Zadanie č. 3

Strojové učenie a neurónové siete

Bc. Miroslav Malíšek

Analýza zadania

Cieľom tohto zadania je práca s konvolučnými neurónovými sieťami. Našim problémom sú obrázky zvierať a ich priradenie k správnemu druhu. V prvej časti zadania si samotné obrázky zobrazíme a lepšie ich spoznáme. Následne použijeme model trénovaný na databáze ImageNet, ktorý nám pre každý obrázok vie dať zaradenie do tried, ktoré má on definované.

V druhej časti si vytvoríme vlastnú CNN.

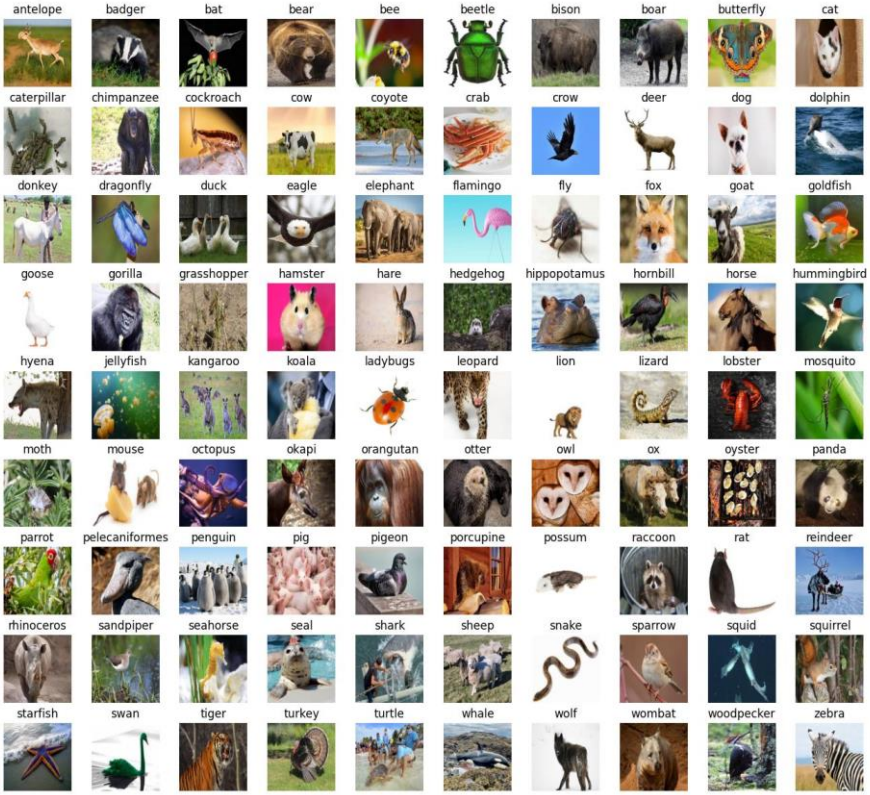
Tretia časť sa bude zaoberať transfer learningom. Znovu použijeme model trénovaný na databáze ImageNet. Najskôr si pomocou neho vygenerujeme príznaky pre každý obrázok a následne sa ich pomocou algoritmu učenia bez učiteľa pokúsime zhlukovať. K jednotlivým zhlukom si následne zobrazíme ich reprezentantov a taktiež priemerné obrázky.

Potom bude nasledovať prenesenie vedomostí na problém zadania, takže natrénovaný model dotrénujeme tak, aby sme získavali len také zaradenie do tried, ktoré máme definované.

1. Časť zadania (Spoznajte dáta)

Obrázky, ktoré sa použijú v tomto zadaní, sú rozdelené do dvoch množín (priečinkov) – trénovacie a testovacie. V oboch množinách sa nachádzajú rovnaké triedy, resp. typy zvierat. Celkovo ich je 90. Ku každému zo zvierat máme celkovo k dispozícii 60 obrázkov, z toho 54 je trénovacích a 6 testovacích. Spolu je to 5400 obrázkov. Tieto informácie som v kóde uložil do dataframe-u, aby to bolo prehľadné.

Nasledujúca koláž zobrazuje reprezentanta každej triedy. Zvolil som ho ako prvý obrázok každej triedy trénovacej množiny.



Analýza podľa modelu trénovaného na ImageNet

Ako model som si vybral EfficientNetB2, ktorý dosahuje úspešnosť 80.1%, resp. 94.9% v TOP-5. Zároveň obsahuje 9.2 miliónov parametrov, čo je rozumné číslo vzhľadom na dostupný hardvér (tento model použijeme aj v ďalších častiach zadania).

Predpovede získavam zo všetkých dostupných obrázkov (trénovacie aj testovacie), takže vyhodnotenie prebieha na 5400 obrázkoch.

Nakoľko model nie je prispôsobený na naše výstupy, nemohol som priamo porovnávať predikované zaradenie obrázku do triedy s reálny zaradením. Často sa totiž stalo, že model to vyhodnotil správne, akurát názov zvieraťa bol iný, ako u nás. Prípadne predikoval už aj poddruh zvieraťa. Ak by sme to priamo porovnávali, prišli by sme o cenné informácie, ktoré vieme získať len ručným vyhodnotením.

Vyhodnotil som to teda tak, že pre každú našu triedu som zobrazil 3 najpočetnejšie predikcie modelu (resp. menej ak model predikoval pre našu triedu všetko do jednej, resp. dvoch tried). Výsledky som uložil do dataframe-u, ktorý vyzerá nasledovne:

	True class	Predicted class	count
1	antelope	b'gazelle'	35
4	antelope	b'impala'	18
3	antelope	b'hartebeest'	4
6	badger	b'badger'	53
9	badger	b'howler_monkey'	2
11	badger	b'polecat'	2
21	bat	b'frilled_lizard'	6
28	bat	b'kite'	5
20	bat	b'fox_squirrel'	3
50	bear	b'brown_bear'	38
49	bear	b'American_black...	18
51	bear	b'fox_squirrel'	1
55	bee	b'bee'	49
59	bee	b'honeycomb'	5
56	bee	b'cardoon'	2
67	beetle	b'ground_beetle'	22
73	beetle	b'rhinoceros_bee...	20
70	beetle	b'leaf_beetle'	4
77	bison	b'bison'	57
76	bison	b'Bouvier_des_Fla...	1
78	bison	b'black_grouse'	1
82	boar	b'wild_boar'	50
80	boar	b'hog'	8
81	boar	b'warthog'	2
92	butterfly	b'monarch'	27
91	butterfly	b'lycaenid'	11
93	butterfly	b'ringlet'	6
99	cat	b'Egyptian_cat'	29
109	cat	b'tabby'	12
102	cat	b'Persian_cat'	4
118	caterpillar	b'centipede'	10
129	caterpillar	b'lacewing'	9
119	caterpillar	b'common_newt'	6
141	chimpanzee	b'chimpanzee'	58
142	chimpanzee	b'siamang'	1

	True class	Predicted class	count
143	chimpanzee	b'titi'	1
144	cockroach	b'cockroach'	58
145	cockroach	b'iron'	1
146	cockroach	b'isopod'	1
154	cow	b'ox'	42
147	cow	b'Great_Dane'	4
156	cow	b'sorrel'	3
160	coyote	b'coyote'	53
163	coyote	b'red_wolf'	2
158	coyote	b'Lakeland_terrier'	1
168	crab	b'king_crab'	18
169	crab	b'rock_crab'	18
165	crab	b'Dungeness_crab'	15
171	crow	b'black_grouse'	19
180	crow	b'water_ouzel'	11
177	crow	b'kite'	7
186	deer	b'gazelle'	19
190	deer	b'impala'	18
189	deer	b'ibex'	7
215	dog	b'golden_retriever'	11
195	dog	b'Border_collie'	5
202	dog	b'Labrador_retrie...	5
231	dolphin	b'grey_whale'	11
233	dolphin	b'killer_whale'	10
236	dolphin	b'tiger_shark'	9
250	donkey	b'sorrel'	10
246	donkey	b'llama'	8
247	donkey	b'ox'	7
255	dragonfly	b'dragonfly'	46
254	dragonfly	b'damselfly'	13
256	dragonfly	b'lacewing'	1
260	duck	b'drake'	22
261	duck	b'goose'	20
265	duck	b'red-breasted_m...	9
268	eagle	b'bald_eagle'	55

	True class	Predicted class	count
269	eagle	b'kite'	4
270	eagle	b'shopping_basket'	1
274	elephant	b'tusker'	28
271	elephant	b'African_elephant'	18
272	elephant	b'Indian_elephant'	13
276	flamingo	b'flamingo'	51
277	flamingo	b'hook'	2
278	flamingo	b'lakeside'	2
284	fly	b'fly'	55
283	fly	b'cicada'	3
282	fly	b'ant'	1
290	fox	b'red_fox'	46
287	fox	b'grey_fox'	7
289	fox	b'kit_fox'	4
303	goat	b'ibex'	14
296	goat	b'Ibizan_hound'	9
306	goat	b'ox'	6
312	goldfish	b'goldfish'	54
313	goldfish	b'tench'	3
311	goldfish	b'beaker'	2
317	goose	b'goose'	55
315	goose	b'black_stork'	2
316	goose	b'bustard'	1
322	gorilla	b'gorilla'	57
320	gorilla	b'Irish_water_spa...	1
321	gorilla	b'chimpanzee'	1
325	grasshopper	b'grasshopper'	47
324	grasshopper	b'cricket'	8
327	grasshopper	b'mongoose'	2
334	hamster	b'hamster'	46
337	hamster	b'polecat'	3
330	hamster	b'Angora'	2
341	hare	b'hare'	55
342	hare	b'wood_rabbit'	4
340	hare	b'Angora'	1

	True class	Predicted class	count
350	hedgehog	b'porcupine'	48
347	hedgehog	b'meerkat'	3
343	hedgehog	b'beaver'	1
355	hippopotamus	b'hippopotamus'	54
357	hippopotamus	b'triceratops'	3
354	hippopotamus	b'Mexican_hairless'	1
361	hornbill	b'hornbill'	56
362	hornbill	b'toucan'	2
359	hornbill	b'black_stork'	1
375	horse	b'sorrel'	38
367	horse	b'Saluki'	7
363	horse	b'Arabian_camel'	3
377	hummingbird	b'hummingbird'	57
378	hummingbird	b'jacamar'	3
381	hyena	b'hyena'	57
379	hyena	b'African_hunting...	1
380	hyena	b'dingo'	1
384	jellyfish	b'jellyfish'	56
386	jellyfish	b'tick'	2
383	jellyfish	b'isopod'	1
393	kangaroo	b'wallaby'	49
387	kangaroo	b'gazelle'	4
388	kangaroo	b'hare'	2
395	koala	b'koala'	56
394	koala	b'indri'	3
396	koala	b'teddy'	1
399	ladybugs	b'ladybug'	48
400	ladybugs	b'leaf_beetle'	8
397	ladybugs	b'European_fire_s...	1
404	leopard	b'leopard'	54
403	leopard	b'jaguar'	3
405	leopard	b'snow_leopard'	3
411	lion	b'lion'	55
406	lion	b'alp'	1
407	lion	b'brown_bear'	1

	True class	Predicted class	count
417	lizard	b'agama'	11
425	lizard	b'whiptail'	9
420	lizard	b'banded_gecko'	8
426	lobster	b'American_lobster'	54
429	lobster	b'spiny_lobster'	4
427	lobster	b'crayfish'	1
431	mosquito	b'barn_spider'	18
430	mosquito	b'ant'	9
439	mosquito	b'long-horned_be...	9
463	moth	b'ringlet'	15
465	moth	b'sulphur_butterfly'	6
460	moth	b'lycaenid'	4
476	mouse	b'hamster'	15
482	mouse	b'mousetrap'	13
486	mouse	b'wood_rabbit'	8
510	octopus	b'starfish'	14
495	octopus	b'electric_ray'	6
511	octopus	b'tailed_frog'	6
526	okapi	b'sorrel'	26
531	okapi	b'zebra'	9
517	okapi	b'Mexican_hairless'	5
533	orangutan	b'orangutan'	59
532	orangutan	b'barrow'	1
538	otter	b'otter'	51
536	otter	b'mink'	5
534	otter	b'beaver'	1
547	owl	b'great_grey_owl'	26
549	owl	b'kite'	7
553	owl	b'prairie_chicken'	6
570	ox	b'ox'	42
571	ox	b'oxcart'	4
562	ox	b'Arabian_camel'	2
592	oyster	b'plate'	12
586	oyster	b'hen-of-the-wo...	11
593	oyster	b'potpie'	5

	True class	Predicted class	count
600	panda	b'giant_panda'	59
599	panda	b'Bedlington_terri...	1
608	parrot	b'macaw'	30
607	parrot	b'lorikeet'	9
602	parrot	b'African_grey'	7
617	pelecanifor...	b'pelican'	50
613	pelecanifor...	b'black_stork'	5
612	pelecanifor...	b'American_egret'	1
621	penguin	b'king_penguin'	54
624	penguin	b'weasel'	2
619	penguin	b'Siberian_husky'	1
626	pig	b'hog'	54
628	pig	b'wild_boar'	4
625	pig	b'Mexican_hairless'	1
642	pigeon	b'partridge'	18
631	pigeon	b'black_grouse'	17
638	pigeon	b'kite'	7
650	porcupine	b'porcupine'	57
647	porcupine	b'baboon'	1
648	porcupine	b'fox_squirrel'	1
652	possum	b'badger'	10
655	possum	b'fox_squirrel'	7
654	possum	b'black-footed_fe...	5
674	raccoon	b'grey_fox'	15
671	raccoon	b'badger'	11
668	raccoon	b'Arctic_fox'	4
694	rat	b'mousetrap'	14
698	rat	b'wombat'	10
692	rat	b'mink'	8
708	reindeer	b'ibex'	32
707	reindeer	b'hartebeest'	6
704	reindeer	b'dogsled'	5
721	rhinoceros	b'warthog'	18
717	rhinoceros	b'hog'	11
719	rhinoceros	b'triceratops'	10

	True class	Predicted class	count
726	sandpiper	b'red-backed_san...	33
725	sandpiper	b'dowitcher'	14
727	sandpiper	b'redshank'	8
751	seahorse	b'sea_cucumber'	10
748	seahorse	b'nematode'	9
732	seahorse	b'banded_gecko'	5
766	seal	b'sea_lion'	48
765	seal	b'otter'	4
762	seal	b'ice_bear'	2
770	shark	b'great_white_sh...	30
774	shark	b'tiger_shark'	21
771	shark	b'hammerhead'	6
786	sheep	b'ram'	33
775	sheep	b'bighorn'	4
781	sheep	b'komondor'	4
791	snake	b'Indian_cobra'	19
799	snake	b'night_snake'	5
798	snake	b'horned_viper'	4
809	sparrow	b'brambling'	48
810	sparrow	b'bulbul'	2
814	sparrow	b'quail'	2
825	squid	b'electric_ray'	8
835	squid	b'isopod'	6
829	squid	b'goldfish'	4
850	squirrel	b'fox_squirrel'	57
851	squirrel	b'weasel'	2
849	squirrel	b'comic_book'	1
855	starfish	b'starfish'	57
852	starfish	b'mixing_bowl'	1
853	starfish	b'nematode'	1
859	swan	b'goose'	30
857	swan	b'black_swan'	15
856	swan	b'American_egret'	7
869	tiger	b'tiger'	56
865	tiger	b'Norwich_terrier'	1
866	tiger	b'barn_spider'	1
878	turkey	b'prairie_chicken'	27
871	turkey	b'black_grouse'	10
874	turkey	b'hen'	10
884	turtle	b'loggerhead'	39
883	turtle	b'leatherback_tur...	7
881	turtle	b'box_turtle'	5
890	whale	b'grey_whale'	32
892	whale	b'killer_whale'	8
889	whale	b'great_white_sh...	6
900	wolf	b'timber_wolf'	48
897	wolf	b'coyote'	5
901	wolf	b'white_wolf'	4
902	wombat	b'wombat'	60
914	woodpecker	b'hornbill'	11
907	woodpecker	b'black_grouse'	8
916	woodpecker	b'jacamar'	7
926	zebra	b'zebra'	57
924	zebra	b'cheetah'	2
925	zebra	b'kelpie'	1

Stĺpec vľavo reprezentuje naše triedy zvierat (pravdivé), stredný predpovedané modelom a stĺpec vpravo zobrazuje, koľkokrát model predpovedal hodnotu v strednom stĺpci pre pravdivú triedu zvierat vľavo.

Pre každé zviera sme mohli mať maximálny počet predpovedí 60, takže najlepšie je, ak je číslo vpravo práve 60. To sa podarilo v jedinom prípade a to pre zviera *wombat*. Iné predikcie teda pre toto zviera už model nedal. Pre dve zvieratá predpovedal model 59x rovnakú triedu, konkrétne pre *orangutan* a *panda*. V prípade pandy však vidíme, že model predpovedal „giant panda“, takže tento prípad potvrdzuje to, že sme to nemohli presne porovnávať. Vyšlo by nám, že pre pandu sa model netrafil ani raz, čo by nebola pravda. V rozmedzí 50 až 58 model predpovedal rovnakú triedu v týchto zvieratách: *badger, bison, boar, chimpanzee, cockroach, coyote, eagle, flamingo, fly, goldfish, goose, gorilla, hare, hippopotamus, hornbill, hummingbird, hyena, jellyfish, koala, leopard, lion, lobster, otter, pelican, penguin, pig, porcupine, squirrel, starfish, tiger, zebra.* Spolu teda model predpovedal 50 alebo viackrát rovnako pri 34 zvieratách.

Pri hlbšej analýze by sme ešte zistili, že sú zvieratá, pri ktorých model síce nepredpovedal až tak veľakrát jednu triedu, avšak predpovedal už aj druhy jednotlivých zvierat, ako napríklad v prípade *bear, crab, elephant* alebo *shark*. Ak by sme brali druhy ako jedno zviera, tak v týchto prípadoch by sme sa tiež dostali nad 50 správnych predpovedí.

Na druhú stranu, pri niektorých zvieratách sa model výrazne mýlil. Uvediem len tie prípady, v ktorých sa v troch najčastejších predpovediach nevyskytuje ani raz správna predpoveď. Sú to *bat cow, donkey, goat, mosquito, mouse, octopus, okapi, oyster, pigeon, possum, raccoon, rat, rhinoceros, seahorse, squid, woodpecker.* Je pravdepodobné, že model žiadne obrázky podobné týmto zvieratám predtým nevidel a teda nebol ich schopný správne zaradiť, prípadne nemá také triedy definované.

Medzi najviac podobné triedy podľa predpovedí patria *antelope – deer* (predpovede gazelle), *starfish – octoupus* (predpovede starfish), *badger – possum* (predpovede badger), *dolphin – whale* (predpovede gray whale a killer whale), *cow – ox* (predpovede ox). V niektorých prípadoch sú podobnosti zjavné (*antelope – deer, badger – possum*). V iných prípadoch je to ale najmä preto, lebo jedna z tých tried nie je na modeli natrénovaná a nemá svoju vlastnú triedu. Model ju teda priradil do inej triedy na základe nejakých znakov (*starfish – octopus* sú vodné zvieratá, *cow* nemá vlastnú triedu a najviac sa podobá z modelu práve na *ox*).

2. Časť zadania

V tejto časti vytvárame a trénujeme konvolučnú neurónovú sieť na klasifikáciu zvierat.

Na vytvorenie trénovacej, validačnej aj testovacej množiny použijeme funkciu image_dataset_from_directory() z knižnice Keras. Validačné dáta vytvoríme z trénovacej množiny v pomere 1:9. Veľkosti obrázkov nastavíme na 180x180 pixelov a batch size na 32. Trénovacie a validačné dáta zamiešame. Výstupné triedy budú vo forme Categorical, teda budú mať tvar ako po One – hot encodingu. Ako samotné obrázky použijeme tie dodané so zadaním, pričom vo funkcii sa využíva to, že sú uložené do priečinkov.

Nasledujúci kód teda slúži na vytvorenie generátoru dát:

```
train_ds_cnn = tf.keras.utils.image_dataset_from_directory(
    train_dir,
    label_mode="categorical",
    shuffle=True,
    validation_split=0.1,
    subset="training",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

val_ds_cnn = tf.keras.utils.image_dataset_from_directory(
    train_dir,
    label_mode="categorical",
    validation_split=0.1,
    subset="validation",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

test_ds_cnn = tf.keras.utils.image_dataset_from_directory(
    test_dir,
    shuffle=False,
    label_mode="categorical",
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)

class_names = train_ds_cnn.class_names

AUTOTUNE = tf.data.AUTOTUNE
train_ds_cnn = train_ds_cnn.cache().prefetch(buffer_size=AUTOTUNE)
val_ds_cnn = val_ds_cnn.cache().prefetch(buffer_size=AUTOTUNE)
```

Následne si vytvoríme konvolučnú neurónovú sieť. Bude obsahovať 2 konvolučné vrstvy, čo znamená 2 vrstvy Conv2D + MaxPooling2D.

Ešte pred prvou konvolučnou vrstvou normalizujeme vstupy do siete použitím preprocesnej vrstvy Rescaling. Samotné množiny teda upravovať normalizáciou nebudeme.

V prvej vrstve som zvolil počet filtrov 16 a kernelovú dimenziu 3x3. V druhej vrstve zase 32 filtrov, kernelová dimenzia ostala rovnaká. Aktivačná funkcia bude v oboch vrstvách relu. Maximálnu veľkosť poolingu som v oboch vrstvách zvolil 3x3.

Nasleduje vrstva Flatten, ktorá predchádza plne prepojeným vrstvám.

Plne prepojené vrstvy sú tvorené vrstvami typu Dense, pričom ich počet a aj počet neurónov v každej z nich už bude závisieť od konkrétnej realizácie siete. Budú tú uvedené rôzne konfigurácie. Avšak každá skrytá vrstva má aktivačnú funkciu relu a posledná, výstupná má aktivačnú funkciu softmax s počtom neurónov 90 (toľko máme rôznych druhov zvierat, resp. tried).

Ako kriteriálna funkcia je použitá categorical_crossentropy. Je to kvôli tomu, že sa jedná o viactriednu klasifikáciu a výstupom teda bude one – hot vector s predikciami. Solver bude Adam a metriku budeme sledovať úspešnosť.

Do modelu zahrnieme tiež early_stopping, ktorý bude monitorovať loss na validačnej množine a ak sa nebude znižovať po 3 za sebou idúcich epochách, tak sa trénovanie zastaví.

Po natrénovaní najlepšieho modelu si zobrazíme grafy priebehov trénovania na trénovacej aj validačnej množine. Pre trénovaciu a testovaciu množinu si zobrazíme aj konfúzne matice. Vzhľadom na vysokú početnosť tried sa budeme zameriavať iba na farebnosť diagonály.

Konfigurácie:

	Augmentácia	1. Conv2D	2. Conv2D	Dropout	Počet neurónov v skrytých vrstvách	Úspešnosť na trénovacej množine	Úspešnosť na testovacej množine
1.	-	-	-	-	256, 128	0.82	0.30
2.	-	L2(0.1)	L1(0.1)	-	256, 128	0.082	0.053
3.	Horizontal flip, rotation (0.1)	L2(0.01)	-	-	128	0.38	0.28
4.	Horizontal flip, rotation (0.1)	L2(0.1)	L2(0.001)	Dropout(0.2) po 1. Conv2D	128	0.25	0.20

V prvom trénovaní som nepoužil žiadne regularizátory a počet neurónov som sa snažil nájsť taký, aby vzniklo veľké pretrénovanie. Vidíme, že rozdiel v úspešnosti je viac ako 0.5 v prospech trénovacej množiny, takže model sa výrazne pretrénoval.

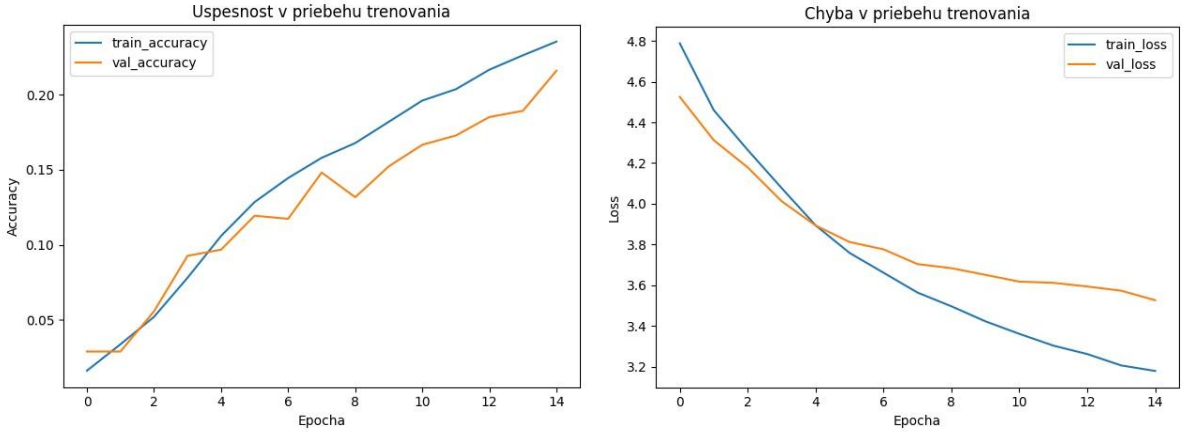
V druhom som neurónové vrstvy nechal nezmenené a zaviedol som 2 regularizátory – v prvej konvolučnej vrstve L2 (0.1) a v druhej L1 (0.1). Podľa úspešnosti ale vidíme, že táto kombinácia bola príliš silná a nedovolila sa modelu učiť. Po poslednej epoche sa úspešnosť na trénovacích dátach dostala iba na 0.082 a na testovacích 0.053.

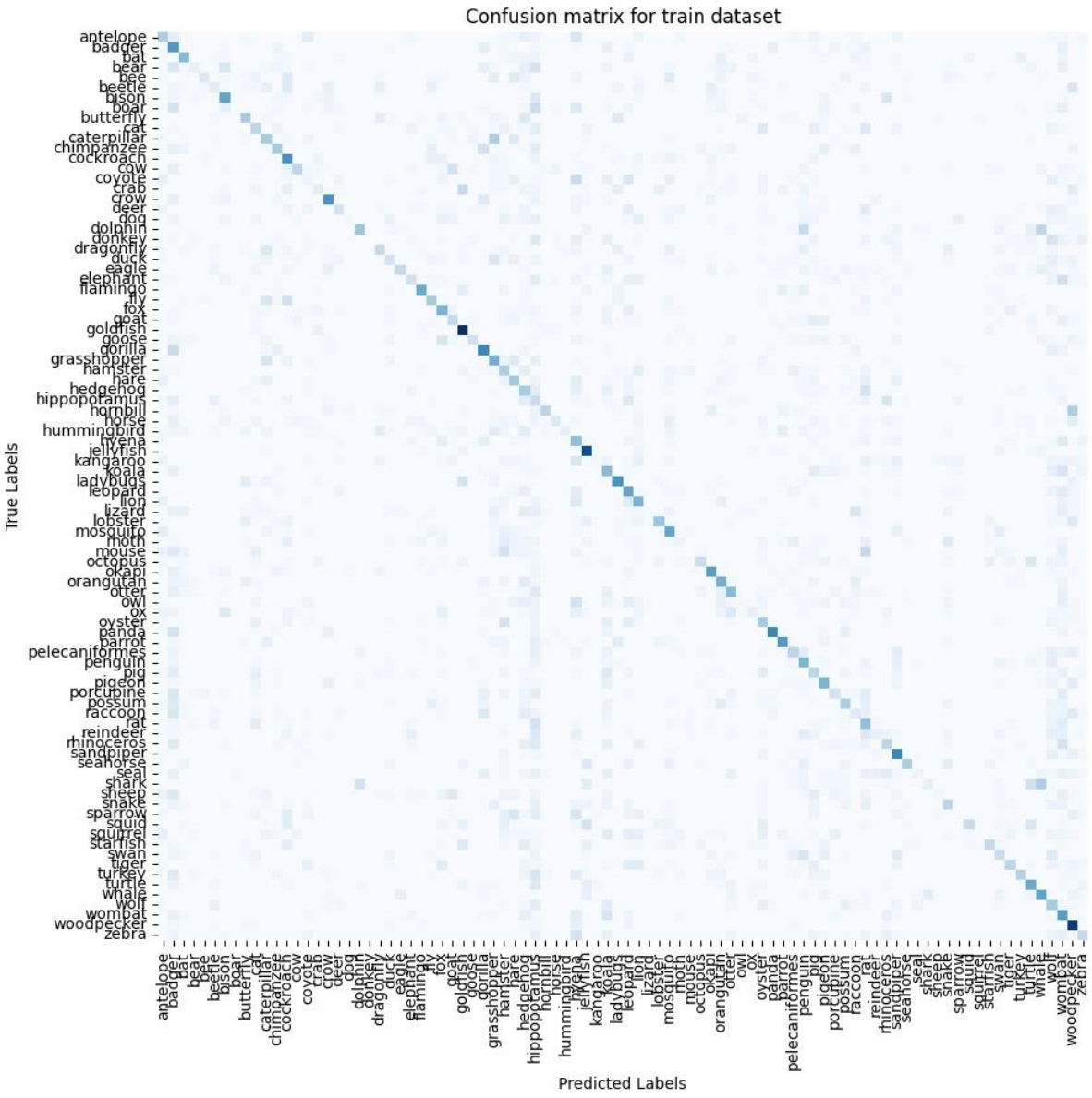
V treťom trénovaní som pridal tiež augmentáciu obrázkov ako prvú vrstvu konvolučnej siete. Konkrétne som aplikoval horizontálne otočenie a tiež rotáciu s faktorom 0.1. V prvej konvolučnej vrstve som zaviedol regularizátor L2(0.01). Okrem toho, počet skrytých vrstiev som znížil na jednu s počtom neurónov 128. Výsledkom bolo znížené pretrénovanie modelu.

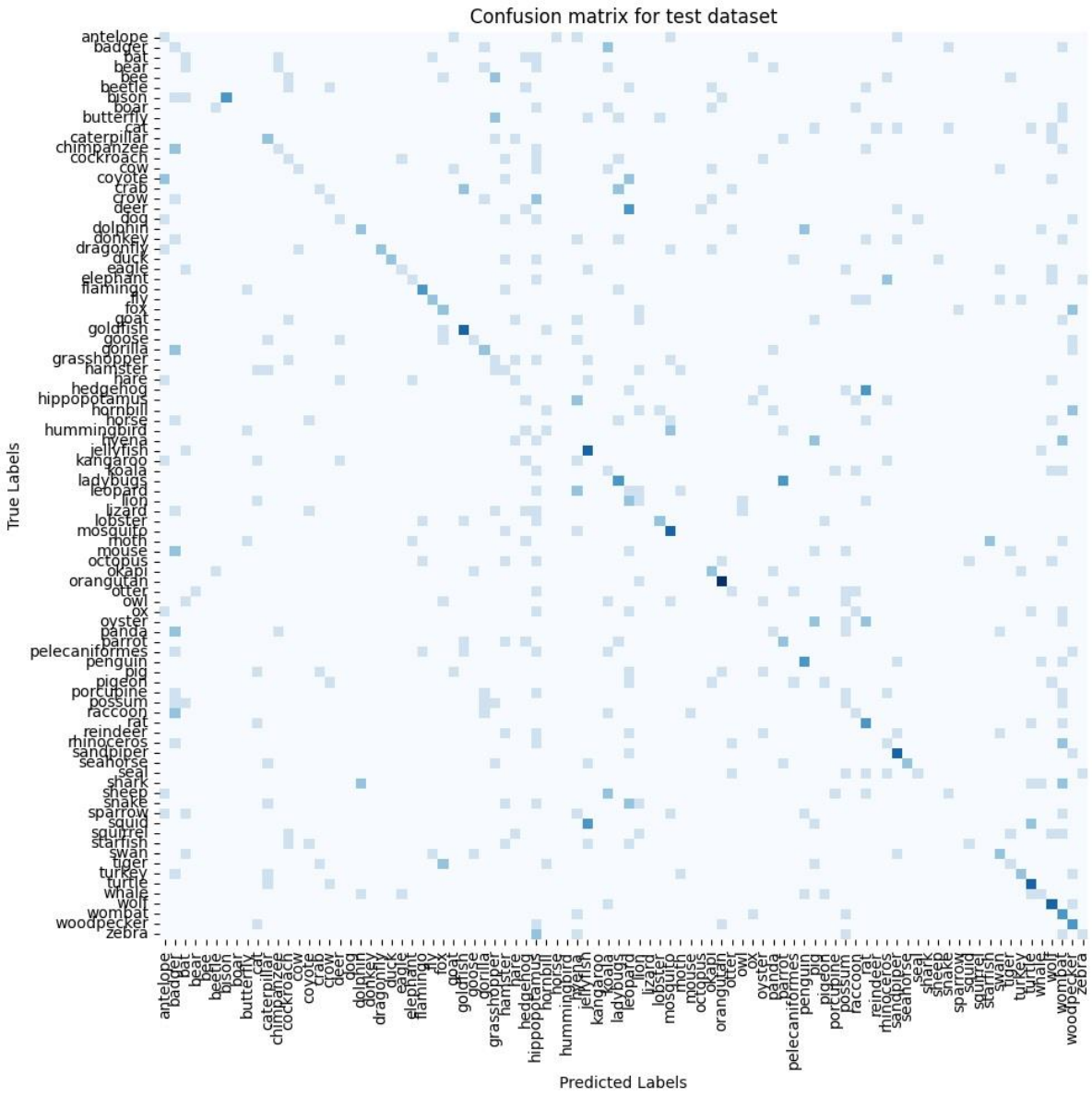
Na trénovacích dátach sa dosiahla úspešnosť 0.38 a na testovacích 0.28. Rozdiel je teda 0.1, čo sa blíži k prijateľnej hranici, ktorú by sme chceli dosiahnuť.

V poslednom štvrtom trénovaní som ponechal rovnakú augmentáciu, v prvej konvolučnej vrstve som použil L2(0.1) a v druhej L2(0.001). Taktiež som medzi prvú a druhú vrstvu vložil vypnutie neurónov Dropout v pomere 0.2. Plne prepojená vrstva ostala rovnaká ako v treťom trénovaní. Pri tejto konfigurácii sme dosiahli úspešnosť na trénovacej množine 0.25 a na testovacej 0.20. Úspešnosti síce neboli príliš vysoké, v predošlých sme dokázali získať lepšie výsledky, avšak tu sme výrazne minimalizovali pretrénovanie, čo bolo hlavným cieľom.

Nasledujúce obrázky sa viažu k najlepšiemu trénovaniu, ktoré je práve posledné, štvrté.




Confusion matrix for train dataset
True Labels
Predicted Labels
antelope
badger
bat
bear
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
turtle
whale
wolf
wombat
woodpecker
zebra



Prvé dva obrázky zobrazujú grafy úspešnosti a chyby na trénovacej a validačnej množine v priebehu trénovania. Vidíme, že rozdiely nie sú veľké a krivky sú k sebe blízko. S ďalšími epochami by úspešnosť na oboch množinách pravdepodobne stále rástla, avšak pri loss vidíme, že na trénovacích začína postupne klesať rýchlejšie ako v prípade validačných. Tento rozdiel by sa pravdepodobne ďalej prehlboval. Avšak v rámci 15 epoch je to takmer ideálny priebeh.

Ďalšie dva obrázky ukazujú konfúzne matice pre trénovacie a testovacie dáta. Na oboch je diagonála zreteľne viditeľná, takže tento cieľ sme splnili. Na trénovacích je však predsa len trochu viditeľnejšia, keďže úspešnosť na týchto dátach je o 0.05 vyššia. Nie je to však príliš veľký rozdiel, takže aj toto potvrdzuje, že model nemá takmer žiadne pretrénovanie.

3. Časť zadania

Zhlukovanie

Najskôr si uložíme príznaky do dataframe-u. Vytvoríme nové generátory dát, jeden pre trénovacie dáta a jeden pre testovacie (kvôli rozdeleniu do priečinkov) – na konci ich spojíme do jedného celku. Ani jednu množinu nemiešame, nie je to potrebné. Zároveň si z generátorov extrahujeme cesty k jednotlivým obrázkom a tiež výstupné triedy.

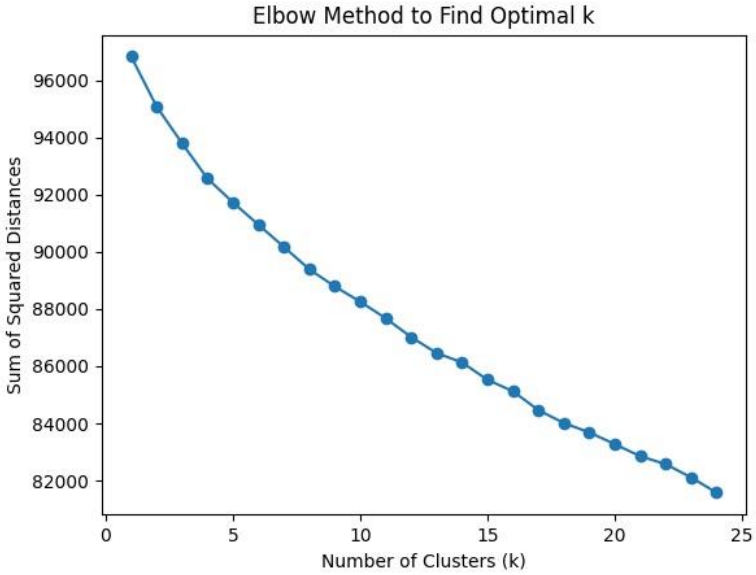
Predtrénovanú sieť použijeme rovnakú ako v prvej časti zadania, teda EfficientNetB2, akurát s tým rozdielom, že bez vrchnej klasifikačnej vrstvy.

Pre obe množiny dát zavoláme metódu predict, ktorá nám pre model bez vrchnej vrstvy vráti pre každý obrázok maticu príznakov o rozmere (6, 6, 1408). Následne funkcia GlobalAveragePooling2D pomocou priemeru konvertuje danú maticu na vektor dĺžky 1408. Takýto vektor získame pre všetky obrázky z oboch množín a uložíme ich ako riadky do dataframe-u. Okrem toho posledné dva stĺpce budú obsahovať cestu k obrázku, pre ktorý sme daný vektor príznakov získali a tiež jeho triedu. Na záver už len spojíme takéto dva dataframy pre trénovacie, resp. testovacie dáta do jedného spoločného dataframu. Tento dataframe teda bude mať 1410 stĺpcov (počet príznakov + cesta + trieda) a počet riadkov 5400.

Zhlukovací algoritmus

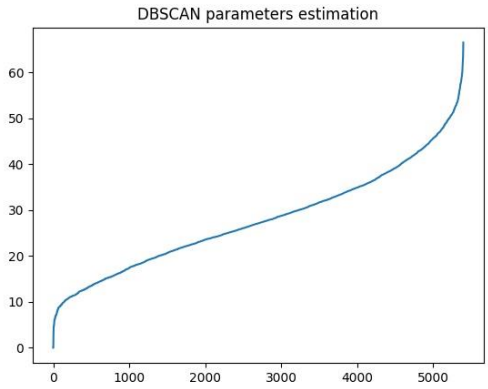
Vyberal som si medzi algoritmami K-means a DBSCAN. Pri oboch je potrebné určiť isté parametre, ktoré sa ale z našich dát určujú ťažko, keďže sú dosť abstraktné. Použil som teda všeobecné metódy, ktoré sa zvyknú používať na určenie parametrov.

Pri K-means je to parameter k, teda počet klastrov. Známa je Elbow metóda, v ktorej ide o to, že sa K-means vykoná pre všetky k z určitého intervalu a zistí sa kompaktnosť jednotlivých klustrov. Následne sa tieto hodnoty zobrazia do grafu a v ideálnom prípade by mal vyzerať ako „lakeť“ a hodnota v tomto lakti bude nové k. V našom prípade som však žiadny lakeť nevedel dostať. K som skúšal v intervale od 1 do 25. Skúšal som vstupné dáta bez úpravy, normalizovať, škálovať alebo redukovať pomocou PCA, avšak vo všetkých prípadoch vyšiel podobný graf ako tento:

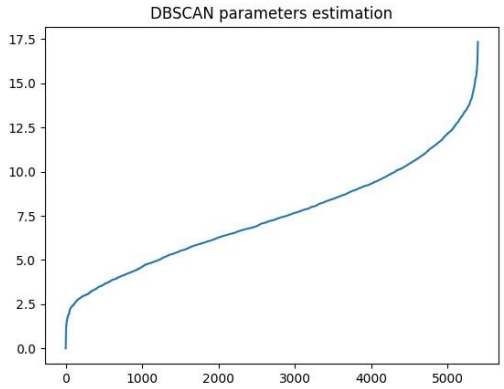


Nedá sa z neho presne určiť parameter k. Preto som musel skúsiť iný algoritmus, či pri ňom získam niečo relevantné.

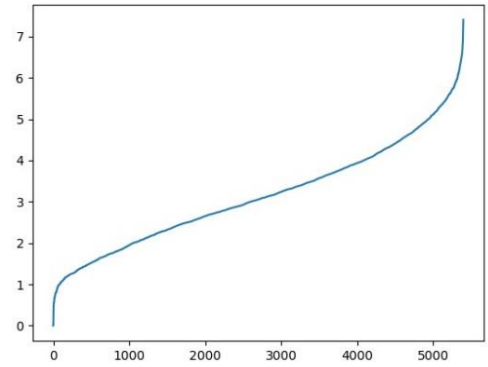
DBSCAN je tzv. density – based algoritmus, ktorý požaduje 2 parametre. Prvým je minimálny počet vzoriek (min_samples) potrebných pre vytvorenie klastra. Druhým je epsilon, ktorý určuje maximálnu vzdialenosť dvoch bodov, aby stále patrili do rovnakého klastra. Tu opäť existuje metóda, ktorá dokáže tieto parametre ukázať. Min_samples nastavíme na dvojnásobok dimenzie datasetu, čo je v našom prípade 2x1408 = 2816. Následne sa vypočíta priemerná vzdialenosť medzi každým bodom a jeho k najbližšími susedmi (k = 2816). Tieto vzdialenosti sa potom vzostupne zobrazia do grafu. Vznikne krivka, z ktorej určíme epsilon tak, že určíme bod najväčšieho zahnutia a hodnota na y osi bude hľadaným epsilonom. Opäť som tento postup vyskúšal na viacerých úpravách datasetu a takéto sú výsledky:



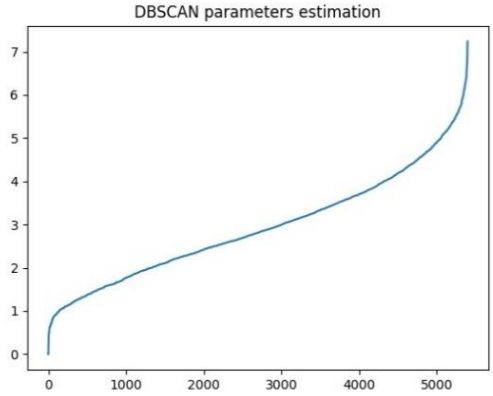
Neupravený dataset



Škálované vstupné dáta



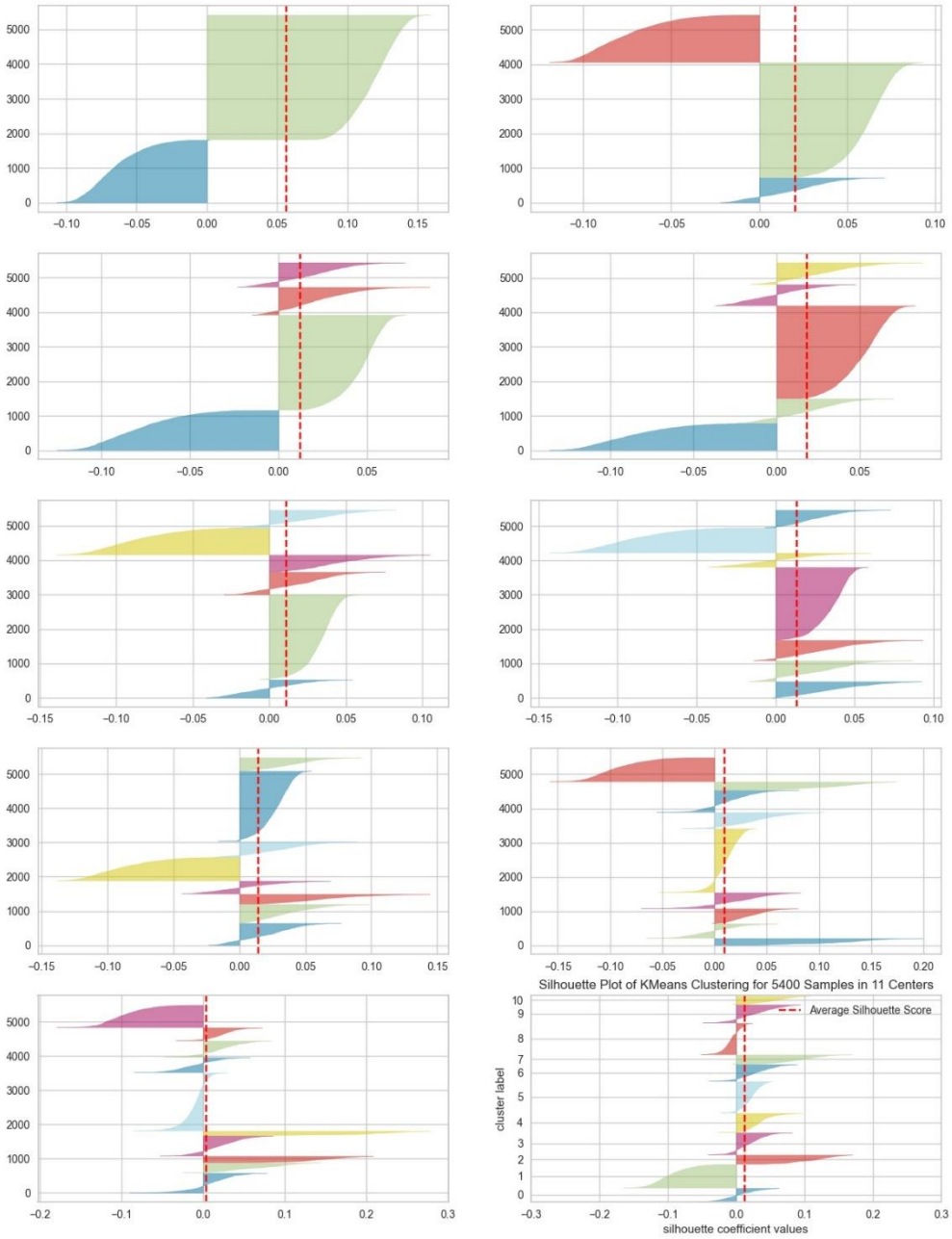
Normalizácia vstupných dát



Normalizácia + PCA (var 0.95)

Ako vidíme, grafy vyzerajú takmer identicky. Líšia sa len v hodnotách na osi y (vzdialenosť). Avšak, už len min_samples tým, že vyšiel 2816, tak to napovedá, že to asi nie je správny postup. Pri tejto metóde sa počíta s tým, že dimenzionalita nebude vzhľadom na počet vzoriek taká veľká, ako v našom prípade. Nepomôže tomu ani redukcia PCA, nakoľko ak chceme zachovať dostatočnú varianciu, tak dimenzii bude stále príliš veľa (viac ako 800). Túto metódu teda nemôžem použiť.

Tretia metóda sa viaže opäť ku K-means a volá sa Silhouette metóda. Táto metóda sa používa, ak nepomôže elbow metóda. Funguje na princípe tzv. Silhouette skóre, ktoré je vypočítané na základe intra klastrovej a inter klastrovej vzdialenosti. Knižnica Yellowbrick poskytuje implementáciu tejto metódy aj s výstupným grafom. Nasledujúci obrázok zobrazuje práve takýto graf, ktorý bol vytvorený na vstupných dátach bez úpravy.



Nachádza sa tu 10 grafov pre hodnoty k od 2 do 11. Ideálne rozdelenie je také, ak všetky siluety dosiahnu vyššie ako priemerné skóre a taktiež ak je ich výška približne rovnaká. Tomu sa najviac približuje posledný graf, teda pre k = 11.

Nakoľko k = 11 je jediný odhad akéhokoľvek parametru zhlukovacích algoritmov, ďalej v zadaní použijem práve algoritmus **K-means pre k = 11.** Vstupné dáta budú bez úpravy.

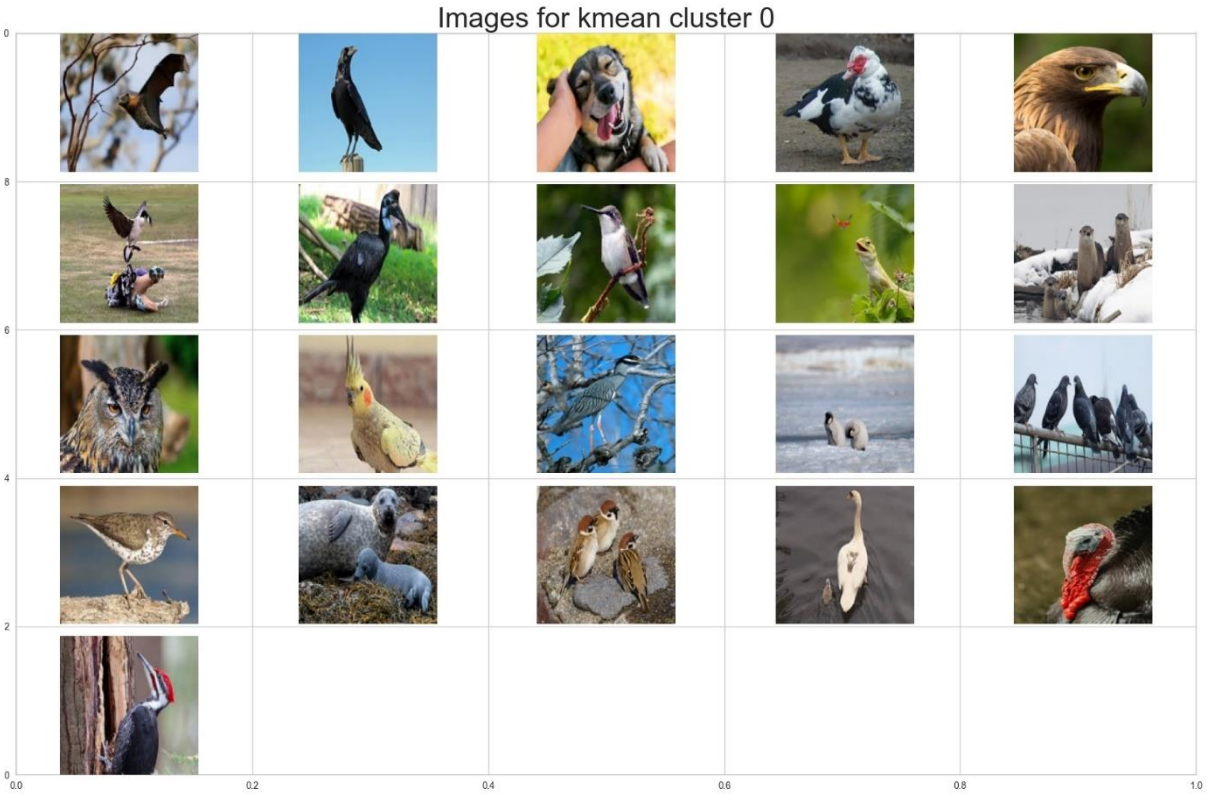
Podmnožiny obrázkov a priemerný obrázok pre každý zhluk

Implementácia algoritmu K-means knižnice Sklearn nám vráti taktiež číslo zhluku, do ktorého každý príznak zaradil. Tým pádom vieme, ktoré obrázky kam patria a môžeme si ich zobraziť.

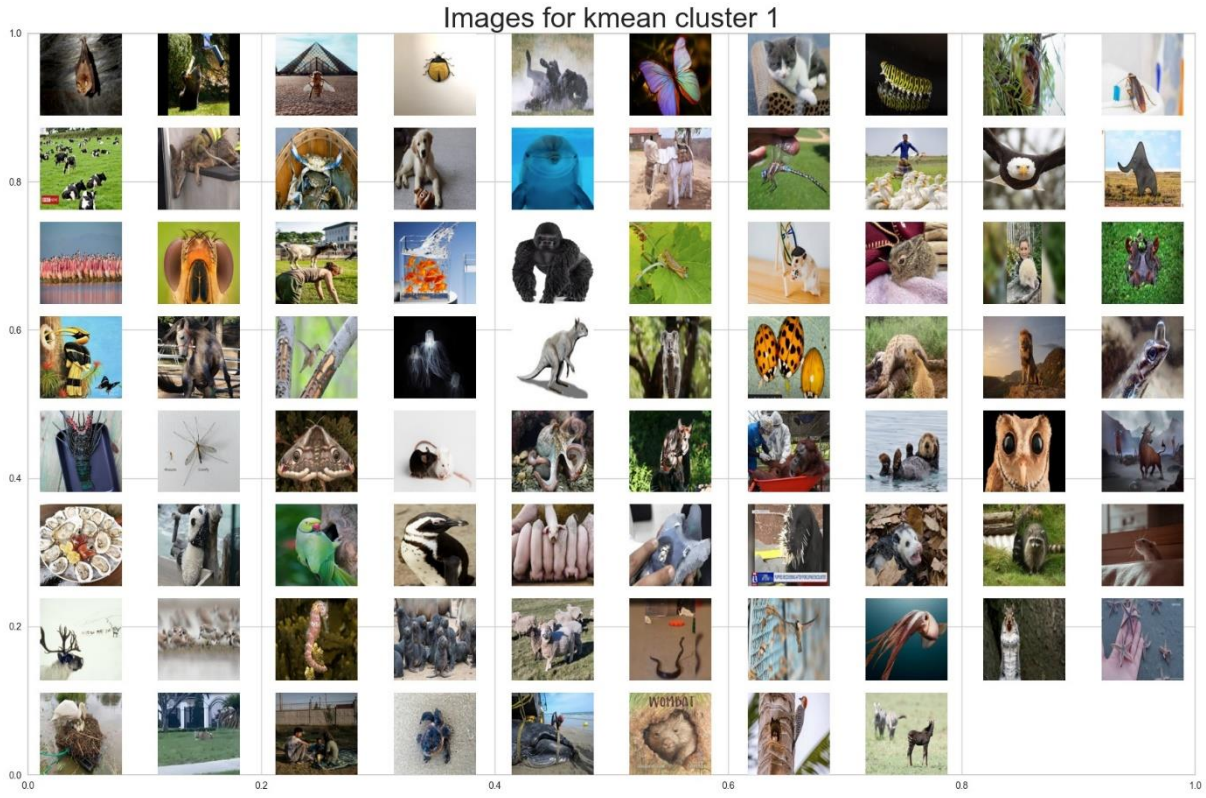
Obrázky som zobrazoval nasledujúcim spôsobom. Každý z 11 klastrov obsahoval viac rôznych reálnych tried, v niektorých prípadoch aj viac obrázkov rovnakej triedy. Preto som pre každú unikátnu reálnu triedu v každom zhluku zobrazil prvý obrázok v poradí. Na samotnom výbere obrázku z triedy by nemalo záležať, lebo predpokladáme, že vyzerajú podobne, najmä ak patria do rovnakého zhluku.

Celkovo teda budeme mať 11 veľkých obrázkov a každý bude obsahovať toľko obrázkov zvierat, koľko unikátnych reálnych tried k nemu algoritmus priradil.

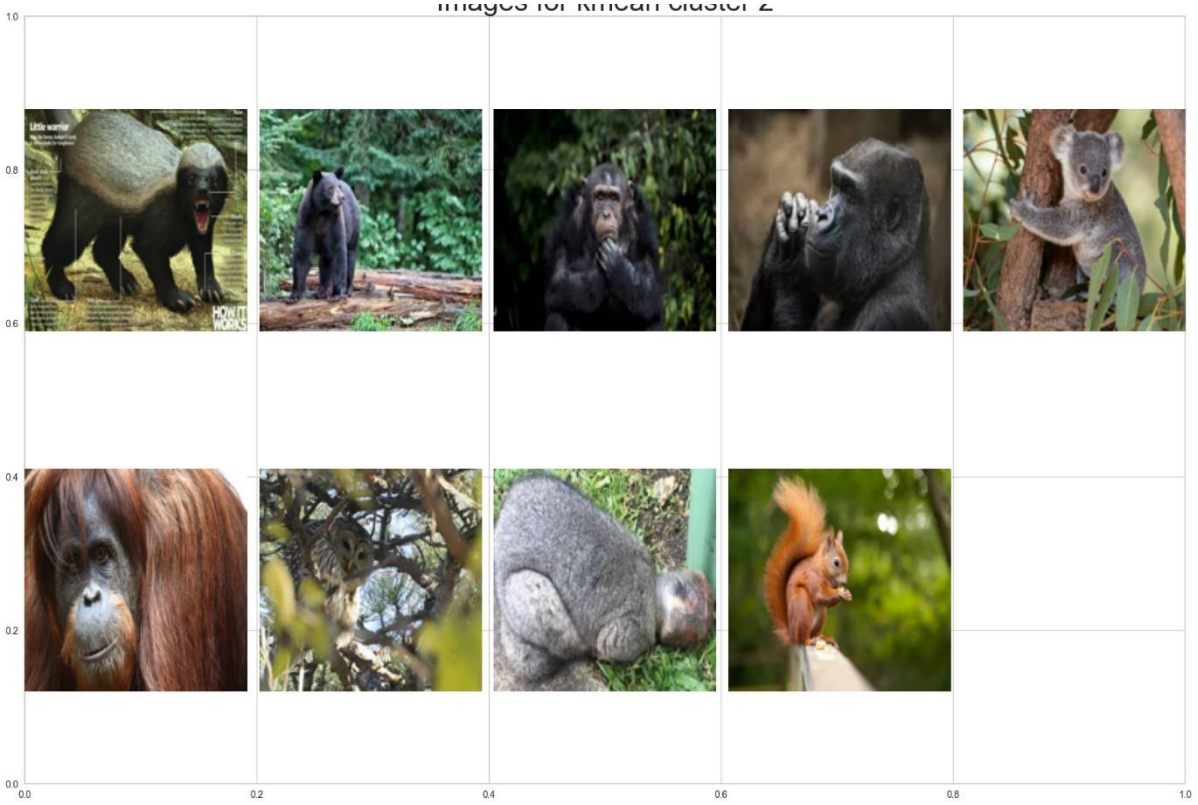
Okrem toho rovno uvádzam aj priemerné obrázky pre každý zhluk. Priemerný obrázok som počítal len z vybraných obrázkov pre jednotlivé klastre, ktoré sú na veľkých obrázkoch. Skúšal som to aj na všetkých obrázkoch, ktoré patria do jednotlivých klastrov, avšak vtedy sa z obrázkov nedalo rozpoznať takmer nič, lebo ho tvorilo príliš veľa rôznych obrázkov a vznikali väčšinou iba jednofarebné obrazy.



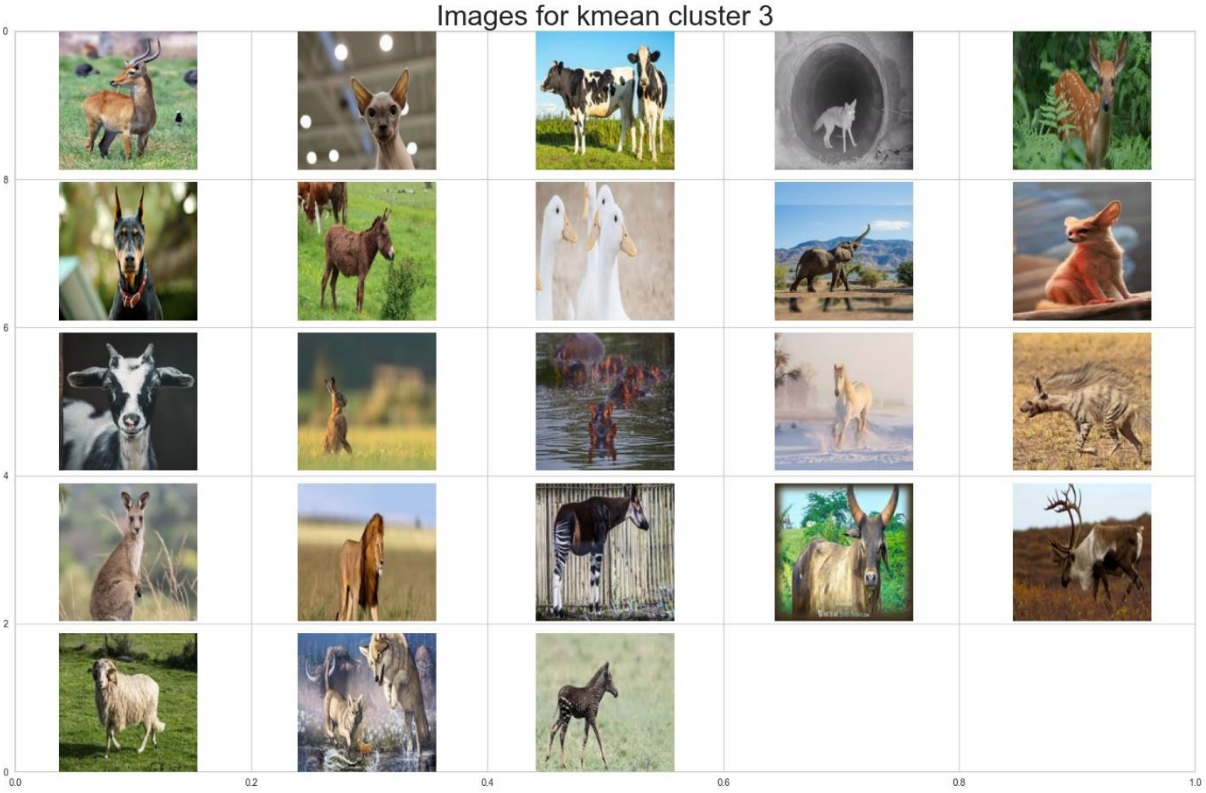
V tomto klastri sú prevažne vtáky. Z priemerného obrázku sa veľa vyčítať nedá, dá len rozpoznať obrys sovy a celkovo zelenšia farba.



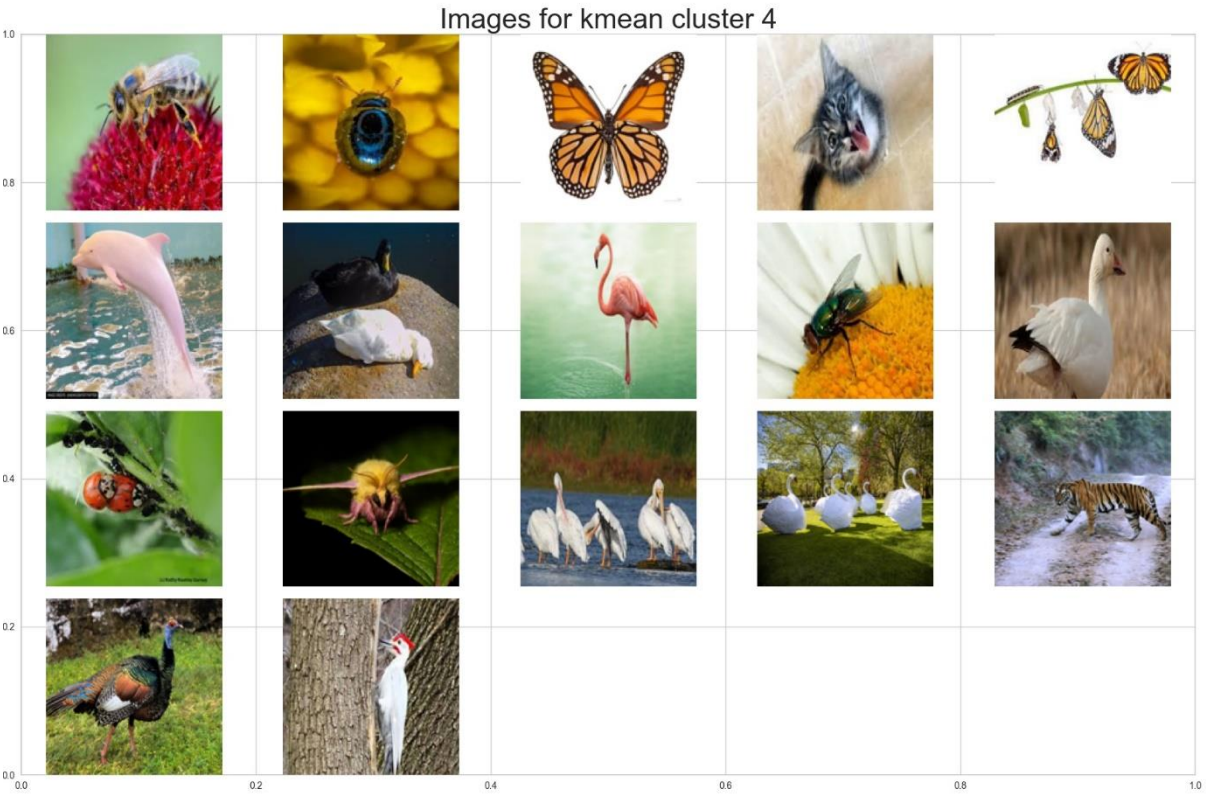
V tomto zhluku je celkom dosť obrázkov a nie je medzi nimi výraznejšia spoločná charakteristika, čo potvrdzuje aj priemerný obrázok, z ktorého sa nedá rozpoznať nič. Avšak dalo by sa povedať, že samotné zvieratá tvoria dosť veľkú časť obrázkov, tak možno to ich spája. Je v nich málo pozadia. Z priemerného sa však môže zdať, že tmavšie farby sa koncentrujú v centrálnej časti a svetlejšie v okrajových, čo znamená presný opak. Treba ale poznamenať, že priemerný obrázok je tvorený z veľkého počtu obrázkov, čo môže vytvoriť nečitateľný priemerný obrázok.



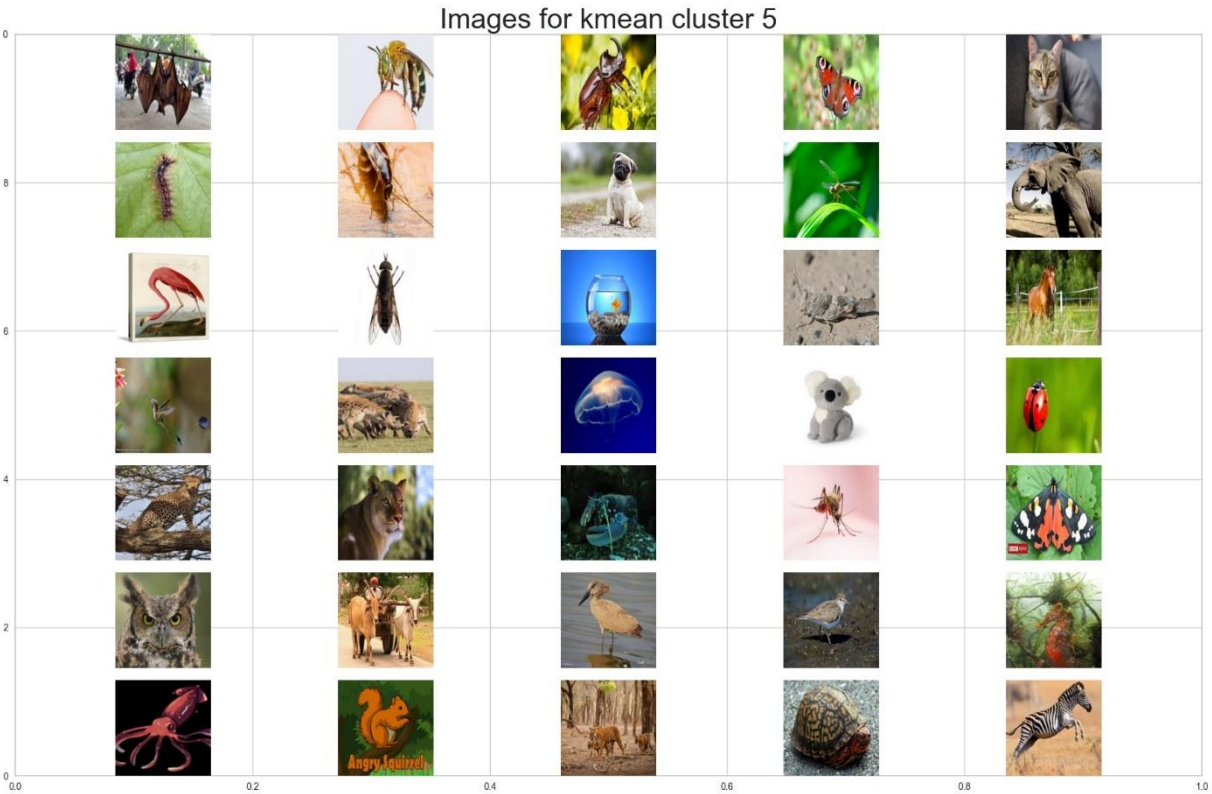
V tomto klastri je jasné, že tu prevládajú opice. Aj v prvej časti sme videli, že opice vedela táto sieť predpovedať veľmi dobre, a podobne to dokáže aj K-means. Prípadne je tam ešte spojitosť stromov a zelenej farby v pozadí. Priemerný obraz je dosť výrazne tmavý a tiež dosť tmavozelený. Takže to potvrdzuje spojitosť obrázkov.



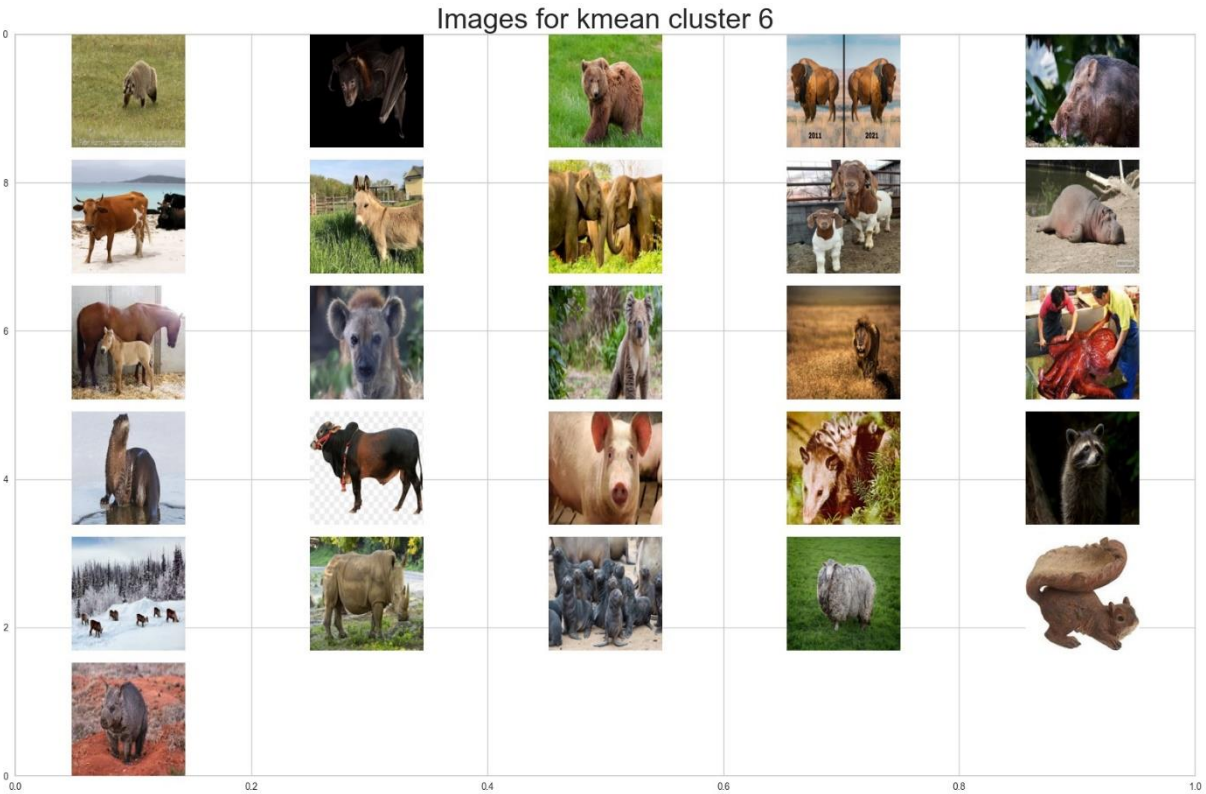
Tieto obrázky majú spoločné to, že sú na ňom hlavne štvornohé zvieratá, zvyčajne divo žijúce. Taktiež sú prevažne v prírode a v pozadí prevláda zelená farba. Z priemerného obrázku sa dá vidieť, že horná časť je viac modrá, čo by mohlo znamenať oblohu a spodok zelený, čo zase znamená trávu a prírodu. V strede sa koncentrujú samotné zvieratá.



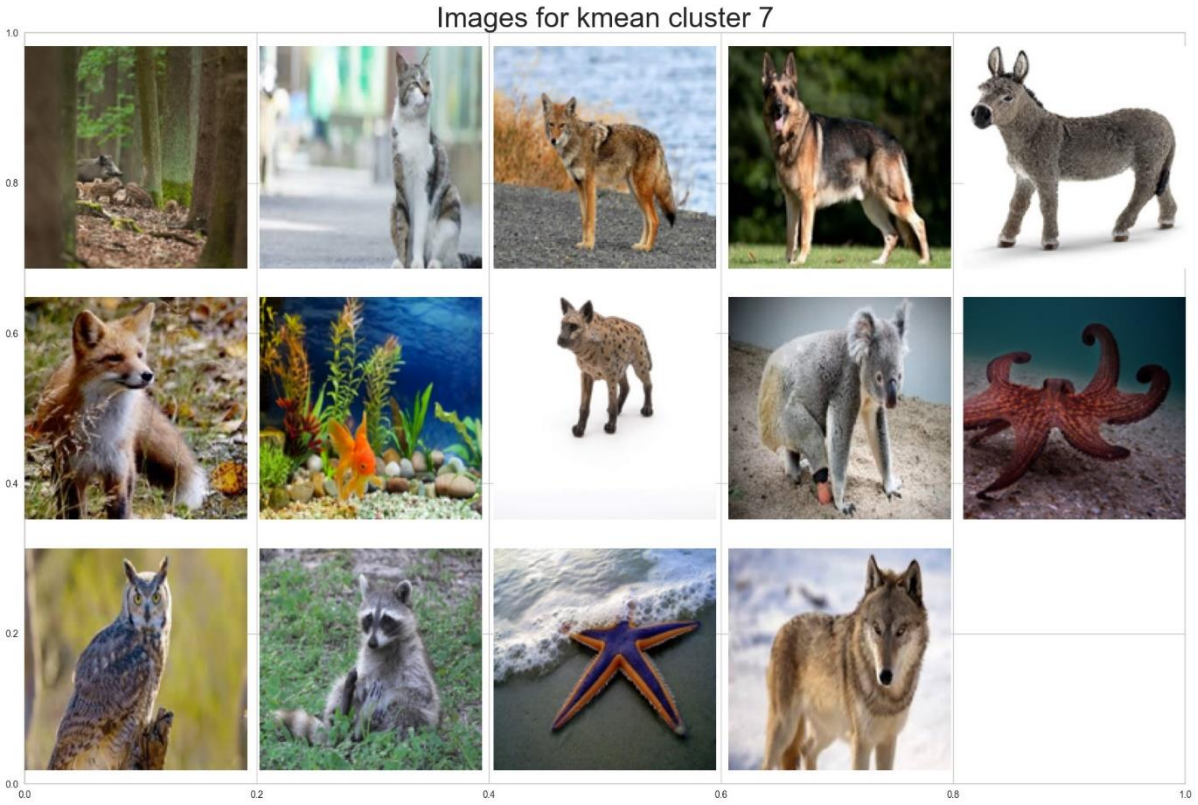
V tomto zhluku sa nachádzajú rôzne druhy hmyzu a vtákov. Pri niektorých sú viditeľné krídla, tak aj to môže byť spoločný znak. V priemernom obrázku dominuje motýľ s veľkými krídlami, ktorý zatieňuje všetko ostatné.



Aj v tomto zhluku sa dá povedať, že sa nachádza hlavne hmyz a vtáky. Je tu však aj viac iných obrázkov, ktoré sú bez výraznejšej spojitosti. V centre priemerného obrázku môžeme vidieť tvar podobný muche, čo potvrdzuje prítomnosť hmyzu.

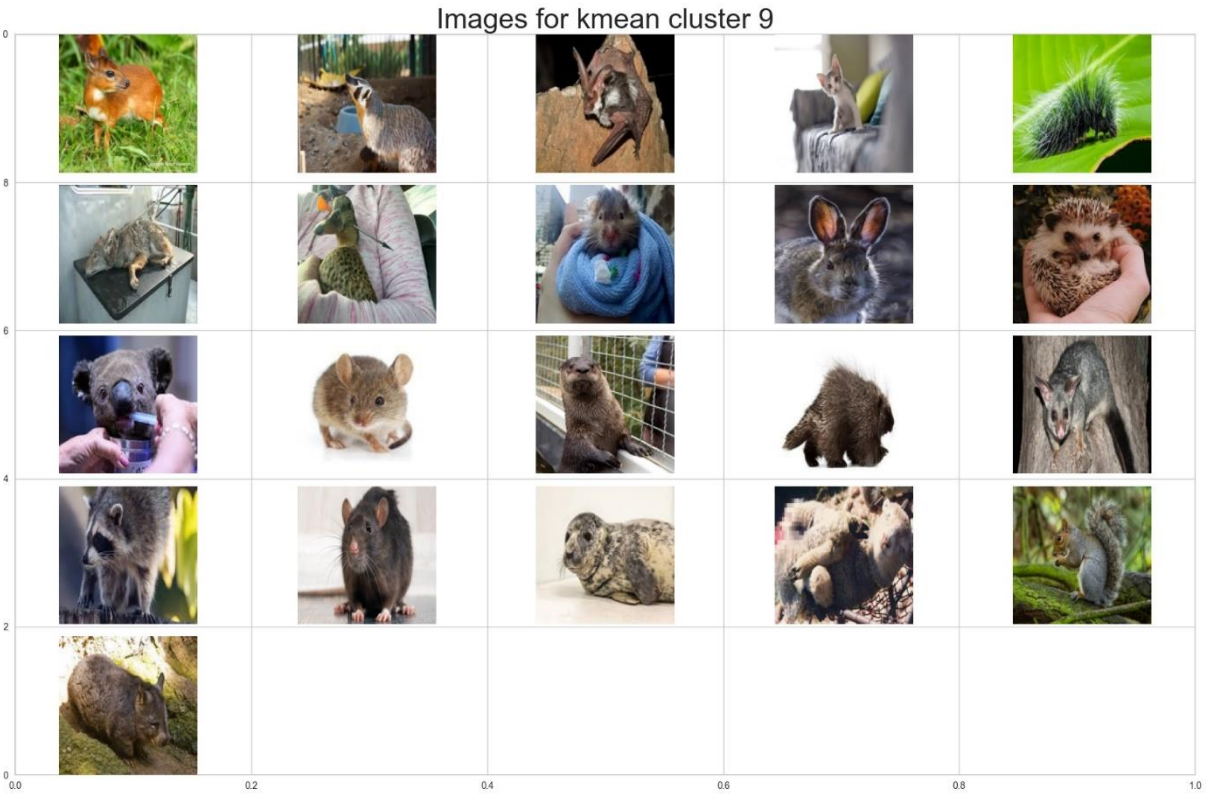


Tieto obrázky spája to, že sa na nich nachádzajú veľké, mohutné štvornohé zvieratá, ktoré sú zväčša v prírode. Pri bližšom pohľade na priemerný obrázok sa môže zdať, že jeho veľkú časť tvorí obrys zvieraťa podobného býkovi alebo zubrovi. Okolie je jednofarebné, takže pozadie tvorí väčšinou tráva a nie je tam obloha.

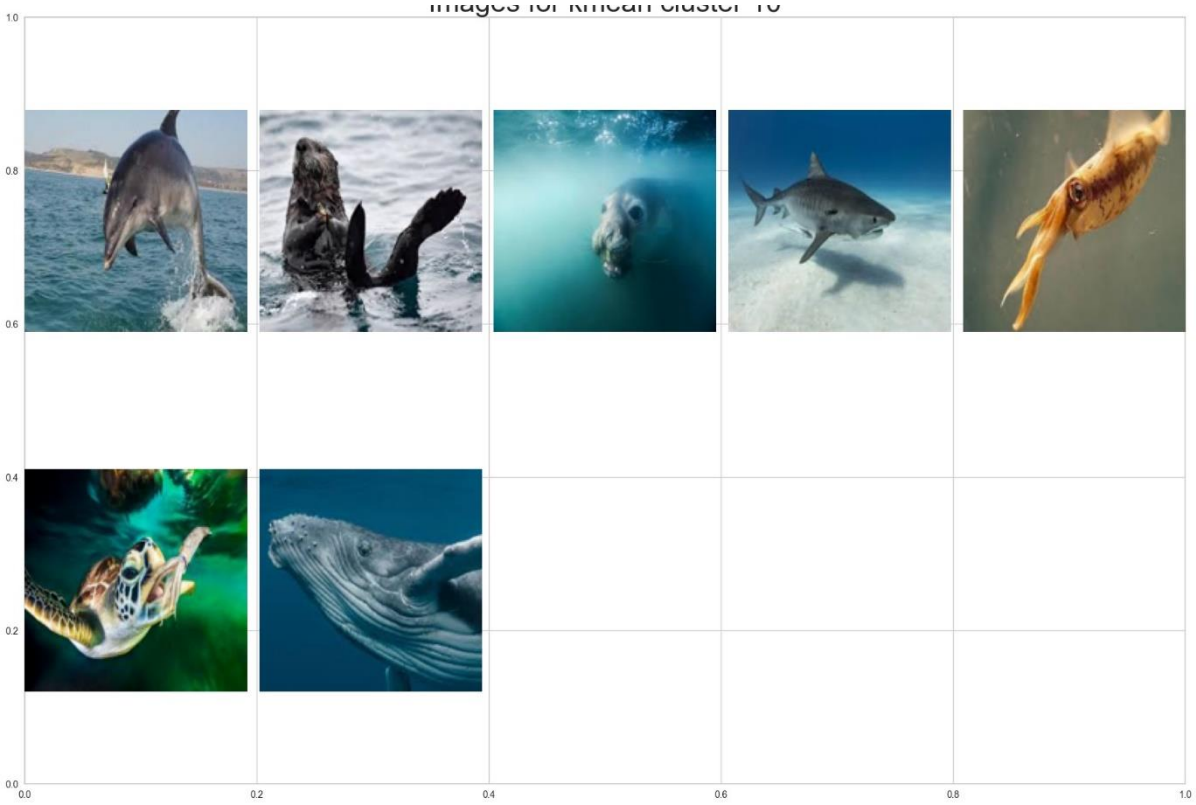


Na druhú stranu, v tomto zhluku sa nachádzajú hlavne menšie štvornohé zvieratá, ktoré sa podobajú na psovité šelmy. Potvrdzuje to aj priemerný obrázok, kde niektoré zvieratá celkom zreteľne vidíme. Okrem toho, v pravej hornej časti prevažuje modrá farba, čo celkom korešponduje s obrázkami.

V tomto klastri je veľa obrázkov a nedá sa určiť výraznejšia spoločná črta. Celkovo však obrázky pôsobia svetlejšie a majú živšie farby. Priemerný obrázok nám neukazuje nič. To by mohlo potvrdzovať, že tam výraznejšia spojitosť nie je. Avšak je tvorený celkom veľkým počtom obrázkov, čo tiež mohlo spôsobiť nejasnosť.



Do tohto zhluku patria menšie zvieratá, mláďatá a najmä hlodavce. Samotné zvieratá majú tmavšiu farbu. Potvrdzuje to aj priemerný obrázok, v ktorého centrálnej časti sú koncentrované zvieratá. Môžeme tam dokonca rozoznať aj nejaké hlavy. Okolie je celkom jasne oddelené.



Obrázky v tomto klastri sa dajú pekne určiť ako morské zvieratá. Pozadie je najmä modré kvôli vode. Aj priemerný obrázok je dosť výrazne modrý. Vzhľadom na nízky počet obrázkov, ktoré ho tvoria, môžeme jasne vidieť aj samotné zvieratá.

Transfer learning

Z obrázkov v priečinkoch si vytvoríme trénovaciu, validačnú a testovaciu množinu dát rovnakým spôsobom ako v druhej časti pri vytváraní vlastnej CNN. Ako model natrénovaný na ImageNet som si opäť zvolil EfficientNetB2, nakoľko dosahuje úspešnosť 80.1%, resp. 94.9% v TOP-5. Zároveň obsahuje 9.2 miliónov parametrov, čo je rozumné číslo vzhľadom na dostupný hardvér (tento model použijeme aj v ďalších častiach zadania). Opäť ho získame bez vrchnej klasifikačnej vrstvy, tentokrát preto, aby sme ju nahradili plne prepojenou klasifikačnou vrstvou na náš problém klasifikácie zvierat do 90 tried.

Následne konvolučné vrstvy zamrazíme, aby sme ich netrénovali. Takto môžeme použiť už natrénovaný model, čo zvýši presnosť a ušetrí čas.

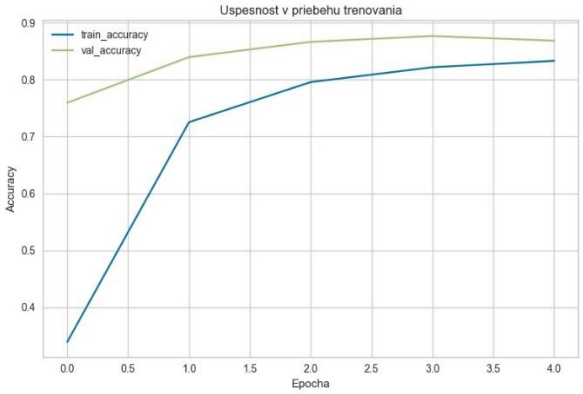
K takto pripravenému modelu môžeme pripojiť klasifikačnú vrstvu. Výstupná vrstva bude mať vždy počet neurónov 90 a aktivačnú funkciu Softmax. Skryté vrstvy sú nasledujúcej architektúry:

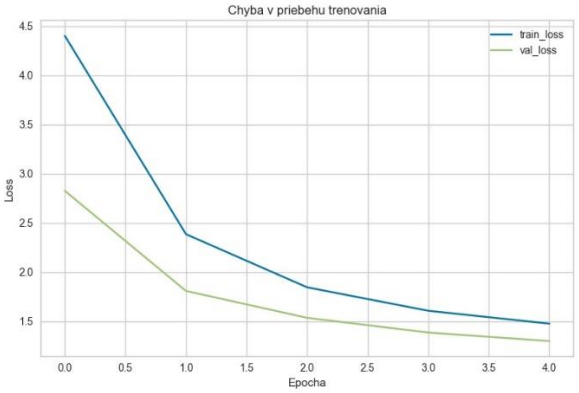
- Dropout(0.3)
- Dense(64, activation='relu', kernel_regularizer=l2(0.01))
- Dropout(0.2)

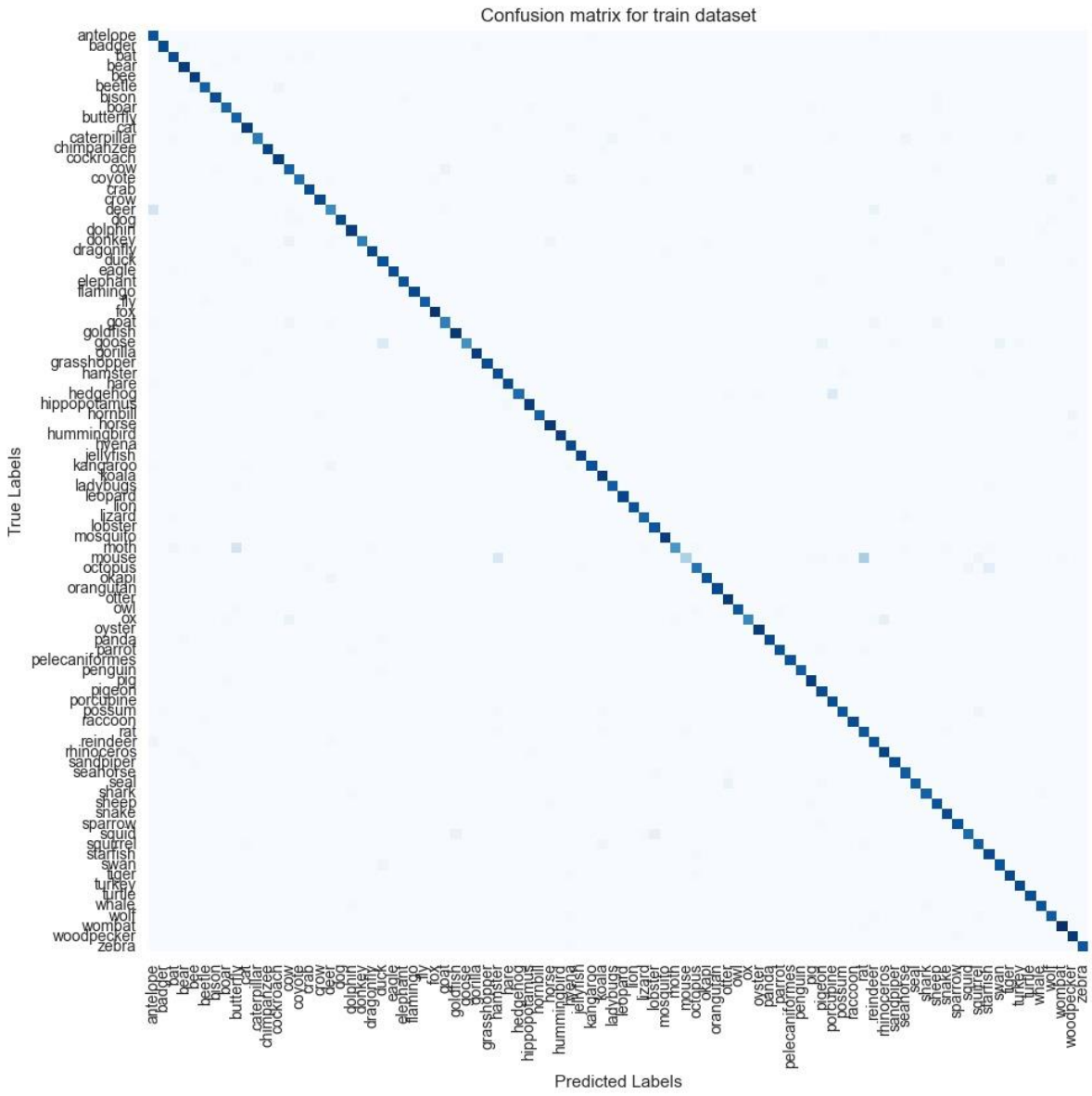
Model skompilujeme použitím solveru Adam a kriteriálnej funkcie Categorical Crossentropy. Zavedieme aj Early Stopping. Počet epoch som nastavil na 5, bolo to úplne postačujúce.

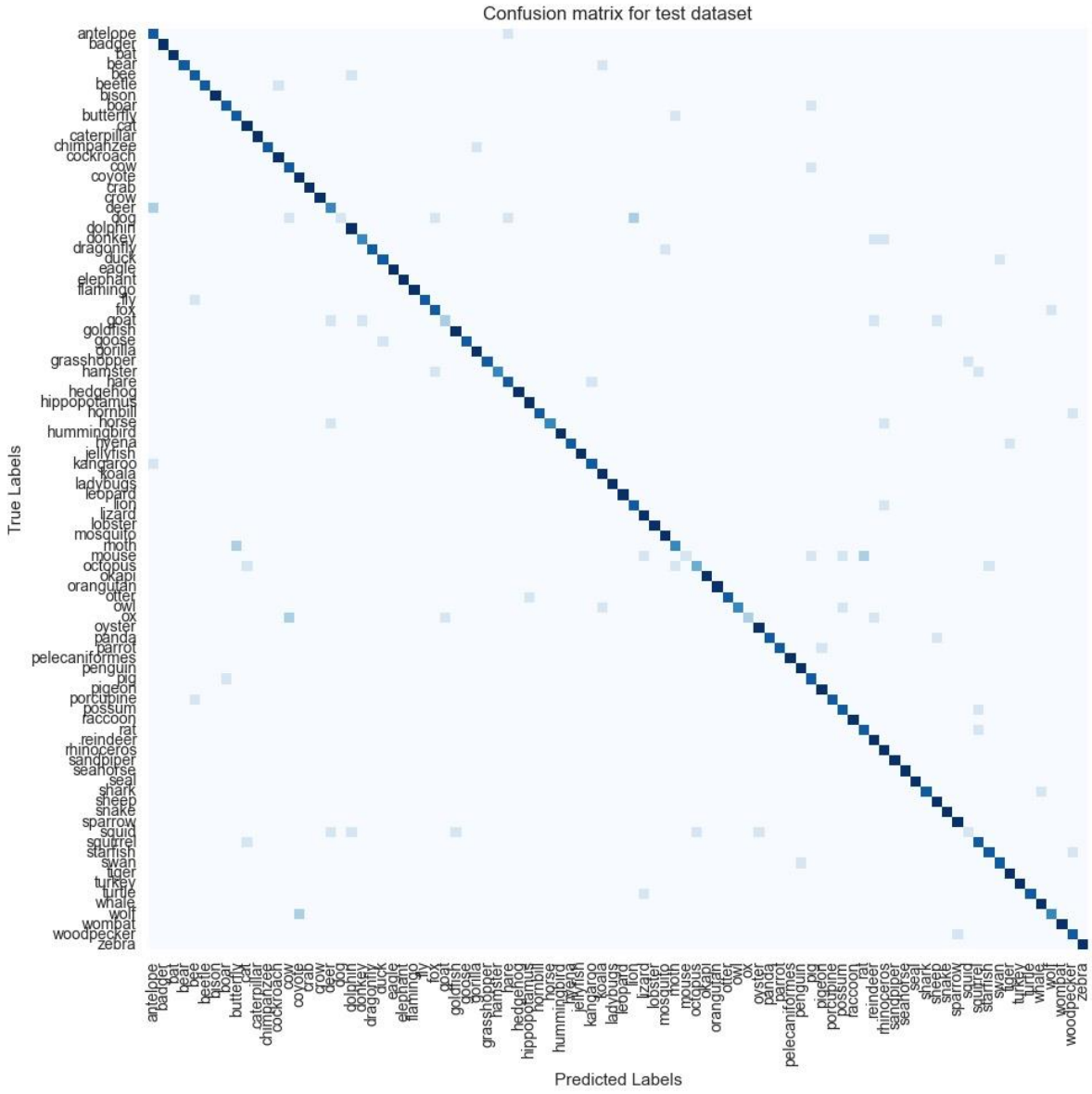
Takýto bol výsledok:

```
Uspesnost (accuracy) na trenovacich datach: 0.9374
Uspesnost (accuracy) na testovacich datach: 0.8667
```






Confusion matrix for train dataset
True Labels
Predicted Labels
antelope
badger
bat
bear
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
turtle
whale
wolf
wombat
woodpecker
zebra



Z výsledkov vidíme, že rozdiel v úspešnosti medzi trénovacími a testovacími dátami bol 0.07 v prospech trénovacích, čo je pri úspešnosti nad 0.93 v poriadku a nejde o pretrénovanie. Grafy priebehov však vyzerajú trochu inak ako by sme očakávali a v oboch vidíme, že počas trénovania vychádzali lepšie výsledky pre validačnú množinu ako trénovaciu. Postupne sa však rozdiel zmenšoval a pri viac epochách by sa to asi otočilo.

Vyskúšal som veľa rôznych architektúr skrytých vrstiev. Veľa z nich malo krajšie grafy priebehov, avšak výsledná úspešnosť medzi trénovacími a testovacími dátami bola vo väčšine prípadov príliš rozdielna, aj cez 0.1. To sa mi zdalo byť už príliš veľa a preto som sa pokúšal nájsť takú architektúru, ktorá by minimalizovala práve tento rozdiel v úspešnostiach. Výsledok, ktorý som uviedol predstavuje najmenší rozdiel, aký sa mi podarilo získať.

Vysoká úspešnosť na oboch množinách sa premietla aj do konfúznych matíc. Obe majú takmer dokonalú diagonálu.

Z konfúznej matice na trénovacích dátach vidíme, že model sa najviac mýlil v týchto prípadoch:

- Deer – najčastejšia predpoveď Antelope
    - Je tam zjavná vonkajšia podobnosť.
- Goose - najčastejšia predpoveď Duck
    - Je tam zjavná vonkajšia podobnosť.
- Moth - najčastejšia predpoveď Butterfly
    - Je tam zjavná vonkajšia podobnosť.
- Mouse – najčastejšie predpovede Hamster a Rat
    - Je tam zjavná vonkajšia podobnosť.
- Ox – najčastejšie predpovede Cow a Rhinoceros
    - Je tam zjavná vonkajšia podobnosť s Cow, s Rhinoceros ani nie a ťažko povedať, prečo to dáva predikcie. Môžu mať rovnakú farbu kože, prípadne byť v podobnom prostredí.

Z konfúznej matice na testovacích dátach vidíme, že model sa najviac mýlil v týchto prípadoch:

- Dog – najčastejšie predpovede Cow, Fox, Hare, Lion
    - Teoreticky môže medzi týmito zvieratami nastať vonkajšia podobnosť.
- Goat – najčastejšie predpovede Deer, Donkey, Reindeer, Sheep
    - Je tam zjavná vonkajšia podobnosť.
- Mouse – Lizard, Pig, Possum, Rat
    - S Possum a Rat je zjavná vonkajšia podobnosť, s Lizard a Pig ani nie. Nedá sa povedať, prečo to takto predpovedalo.
- Ox – najčastejšie predpovede Cow, Goat, Reindeer
    - Je tam zjavná vonkajšia podobnosť.
- Squid – najčastejšie predpovede Deer, Dolphin, Goldfish, Octopus, Oyster
    - Okrem Deer sa dá predpokladať, že tam je podobnosť kvôli vodnému prostrediu. S Deer sa to však nedá spojiť na základne ničoho.

Celkovo môžeme povedať, že trénovacie dáta vie model predikovať výborne a takmer vždy sa mýli len na zvieratách, ktoré sa na seba veľmi podobajú. Na trénovacích dátach je vidieť, že

v niektorých prípadoch model predpovedá aj také zviera, ktoré s pravým nemá skoro nič spoločné. Taktiež je tam viac predpovedí pre každé zviera. Pri testovacích to boli 1 až 2, tu 3 až 5.