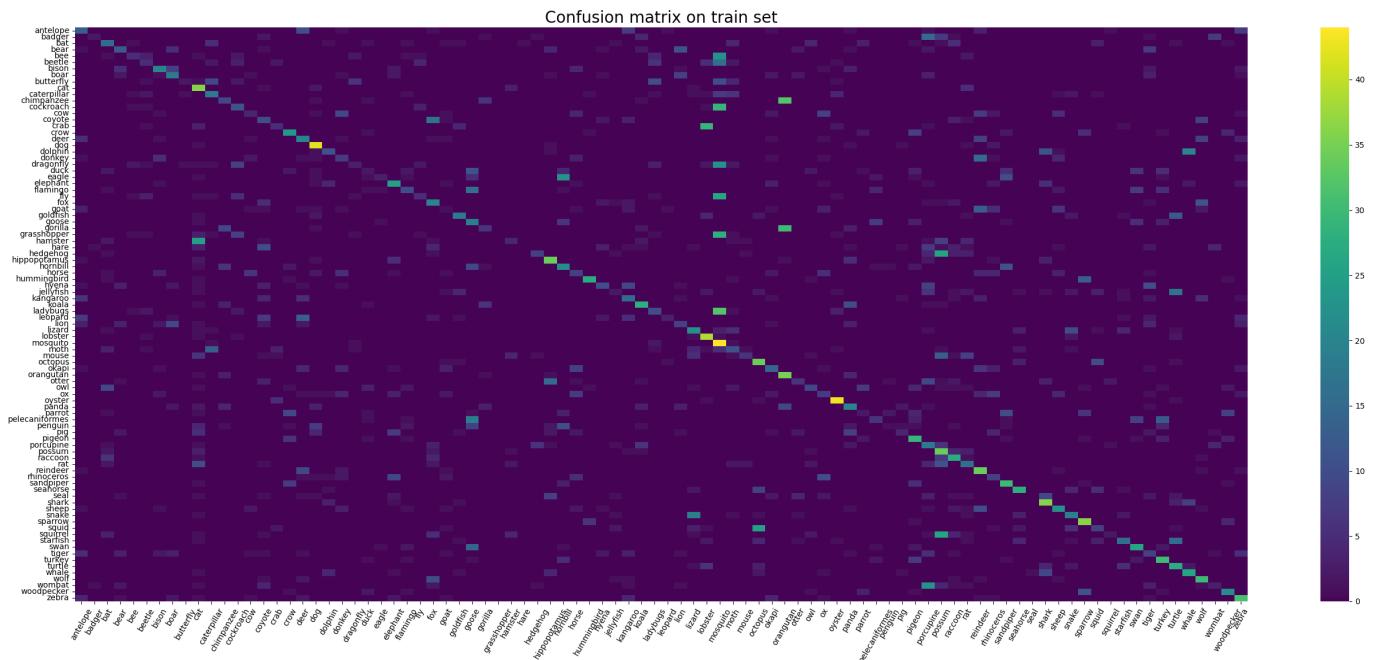


Figure 18: Konfúzna matica trénovacích dát

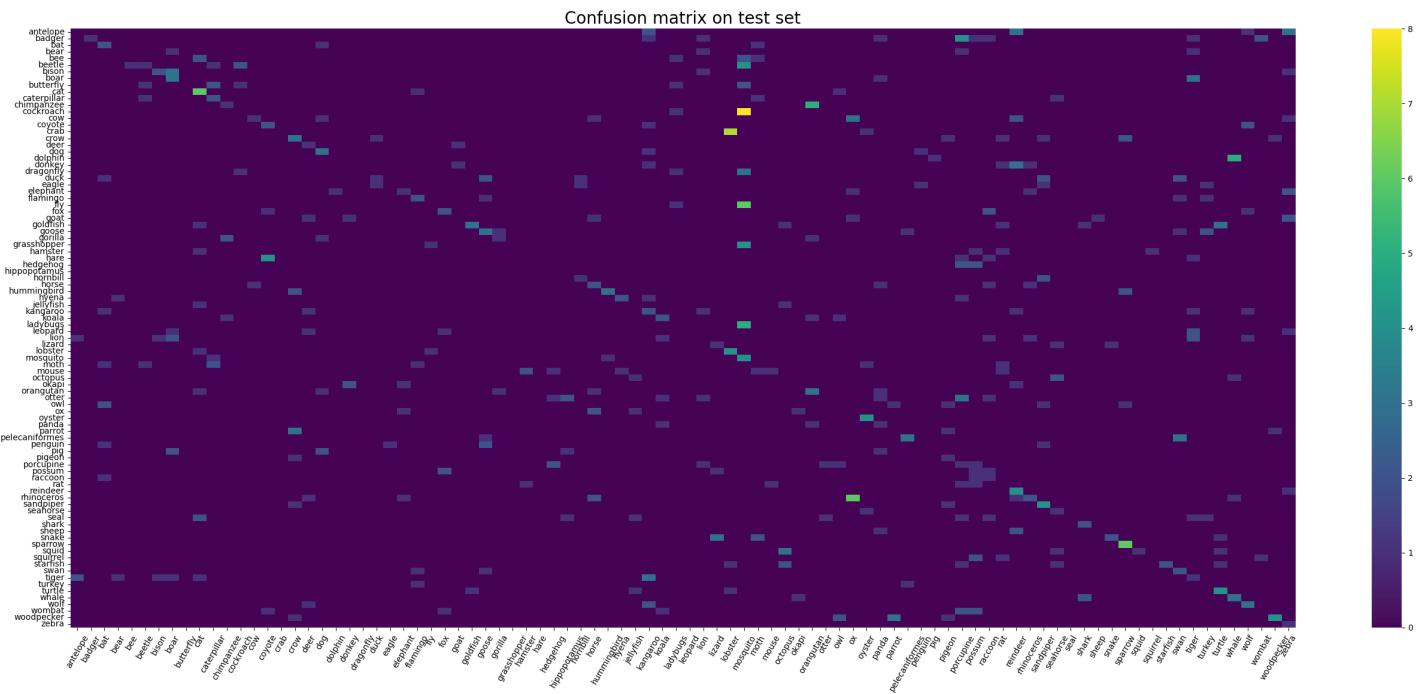


Figure 19: Konfúzna matica testovacích dát

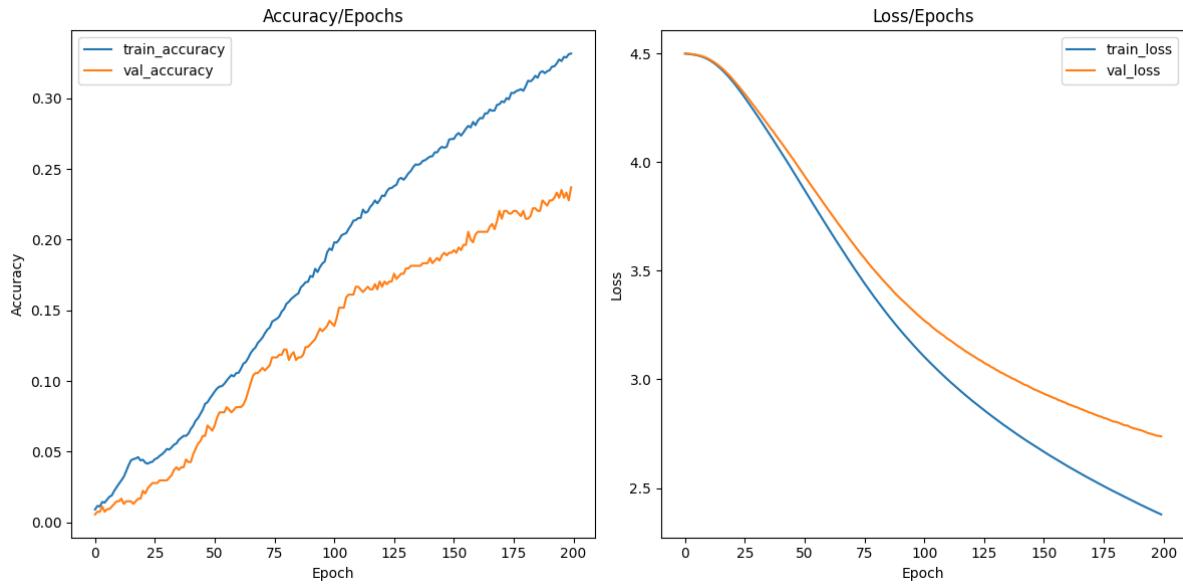


Figure 20: Priebeh trénovania

5 Prílohy

5.1 Celý výpis predikcií modelu EfficientNetB4 na našich dátach

```
1 antelope:[('gazelle', 0.5), ('impala', 0.33), ('hartebeest', 0.12)]
2 badger:[('badger', 0.96), ('brown_bear', 0.02), ('skunk', 0.02)]
3 bat:[('weasel', 0.1), ('howler_monkey', 0.06), ('walking_stick', 0.06)]
4 bear:[('brown_bear', 0.69), ('American_black_bear', 0.27), ('red_wolf', 0.02)]
5 bee:[('bee', 0.9), ('honeycomb', 0.08), ('cardoon', 0.02)]
6 beetle:[('rhinoceros_beetle', 0.32), ('ground_beetle', 0.21), ('leaf_beetle', 0.15)]
7 bison:[('bison', 0.96), ('ox', 0.02), ('black_grouse', 0.02)]
8 boar:[('wild_boar', 0.9), ('hog', 0.08), ('warthog', 0.02)]
9 butterfly:[('monarch', 0.47), ('lycaenid', 0.2), ('ringlet', 0.16)]
10 cat:[('Egyptian_cat', 0.46), ('tabby', 0.25), ('tiger_cat', 0.08)]
11 caterpillar:[('centipede', 0.25), ('lacewing', 0.12), ('leaf_beetle', 0.06)]
12 chimpanzee:[('chimpanzee', 0.98), ('orangutan', 0.02)]
13 cockroach:[('cockroach', 1.0)]
14 cow:[('ox', 0.82), ('Great_Dane', 0.08), ('Ibizan_hound', 0.04)]
15 coyote:[('coyote', 0.88), ('red_wolf', 0.04), ('dingo', 0.02)]
16 crab:[('Dungeness_crab', 0.47), ('rock_crab', 0.21), ('king_crab', 0.19)]
17 crow:[('magpie', 0.35), ('water_ouzel', 0.24), ('black_grouse', 0.14)]
18 deer:[('impala', 0.44), ('gazelle', 0.17), ('ibex', 0.17)]
19 dog:[('golden_retriever', 0.22), ('Labrador_retriever', 0.1), ('Chihuahua', 0.06)]
20 dolphin:[('dugong', 0.19), ('killer_whale', 0.15), ('grey_whale', 0.13)]
21 donkey:[('llama', 0.27), ('sorrel', 0.16), ('ox', 0.11)]
22 dragonfly:[('dragonfly', 0.82), ('damselfly', 0.16), ('lacewing', 0.02)]
23 duck:[('drake', 0.44), ('goose', 0.4), ('red-breasted_merganser', 0.08)]
24 eagle:[('bald_eagle', 0.92), ('kite', 0.08)]
25 elephant:[('African_elephant', 0.45), ('tusker', 0.28), ('Indian_elephant', 0.26)]
26 flamingo:[('flamingo', 0.94), ('rocking_chair', 0.04), ('spoonbill', 0.02)]
27 fly:[('fly', 0.93), ('cicada', 0.04), ('lacewing', 0.02)]
28 fox:[('red_fox', 0.77), ('kit_fox', 0.12), ('grey_fox', 0.08)]
29 goat:[('ibex', 0.24), ('ram', 0.16), ('Ibizan_hound', 0.16)]
30 goldfish:[('goldfish', 0.94), ('tench', 0.04), ('water_jug', 0.02)]
31 goose:[('goose', 0.96), ('black_stork', 0.02), ('kite', 0.02)]
32 gorilla:[('gorilla', 0.96), ('ski_mask', 0.02), ('chimpanzee', 0.02)]
33 grasshopper:[('grasshopper', 0.81), ('cricket', 0.17), ('mongoose', 0.02)]
34 hamster:[('hamster', 0.92), ('hare', 0.02), ('mousetrap', 0.02)]
35 hare:[('hare', 0.9), ('wood_rabbit', 0.08), ('Persian_cat', 0.02)]
36 hedgehog:[('porcupine', 0.94), ('weasel', 0.02), ('beaver', 0.02)]
37 hippopotamus:[('hippopotamus', 0.96), ('Mexican_hairless', 0.02), ('triceratops', 0.02)]
38 hornbill:[('hornbill', 0.96), ('toucan', 0.04)]
39 horse:[('sorrel', 0.75), ('Saluki', 0.1), ('ox', 0.04)]
40 hummingbird:[('hummingbird', 0.94), ('jacamar', 0.06)]
41 hyena:[('hyena', 1.0)]
42 jellyfish:[('jellyfish', 0.98), ('nipple', 0.02)]
43 kangaroo:[('wallaby', 0.86), ('gazelle', 0.06), ('impala', 0.04)]
44 koala:[('koala', 0.94), ('indri', 0.04), ('teddy', 0.02)]
45 ladybugs:[('ladybug', 0.91), ('leaf_beetle', 0.07), ('buckeye', 0.02)]
46 leopard:[('leopard', 0.86), ('snow_leopard', 0.06), ('jaguar', 0.06)]
47 lion:[('lion', 0.98), ('hartebeest', 0.02)]
48 lizard:[('agama', 0.2), ('whiptail', 0.16), ('banded_gecko', 0.14)]
49 lobster:[('American_lobster', 0.93), ('spiny_lobster', 0.07)]
```

```

50 mosquito:[('ant',0.18),('barn_spider',0.14),('tiger_beetle',0.12)]
51 moth:[('ringlet',0.32),('sulphur_butterfly',0.12),('long-horned_beetle'
      ,0.06)]
52 mouse:[('hamster',0.31),('wood_rabbit',0.22),('mousetrap',0.16)]
53 octopus:[('starfish',0.23),('electric_ray',0.15),('tailed_frog',0.08)]
54 okapi:[('zebra',0.35),('sorrel',0.25),('gazelle',0.17)]
55 orangutan:[('orangutan',1.0)]
56 otter:[('otter',0.98),('beaver',0.02)]
57 owl:[('great_grey_owl',0.5),('kite',0.19),('custard_apple',0.04)]
58 ox:[('ox',0.71),('oxcart',0.09),('triceratops',0.07)]
59 oyster:[('plate',0.22),('chiton',0.12),('hen-of-the-woods',0.1)]
60 panda:[('giant_panda',1.0)]
61 parrot:[('macaw',0.59),('lorikeet',0.14),('African_grey',0.08)]
62 pelecaniformes:[('pelican',0.87),('little_blue_heron',0.04),('black_stork'
      ,0.04)]
63 penguin:[('king_penguin',0.98),('partridge',0.02)]
64 pig:[('hog',0.96),('wild_boar',0.02),('Mexican_hairless',0.02)]
65 pigeon:[('black_grouse',0.26),('partridge',0.23),('European_gallinule'
      ,0.17)]
66 porcupine:[('porcupine',1.0)]
67 possum:[('polecat',0.24),('badger',0.18),('weasel',0.12)]
68 raccoon:[('grey_fox',0.21),('badger',0.15),('polecat',0.1)]
69 rat:[('hamster',0.43),('mink',0.14),('mousetrap',0.12)]
70 reindeer:[('ibex',0.67),('gazelle',0.06),('dogsled',0.06)]
71 rhinoceros:[('warthog',0.24),('Indian_elephant',0.18),('triceratops',0.18)]
72 sandpiper:[('red-backed_sandpiper',0.69),('redshank',0.27),('dowitcher'
      ,0.04)]
73 seahorse:[('nematode',0.13),('sea_cucumber',0.13),('sea_slug',0.13)]
74 seal:[('sea_lion',0.88),('otter',0.08),('bluetick',0.02)]
75 shark:[('great_white_shark',0.49),('tiger_shark',0.42),('hammerhead',0.07)]
76 sheep:[('ram',0.67),('kelpie',0.08),('standard_poodle',0.02)]
77 snake:[('Indian_cobra',0.27),('night_snake',0.12),('thunder_snake',0.1)]
78 sparrow:[('brambling',0.79),('house Finch',0.06),('partridge',0.04)]
79 squid:[('electric_ray',0.16),('crayfish',0.08),('isopod',0.08)]
80 squirrel:[('fox_squirrel',0.98),('comic_book',0.02)]
81 starfish:[('starfish',0.96),('isopod',0.02),('nematode',0.02)]
82 swan:[('goose',0.62),('black_swan',0.15),('American_egret',0.08)]
83 tiger:[('tiger',0.96),('sea_snake',0.02),('snow_leopard',0.02)]
84 turkey:[('prairie_chicken',0.33),('hen',0.27),('black_grouse',0.15)]
85 turtle:[('loggerhead',0.72),('box_turtle',0.11),('terrapin',0.09)]
86 whale:[('grey_whale',0.67),('killer_whale',0.1),('dugong',0.1)]
87 wolf:[('timber_wolf',0.77),('white_wolf',0.11),('red_wolf',0.06)]
88 wombat:[('wombat',0.98),('koala',0.02)]
89 woodpecker:[('black_grouse',0.23),('bee_eater',0.15),('jacamar',0.13)]
90 zebra:[('zebra',0.98),('hartebeest',0.02)]

```

Listing 4: Predikcie modelu EfficientNetB4

5.2 Classification report

1	---	precision	recall	f1-score	support
2	TRAIN				
3					
4					
5	0	0.81	0.92	0.86	48
6	1	0.91	0.98	0.94	50
7	2	1.00	0.70	0.82	50
8	3	0.89	0.92	0.90	51
9	4	0.94	1.00	0.97	50
10	5	0.96	0.91	0.93	47
11	6	0.90	0.96	0.93	49
12	7	0.78	0.90	0.83	48
13	8	0.89	0.91	0.90	45
14	9	0.96	0.92	0.94	52
15	10	1.00	0.81	0.90	48
16	11	0.89	0.98	0.93	50
17	12	0.96	0.96	0.96	50
18	13	0.76	0.88	0.81	50
19	14	0.83	0.92	0.87	49
20	15	0.88	0.98	0.93	47
21	16	0.94	0.98	0.96	49
22	17	0.82	0.75	0.78	48
23	18	0.81	0.76	0.79	51
24	19	0.86	0.71	0.78	52
25	20	0.86	0.71	0.78	45
26	21	0.94	1.00	0.97	50
27	22	0.93	0.86	0.90	50
28	23	0.98	0.98	0.98	49
29	24	0.94	0.98	0.96	47
30	25	0.92	1.00	0.96	48
31	26	0.93	0.96	0.95	45
32	27	0.91	0.92	0.91	52
33	28	0.74	0.58	0.65	45
34	29	0.96	1.00	0.98	51
35	30	0.80	0.98	0.88	48
36	31	0.98	0.98	0.98	51
37	32	0.94	1.00	0.97	47
38	33	0.87	0.94	0.90	50
39	34	0.93	0.84	0.88	50
40	35	0.98	0.84	0.90	50
41	36	0.88	0.88	0.88	51
42	37	0.90	0.98	0.94	46
43	38	0.72	0.80	0.76	51
44	39	0.91	0.98	0.94	51
45	40	0.92	0.94	0.93	48
46	41	0.98	1.00	0.99	49
47	42	0.96	0.90	0.93	51
48	43	0.92	0.98	0.95	50
49	44	0.94	0.96	0.95	46
50	45	0.96	0.94	0.95	49
51	46	0.88	0.91	0.89	46
52	47	0.93	0.86	0.89	44
53	48	1.00	0.98	0.99	45
54	49	1.00	0.92	0.96	50
55	50	0.98	0.80	0.88	50
56	51	0.69	0.55	0.61	49

57	52	0.91	0.65	0.76	48
58	53	0.94	0.96	0.95	48
59	54	0.96	0.98	0.97	48
60	55	0.90	0.90	0.90	51
61	56	0.87	0.83	0.85	48
62	57	0.90	0.78	0.83	45
63	58	0.98	1.00	0.99	50
64	59	1.00	0.96	0.98	50
65	60	0.92	0.94	0.93	49
66	61	0.98	0.93	0.96	46
67	62	0.98	0.96	0.97	45
68	63	0.75	0.90	0.82	51
69	64	0.91	0.91	0.91	47
70	65	0.90	1.00	0.95	47
71	66	0.88	0.88	0.88	50
72	67	0.93	0.81	0.87	48
73	68	0.79	0.76	0.77	49
74	69	0.79	0.92	0.85	49
75	70	0.79	0.82	0.81	51
76	71	1.00	1.00	1.00	48
77	72	0.90	0.83	0.86	46
78	73	0.91	0.85	0.88	48
79	74	0.86	0.96	0.91	45
80	75	0.86	0.92	0.89	48
81	76	0.94	1.00	0.97	48
82	77	0.89	1.00	0.94	47
83	78	0.91	0.82	0.86	49
84	79	0.82	0.88	0.85	48
85	80	0.92	0.98	0.95	49
86	81	0.90	0.92	0.91	48
87	82	0.98	0.96	0.97	50
88	83	0.98	0.85	0.91	48
89	84	0.96	1.00	0.98	47
90	85	0.88	0.88	0.88	48
91	86	0.86	0.91	0.89	47
92	87	0.95	1.00	0.97	54
93	88	1.00	0.92	0.96	52
94	89	0.94	0.98	0.96	46
95					
96	accuracy			0.90	4374
97	macro avg	0.91	0.90	0.90	4374
98	weighted avg	0.91	0.90	0.90	4374

Listing 5: Dotrénovanie posledných vrstiev ImageNet siete TRAIN data

1	TEST	precision	recall	f1-score	support
2					
3					
4					
5	0	0.50	0.83	0.62	6
6	1	1.00	1.00	1.00	6
7	2	1.00	0.33	0.50	6
8	3	0.83	0.83	0.83	6
9	4	1.00	0.83	0.91	6
10	5	0.83	0.83	0.83	6
11	6	1.00	1.00	1.00	6
12	7	0.86	1.00	0.92	6
13	8	1.00	1.00	1.00	6
14	9	0.75	1.00	0.86	6
15	10	1.00	0.83	0.91	6
16	11	0.86	1.00	0.92	6
17	12	1.00	1.00	1.00	6
18	13	0.60	0.50	0.55	6
19	14	0.62	0.83	0.71	6
20	15	0.75	1.00	0.86	6
21	16	1.00	1.00	1.00	6
22	17	0.75	0.50	0.60	6
23	18	1.00	0.33	0.50	6
24	19	1.00	0.67	0.80	6
25	20	0.83	0.83	0.83	6
26	21	1.00	0.67	0.80	6
27	22	0.62	0.83	0.71	6
28	23	1.00	1.00	1.00	6
29	24	0.75	1.00	0.86	6
30	25	1.00	0.83	0.91	6
31	26	0.86	1.00	0.92	6
32	27	0.60	1.00	0.75	6
33	28	0.67	0.33	0.44	6
34	29	0.86	1.00	0.92	6
35	30	0.75	1.00	0.86	6
36	31	0.83	0.83	0.83	6
37	32	1.00	0.83	0.91	6
38	33	0.80	0.67	0.73	6
39	34	1.00	0.67	0.80	6
40	35	1.00	1.00	1.00	6
41	36	1.00	1.00	1.00	6
42	37	1.00	1.00	1.00	6
43	38	0.71	0.83	0.77	6
44	39	0.75	1.00	0.86	6
45	40	1.00	0.67	0.80	6
46	41	0.86	1.00	0.92	6
47	42	0.56	0.83	0.67	6
48	43	0.75	1.00	0.86	6
49	44	1.00	1.00	1.00	6
50	45	1.00	1.00	1.00	6
51	46	0.75	1.00	0.86	6
52	47	0.75	1.00	0.86	6
53	48	0.86	1.00	0.92	6
54	49	0.86	1.00	0.92	6
55	50	1.00	0.83	0.91	6
56	51	1.00	0.50	0.67	6
57	52	0.67	0.33	0.44	6
58	53	1.00	1.00	1.00	6

59	54	1.00	1.00	1.00	6
60	55	1.00	1.00	1.00	6
61	56	1.00	0.67	0.80	6
62	57	0.67	0.33	0.44	6
63	58	1.00	1.00	1.00	6
64	59	1.00	1.00	1.00	6
65	60	0.71	0.83	0.77	6
66	61	1.00	1.00	1.00	6
67	62	0.86	1.00	0.92	6
68	63	0.71	0.83	0.77	6
69	64	1.00	0.67	0.80	6
70	65	1.00	1.00	1.00	6
71	66	0.86	1.00	0.92	6
72	67	1.00	1.00	1.00	6
73	68	1.00	1.00	1.00	6
74	69	0.67	0.67	0.67	6
75	70	0.67	0.67	0.67	6
76	71	1.00	1.00	1.00	6
77	72	0.83	0.83	0.83	6
78	73	0.86	1.00	0.92	6
79	74	0.67	0.67	0.67	6
80	75	1.00	1.00	1.00	6
81	76	0.86	1.00	0.92	6
82	77	1.00	1.00	1.00	6
83	78	0.50	0.33	0.40	6
84	79	1.00	0.83	0.91	6
85	80	1.00	0.83	0.91	6
86	81	0.60	0.50	0.55	6
87	82	1.00	1.00	1.00	6
88	83	1.00	0.83	0.91	6
89	84	1.00	0.83	0.91	6
90	85	0.71	0.83	0.77	6
91	86	0.75	1.00	0.86	6
92	87	0.86	1.00	0.92	6
93	88	1.00	0.83	0.91	6
94	89	1.00	1.00	1.00	6
95					
96	accuracy			0.85	540
97	macro avg	0.87	0.85	0.85	540
98	weighted avg	0.87	0.85	0.85	540

Listing 6: Dotrénovanie posledných vrstiev ImageNet siete TEST data