

Klasifikace psů podle plemen

Bc. Vojtěch Moravec

LS 2020

1 Úvod

V tomto projektu do předmětu Metody Analýzy Dat 4 se budeme zabývat kategorizací neboli klasifikací obrazových dat. Přesněji se zaměříme na kategorizaci psů podle plemen. Toto téma jsme zvolili podle úkolu ze stránky Kaggle, který můžeme nalézt na adrese <https://www.kaggle.com/c/dog-breed-identification/overview/description>.

Cílem tohoto úkolu je pomocí trénovacích dat naučit klasifikátor, který bude následně schopen kategorizovat obrázky psů do 120 vybraných plemen. Jelikož se jedná o obrazová data a klasifikaci, tak jsme se rozhodli využít konvolučních sítí s mnoha vrstvami, které jsou v současnosti *state of the art* metodou pro klasifikaci a detekci objektů v obrazových datech. V této práci nejprve uvedeme informace o originálních datech a jak jsme je zpracovali. Následně se budeme zabývat samotnou klasifikací a výsledky.

2 Popis datasetu

Dataset, který je k dispozici k úkolu na stránce Kaggle (viz URL v předchozí kapitole), vychází z datasetu psů vytvořeného na Standfordské univerzitě. Konkrétně se jedná o *Stanford Dogs Dataset* [1]. Kde tento dataset byl vytvořen z ještě většího datasetu ImageNet [2].

Originální úkol na stránce Kaggle obsahuje jak trénovací, tak i testovací dataset, ale labely jsou k dispozici jen pro trénovací dataset. Label je pravdivá hodnota, která nám říká, do které kategorie obrázků spadá. Konkrétně jaké je plemeno psa na daném obrázku. Absence testovacích labelů, znamená nemožnost vyhodnotit naše modely. Proto jsme se rozhodli využít originální *Stanford Dogs Dataset*, ve kterém nalezneme pro všechny obrázky labely. Četnosti jednotlivých psích plemen v trénovacím datasetu jsou pro všechny plemena rovny hodnotě 100. Pro jedno plemeno je tedy vždy k dispozici 100 trénovacích obrázků. Velikost celého trénovacího datasetu je 12000 obrázků. Četnosti v testovacím datasetu se již liší a najdeme je na Obrázku 1. Počet 100 obrázků na jedno plemeno je dosti malý, vzhledem k tomu, že chceme kategorizovat celkem 120 plemen. V této práci vyzkoušíme několik strategií, jak se s tímto vypořádat. Nejprve vybereme pouze 20 plemen, které jsou nejčastější v testovacím datasetu. Poté také vyzkoušíme umělé zvětšení datasetu pomocí augmentace dat a hlavně se zaměříme na *transfer learning*.

2.1 Příprava datasetu

V *Stanford Dogs Dataset* jsou pro nás důležité soubory `images.tar` a `lists.tar`. První archiv obsahuje po rozbalení obrázky psů, rozdělených do podsložek podle plemen. Trénovací i testovací data jednoho plemene jsou tedy uloženy v jedné složce. Druhý archiv obsahuje binární soubory v MATLAB formátu. Konkrétně se jedná o `train_list.mat`, resp `test_list.mat`, tyto soubory obsahují cesty k trénovacím, resp. testovacím obrázkům a zároveň jejich labely. Již upravené soubory budou k dispozici v přílohách této práce, jedná se o:

- `classes.csv` - Seznam všech tříd a příslušících číselných labelů
- `train.csv` - Seznam obrázků, které jsou využity k trénování, spolu s labely
- `test.csv` - Seznam obrázků, které jsou využity k testování, spolu s labely

Co se týče předzpracování dat, před samotným učením, tak my jsme provedli pouze sjednocení velikosti obrázků a normalizaci. Obrázky, ať už v trénovací nebo testovací množině, měli různé rozměry. Konvoluční síť má na vstupu definován rozměr vstupních dat, proto museli být rozměry obrázků sjednoceny. Rozhodli jsme se použít rozměr $300 \times 300 \times 3$, neboli obraz 300 pixelů vysoký a široký se třemi barevnými kanály. Obrázky jsme nepřeváděli do stupňů šedi, neboť si myslíme, že barevná složka je poměrně důležitým faktorem, při klasifikaci psího plemena.

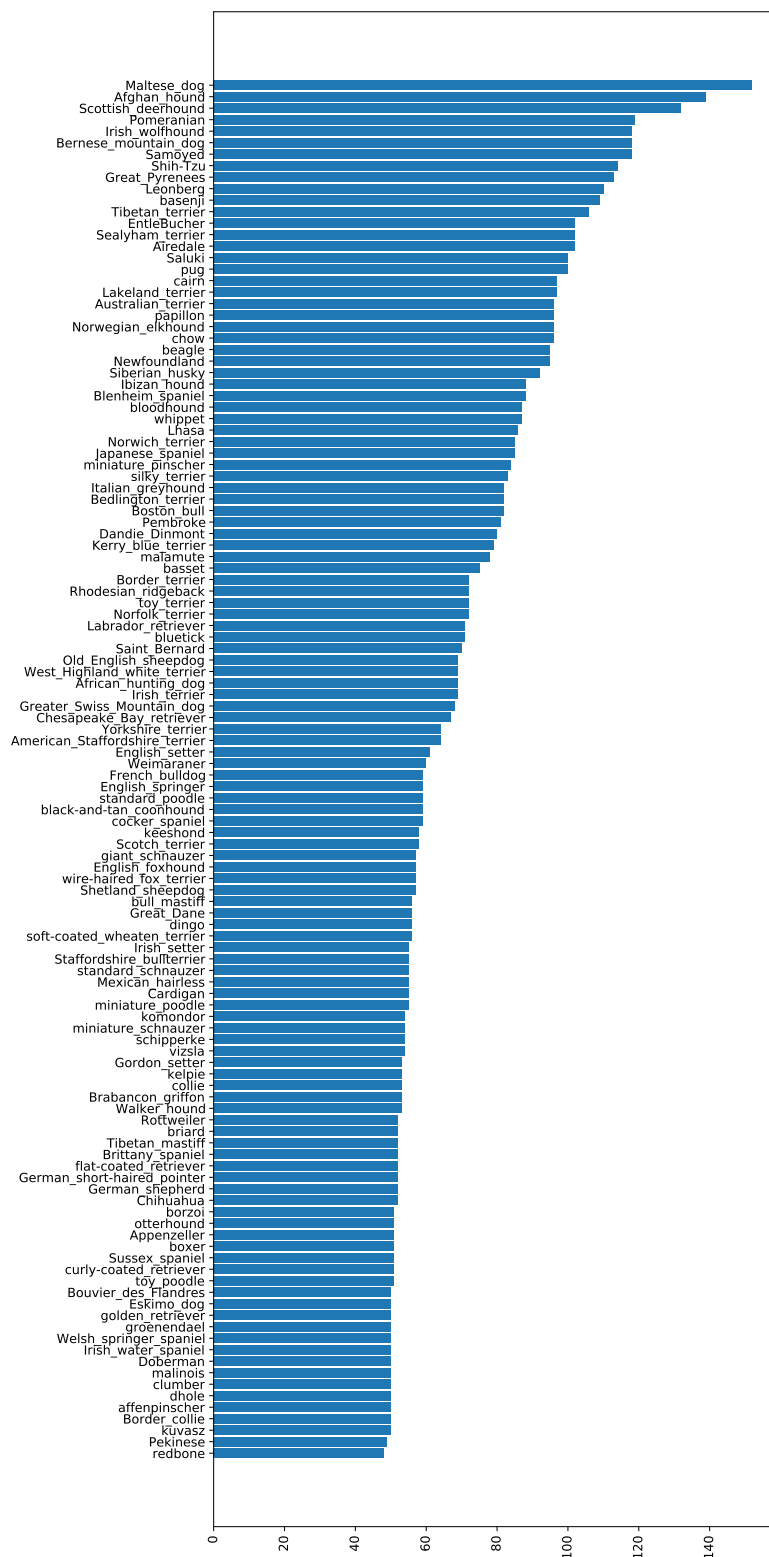
Normalizace byla provedena z rozsahu pixelů $[0; 255]$ na $[0; 1]$. Celé načtení datasetu, spolu se změnou velikosti a normalizací bylo provedeno následující funkcí, za pomoci dvou funkcí `load_img` a `img_to_array` z Python knihovny Keras [3].

```

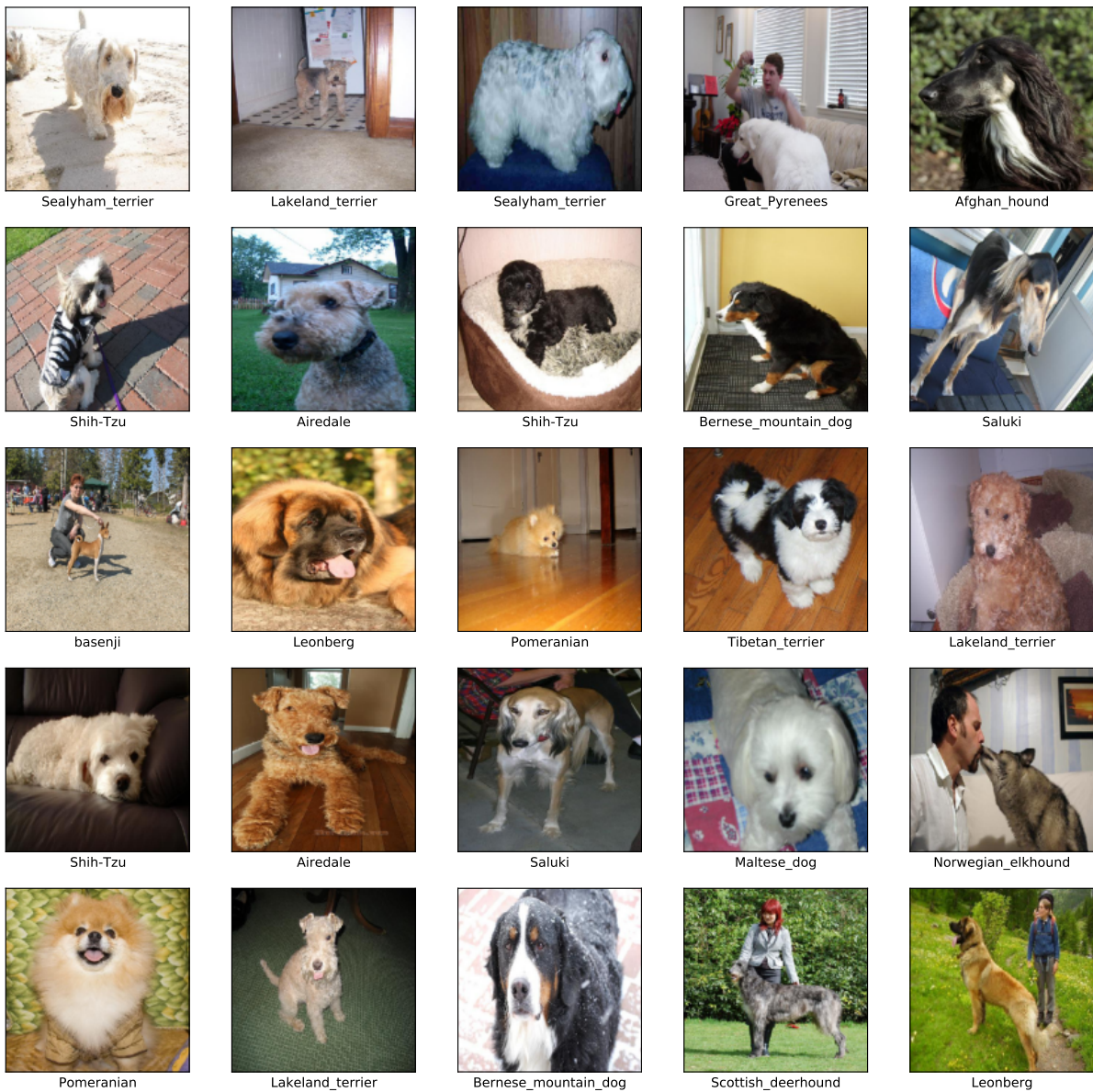
1 IMG_DIM = 300
2 def load_image_dataset(folder, paths):
3     dataset = []
4     count = len(paths)
5     for i in range(count):
6         imagePath = os.path.join(folder, paths[i])
7         image = load_img(imagePath, target_size=(IMG_DIM, IMG_DIM))
8         dataset.append(img_to_array(image, dtype=np.float32) / 255)
9     return np.asarray(dataset)

```

Ukázku obrázků z datasetu můžeme vidět na Obrázku 2. Pod každým obrázkem nalezneme název třídy, do které spadá, tedy psí plemeno.



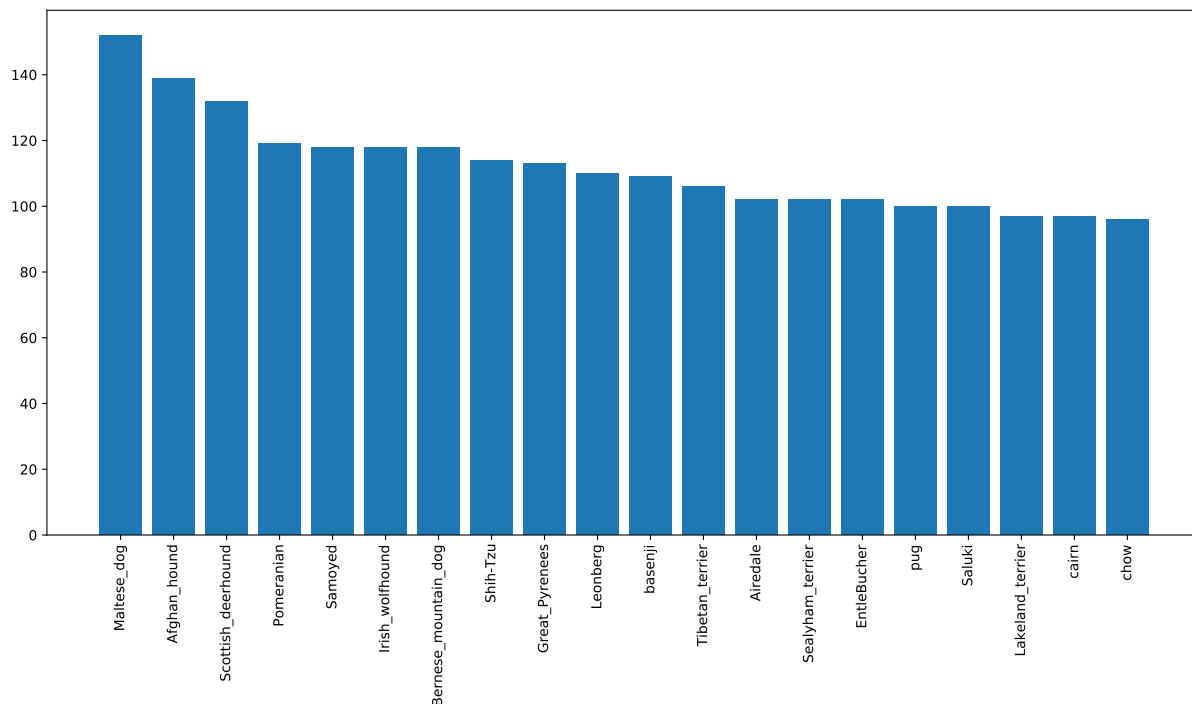
Obrázek 1: Četnost jednotlivých plemen v testovacím datasetu



Obrázek 2: Ukázka z datasetu

3 Klasifikace 20 plemen

V této kapitole se zaměříme na klasifikaci 20 nejčastějších plemen v testovacím datasetu, jejich četnosti můžeme vidět na Obrázku 3.



Obrázek 3: Četnost 20 nejčastějších plemen v testovacím datasetu

V trénovacím datasetu se stále nachází 100 obrázků na jedno plemeno, a proto je velikost trénovací sady 2000 obrázků. Z důvodu malé velikosti, jsme vzali pouze 10% této sady na validační dataset. První model, který jsme vyzkoušeli a taky největší, který nám dovolili hardwarové limitace paměti, je tento:

```
1 model = keras.Sequential(layers=[
2     Input(shape=(IMG_DIM, IMG_DIM, 3), dtype=np.float32), # 300 x 300
3     Conv2D(64, kernel_size=(5,5), activation='relu'),
4     Conv2D(64, kernel_size=(5,5), activation='relu'),
5     Dropout(rate=0.15),
6     Conv2D(64, kernel_size=(5,5), activation='relu'),
7     MaxPooling2D(pool_size=(2,2)), # 150 x 150
8     Conv2D(64, kernel_size=(5,5), activation='relu'),
9     Conv2D(64, kernel_size=(5,5), activation='relu'),
10    MaxPooling2D(pool_size=(2,2)), # 75 x 75
11    Conv2D(128, kernel_size=(5,5), activation='relu'),
12    Dropout(rate=0.15),
13    Conv2D(128, kernel_size=(5,5), activation='relu'),
14    Conv2D(128, kernel_size=(5,5), activation='relu'),
15    MaxPooling2D(pool_size=(2,2)), # 37 x 37
```

```

16 Conv2D(256, kernel_size=(3,3), activation='relu'),
17 Conv2D(256, kernel_size=(3,3), activation='relu'),
18 MaxPooling2D(pool_size=(2,2)), # 18 x 18
19 Conv2D(256, kernel_size=(3,3), activation='relu'),
20 Conv2D(256, kernel_size=(3,3), activation='relu'),
21 MaxPooling2D(pool_size=(2,2)), # 9 x 9
22 GlobalMaxPooling2D(),
23 Dense(units = CLASS_COUNT, activation='softmax')
24 ])

```

V tomto modelu můžeme vidět bloky konvolučních vrstev, následovány MaxPooling2D vrstvou. MaxPooling2D vždy snižuje rozměry výstupu vrstvy dvojnásobně. Tyto rozměry jsou uvedeny v komentářích na pravé straně. Jako poslední je plně propojená vrstva, kde počet neuronů je roven počtu klasifikovaných tříd, v tomto případě tedy 20. Důležitá je aktivační funkce **softmax** v poslední vrstvě, která vlastně generuje pravděpodobnosti jednotlivých tříd. Co se týče ztrátové funkce, tak tento model a spolu s ním všechny modely, které budou uvedeny v této práci, používá *sparse categorical crossentropy*. Tato funkce je zvolena, neboť jednotlivé třídy jsou exkluzivní a pes by měl být pouze jednoho plemene. *Nechceme-li detekovat křížence, což není cílem této práce*. Stejně tak v celé práci je zvolen optimizátor *Adam*, úspěšnost měříme pomocí přesnosti a v základu trénujeme ve 20 epochách.

Výsledky trénování první konvoluční neuronové sítě nalezneme v Tabulce 1. Od 3. epochy se trénování zasekne na validační přesnosti 0,0250 a síť již není schopná se naučit nic nového. Takhle to pokračuje až do 20. epochy. Víme, že toto není způsobeno špatným formátem nebo snad zpracováním dat, neboť si následně ukážeme, jakých výsledků dosáhnout pomocí techniky *transfer learning*. Příkládáme se tedy k názoru, že nemáme dostatečný počet trénovacích dat.

Jako záchrana se nabízí umělá augmentace dat, například pomocí `ImageDataGenerator` z knihovny Keras. S tímto generátorem jsme vyzkoušeli dvě strategie. V první strategii jsme generovali obrázky, které mohli být zrcadlově přetočeny podle obou os, zároveň mohlo dojít k rotaci až o 20 stupňů a náhodnému přiblížení či oddálení o 15 %. V druhé strategii jsme navíc přidali horizontální a vertikální posuny až o 15 %. Bohužel, ani jedna strategie augmentace dat nevedla ke zlepšení a model stále nebyl schopen klasifikovat psi. Následně jsme vyzkoušeli vynechat **Dropout** vrstvy, či zjednodušit architekturu sítě, ale nic nevedlo ke zlepšení.

3.1 Využití *transfer learningu*

Transfer learning (volně přeloženo jako *přenesené učení*) je metoda strojového učení, která se snaží využít či přenést již naučené znalosti v rámci jednoho problému, k řešení jiného problému, který mu je podobný. V našem případě, víme že obrázky psů pochází z datasetu ImageNet [2] a že Keras nabízí modely přímo naučené na tomto datasetu. Můžeme tedy využít znalostí, které se síť naučila na celém tomto datasetu a využít je pouze ke klasifikaci psích plemen.

Epocha	Tr. ztráta	Tr. přesnost	Val. ztráta	Val. přesnost
1	3,0042	0,0428	2,9966	0,0400
2	2,9961	0,0511	2,9965	0,0400
3	2,9960	0,0483	2,9972	0,0250
4	2,9959	0,0528	2,9975	0,0250
5	2,9959	0,0528	2,9978	0,0250

Tabulka 1: Trénování vlastního modelu

Tato technika je velice výhodná v případech, kdy nemáme dostatečný počet trénovacích dat k naučení celé sítě, tak jak je tomu v našem případě. Zároveň je délka trénování mnohem kratší, neboť váhy většiny vrstev jsou zmrazeny a učíme pouze poslední vrstvu či vrstvy, které nahradíme v přeneseném modelu. Modely, které jsme se rozhodli vyzkoušet nalezneme v Tabulce 2.

Model	Počet parametrů	Počet vrstev
VGG19 [4]	143 667 240	26
Xception [5]	22 910 480	126
InceptionV3 [6]	23 851 784	159
InceptionResNetV2 [7]	5 873 736	572

Tabulka 2: Převzaté modely konvolučních neuronových sítí

U těchto přenesených modelů jsme tedy využili váhy, které se naučili na ImageNetu a nastavili dimenzi vstupních dat na již zmíněných $300 \times 300 \times 3$ pixelů. U všech modelů jsme nevyužili originální poslední plně propojené vrstvy a nastavili jsme GlobalMaxPooling2D jako poslední vrstvu převzatého modelu. Na tuto vrstvu jsme následně napojili naši plně propojenou vrstvu s počtem neuronů rovným počtu tříd. Výsledky trénování a testování pro zkombinované modely nalezneme v Tabulce 3.

Model	Trénovací přesnost	Testovací přesnost
VGG19	0,7289	0,5040
Xception	1,0000	0,9626
InceptionV3	1,0000	0,9599
InceptionResNetV2	0,9994	0,9603

Tabulka 3: Výsledky klasifikace pro 20 plemen s použitím techniky *transfer learning*

V této tabulce si všimneme, že oba modely Xception a InceptionV3 dosáhli perfektní trénovací přesnosti, což znamená, že se naučili přesně na trénovací data dochází k *overfittingu*. Toto samo o sobě nemusí být dobré, neboť se může stát, že síť nebude schopna

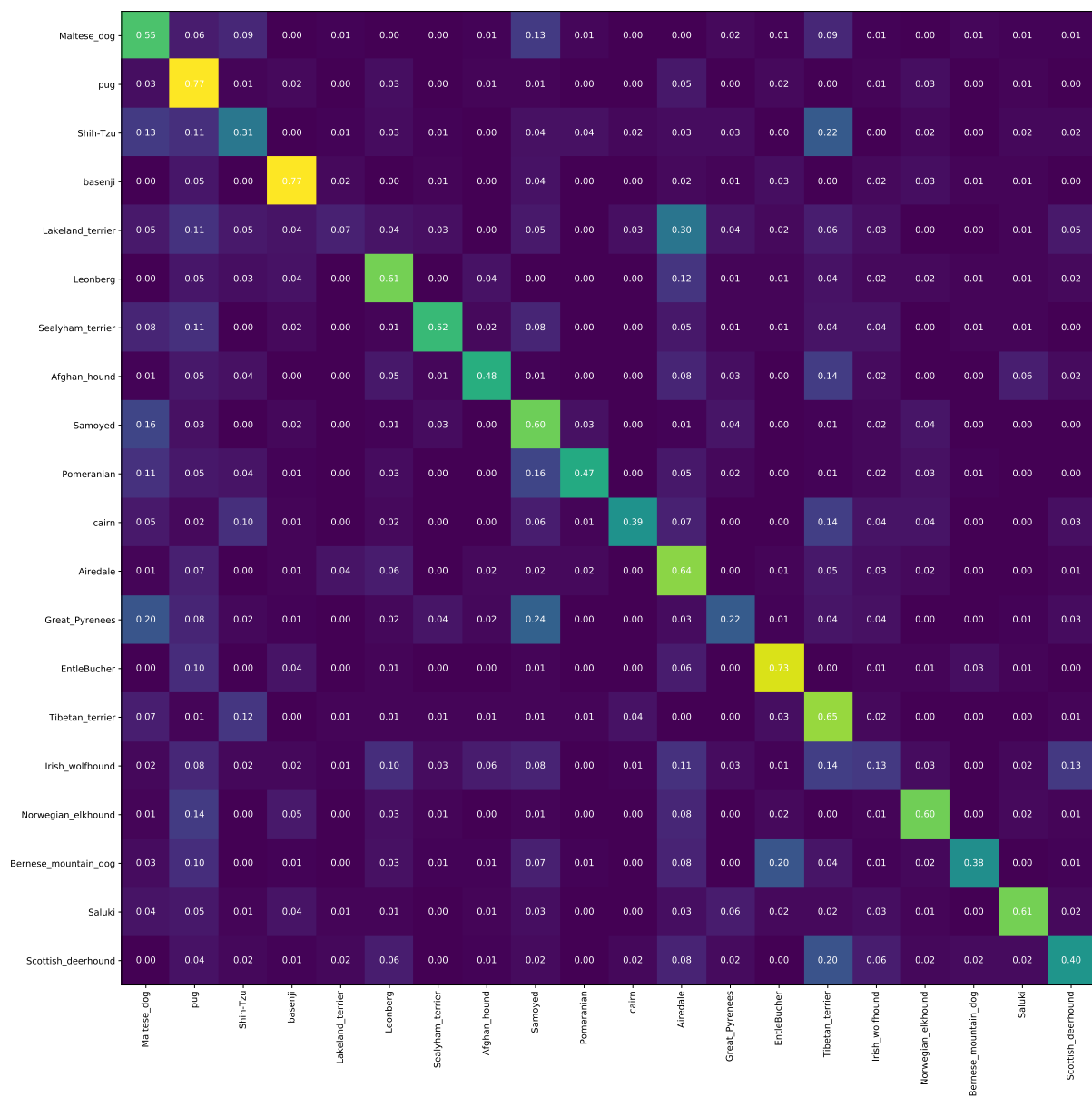
klasifikovat nic jiného než trénovací data. Avšak v našem případě vidíme, že přesnost je na testovací sadě velmi vysoká, kolem 96 %. Nejhuře dopadl model VGG19, který je také nejjednodušší. Jeho výsledky jsme se snažili vylepšit augmentací dat, ale ani zde augmentace nepomohla. Při použití první strategie augmentace došlo k malému zlepšení přesnosti na 0,5150, tohoto výsledku jsme dosáhli až po 40 trénovacích epochách.

Dále jsme vyzkoušeli více plně propojených vrstev za poslední MaxPooling2D vrstvou a před finální detekční vrstvou, výsledky jsou uvedeny Tabulce 4. Tyto přidané **Dense** vrstvy měli každá 256 neuronů a aktivační funkci ReLU. S těmito vrstvami můžeme pozorovat zlepšení trénovací přesnosti z 0,7289 až na 0,8900 při použití 3 vrstev navíc. Oproti tomu testovací přesnost dosáhla nejlepší hodnoty 0,5241 při dvou přidaných vrstvách.

Počet Dense vrstev	Trénovací přesnost	Testovací přesnost
1	0,7933	0,5209
2	0,8700	0,5241
3	0,8900	0,5076

Tabulka 4: VGG19 - Vliv plně propojených vrstev na výsledek klasifikace

Podíváme-li se na matici záměn pro 20 nejčtenějších plemen v testovacím datasetu, pro model VGG19 [4] na Obrázku 4, tak si všimneme, že i když tento model dosahoval nejmenší přesnosti tak je velmi dobrý. Drtivá většina psích plemen je určena správně, respektive správná třída dosahuje největší pravděpodobnosti. Zaměříme-li se na plemeno Lakeland_terrier, tak to je nejvíce zaměňované za Airedale, proč je tomu tak, nám bude jasné při pohledu na obě plemena na Obrázku 5. Podobně je plemeno Great_Pyrenees zaměňováno za plemeno Samoyed. U těchto dvou plemen, byla určena jiná třída s velkou přesností, jinak je tomu u Irish_wolfhound. Psi tohoto plemene se modelu nepodařilo přiřadit žádné třídy, pravděpodobnost byla rozdělena mezi všechny třídy s podobnou hodnotou.



Obrázek 4: Matice záměn pro 20 vybraných plemen, model VGG19



(a) Lakeland_terrier



(b) Airedale



(c) Great_Pyrenees



(d) Samoyed

Obrázek 5: Zaměňované psí plemena

4 Klasifikace všech psích plemen

V této sekci bychom se chtěli zabývat klasifikací všech 120 psích plemen, budou tedy využity všechny data z datasetu. Jak jsme již zjistili v kapitole předchozí, počet dat není dostatečný k natrénování zcela nové konvoluční neuronové sítě. Z tohoto důvodu hned využijeme techniku *transfer learning*. Co se týče modelů, tak znova využijeme ty, uvedené v Tabulce 2.

Podobně jako u modelu VGG19 v předchozí kapitole, jsme i zde vyzkoušeli větší počet plně propojených vrstev, zde pro model InceptionV3. Výsledky zkoumání úprav modelu InceptionV3 nalezneme v Tabulce 5. V této tabulce jsme zkoušeli několik úprav i augmentaci dat. Všimneme si, že jsme nebyli schopni zlepšit přesnost klasifikace, ale i přesto testovací ztráta klesla oproti základnímu modelu.

Název	Testovací ztráta	Testovací přesnost
Základní model	1,2912	0,8501
Dense512	0,7459	0,8367
$2 \times$ Dense512	0,7989	0,8338
$2 \times$ Dense512 + Dense256	0,7278	0,8325
$2 \times$ Dense512 a augmentace	0,6262	0,8169

Tabulka 5: InceptionV3 - úpravy architektury

Podíváme-li se přímo na výsledky převzatých modelů v Tabulce 6, zjistíme že největší přesnosti dosahuje model InceptionResNetV2. Toto jsou pořád výsledky pro obrázky rozměrů 300×300 . Model VGG19 oproti třem ostatním velmi zaostává. Dále se podíváme podrobněji na výsledky nejlepšího modelu. Ten dokázal z 8580 testovacích obrázků správně klasifikovat 7459 obrázků a 1121 tedy bylo špatně zařazeno.

Model	Testovací přesnost
VGG19	0,2787
Xception	0,8469
InceptionV3	0,8501
InceptionResNetV2	0,8693

Tabulka 6: Výsledky klasifikace pro všechna plemena s použitím techniky *transfer learning*

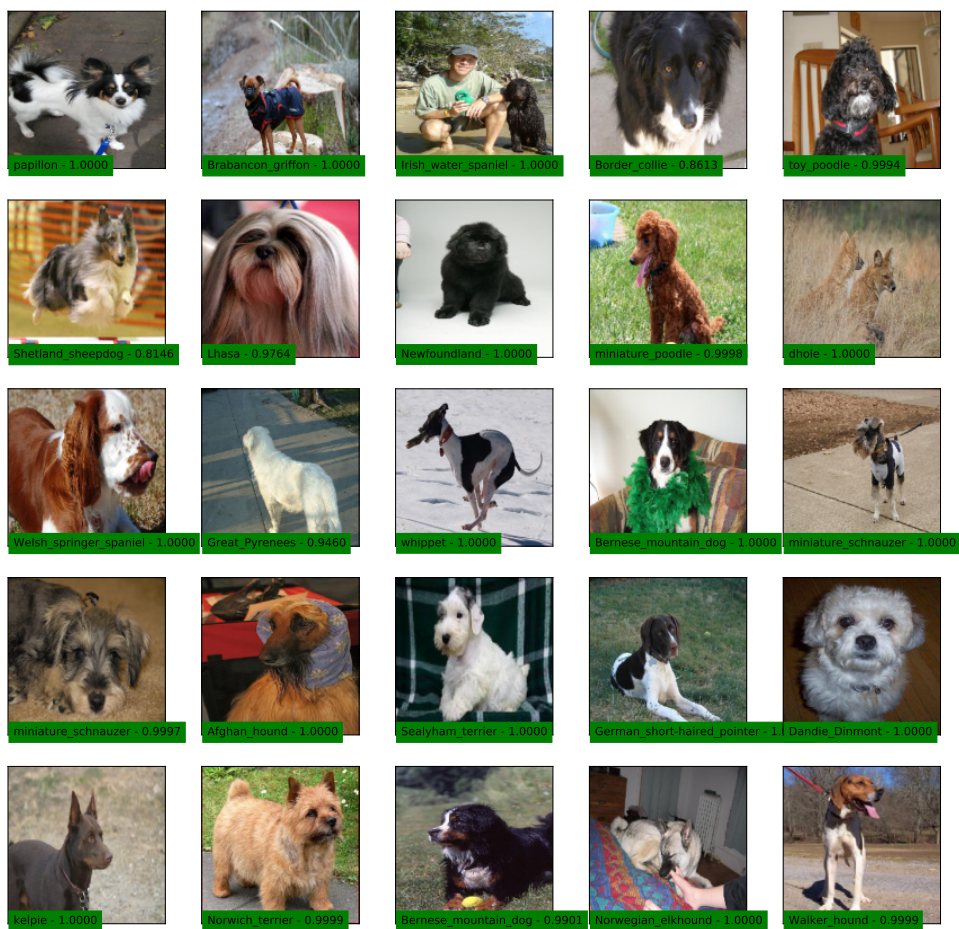
Celá matice záměn bude k dispozici jako příloha, do tohoto textu se nehodí kvůli své velikosti. Co můžeme ale říci, tak v celé matici se nachází pouze čtyři řádky, kde nalezneme maximální pravděpodobnost mimo hlavní diagonálu. Detaily těchto řádků nalezneme v Tabulce 7. U těchto čtyř plemen můžeme očekávat, že ve většině případů dojde ke špatné klasifikaci. Již podle názvu nám dochází, že tyto plemena, správná a zaměňovaná jsou si vizuálně velice podobné.

Správná třída	Pravd.	Zaměňovaná třída	Pravd.
Staffordshire_bullterrier	0,2182	American_Staffordshire_terrier	0,6909
Border_collie	0,3200	collie	0,6000
Eskimo_dog	0,3600	Siberian_husky	0,4800
toy_poodle	0,3333	miniature_poodle	0,6666

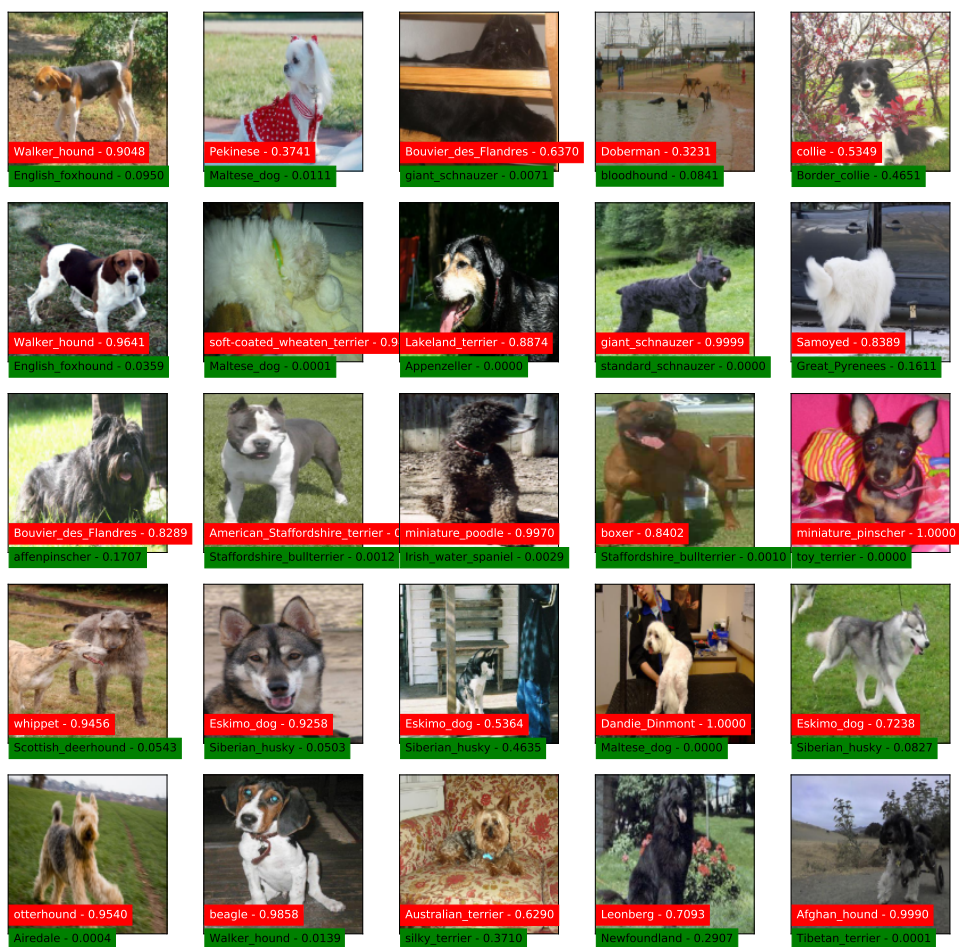
Tabulka 7: Hlavní záměny při klasifikaci všech plemen

Ukázky klasifikací nalezneme na následujících obrázcích. V zeleném obdélníku je vždy uvedena správná třída a v červeném obdélníku je uvedena špatně určená třída. Číslo napravo od názvu plemene značí pravděpodobnost, s jakou byla třída určena. Obrázky správné klasifikace nalezneme na Obrázku 6. Zde vidíme, že téměř všechny plemena jsou určena s pravděpodobností 1,0000 nebo alespoň hodně vysokou hodnotou. Ukázka chybných klasifikací je uvedena na Obrázku 7. Zde už není třída určena s tak velkou pravděpodobností, ale často se stává, že správná třída má dosti malou pravděpodobnost. Toto znamená, že se model nenaučil tyto plemena dobře detekovat. Dalším vysvětlením je málo různých trénovacích dat.

Abychom ještě vylepšili výsledky nejlepšího modelu InceptionResNetV2 vyzkoušeli jsme obrázky zvětšit na 500×500 pixelů. Díky této úpravě klesla testovací ztráta z 0,9943 na 0,5830 a přesnost vzrostla na 0,8956 z původních 0,8693. Zvětšením rozlišení jsme tedy dosáhli zlepšení v rámci cca 2,5 procenta. Nevýhodou tohoto přístupu je větší paměťová a časová náročnost.



Obrázek 6: Ukázka správné klasifikace



Obrázek 7: Ukázka špatné klasifikace

5 Závěr

V této práci jsme se zaměřili na klasifikaci psů podle plemen. Z originálního datasetu, který byl rozdělen na trénovací a testovací část jsme natrénovali několik modelů konvolučních neuronových sítí. Snaha o vytvoření a natrénování vlastního modelu byla marná, z důvodu, který bude uveden dále. Hojně jsme v tomto projektu využili techniky *transfer learning*, která nám dovoluje použít a upravit již předem naučené modely. Hlavním problémem originálního datasetu je nedostatek trénovacích dat. Na každé plemeno je k dispozici pouze 100 obrázků a plemen je celkem 120. Tento problém jsme se snažili vyřešit augmentací dat, která avšak nevedla k lepším výsledkům. Co se týče převzatých modelů, tak těch jsme vyzkoušeli čtyři, od jednoduššího VGG19 až po složitý InceptionResNetV2, který kombinuje techniky ze dvou různých architektur. Konvoluční neuronové sítě jsme vyzkoušeli nejdříve na 20 vybraných plemenech, které byly nejvíce zastoupeny v testovacím datasetu. Zde jsme dosáhli velmi vysokých přesností kolem 96 %. Dále jsme se již vrhli na celou datovou sadu a všech 120 plemen. Zde dosáhli tři převzaté modely přesnosti minimálně 84 % a nejlepší InceptionResNetV2 dosáhl 86,93 %. Tento výsledek jsme dokázali ještě o něco málo zlepšit pomocí zvětšení obrázků.

Reference

- [1] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, “Novel dataset for fine-grained image categorization,” in *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, CO), June 2011.
- [2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [3] F. Chollet *et al.*, “Keras.” <https://keras.io>, 2015.
- [4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
- [5] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2016.
- [6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 2015.
- [7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” 2016.