

Využití konvoluční neuronové sítě pro binární klasifikaci obrázků: Případová studie klasifikace aut a motorek

Vladimír Vinkler

20. května 2023

Abstrakt

Tato studie prozkoumává konvoluční neuronové sítě a jejich aplikaci na binární klasifikaci 4000 obrázků motorek a aut. Studie popisuje předzpracování obrázků, modelování konvoluční neuronové sítě a evaluaci vytvořeného modelu. Studie může být užitečná pro obor počítačové vize, a to například pro vytváření autonomních vozidel.

1 Úvod

Klasifikace obrázků je oborem spadajícím do počítačového vidění. Je to obor který se rychle vyvíjí a používají se v něm komplexní modely hlubokého učení. Vytváření a aplikace těchto modelů pokrývá širokou škálu oblastí, od "lékařského zobrazování" (medical imaging), přes vytváření autonomních zařízení až po "sledovací systémy" (surveillance systems).

V této práci se budeme zabývat klasifikací obrázků motorových vozidel, a to do dvou skupin: motorek a aut. Pro tento účel byl zvolen vyvážený dataset z Kaggle [Sax22] obsahující obrázky 2000 motorek a 2000 aut. Prvně se budeme zabývat standardizací obrázků do jednotného formátu, následně budeme vytvářet konvoluční neuronovou síť, která se obrázky bude snažit klasifikovat do již zmíněných skupin.

Jako inspiraci je použit model klasifikující obrázky koček a psů od deeplizard [dee23], tento model byl založen na modelu VGG16 [SZ15].

2 Metody

2.1 Předzpracování obrázků

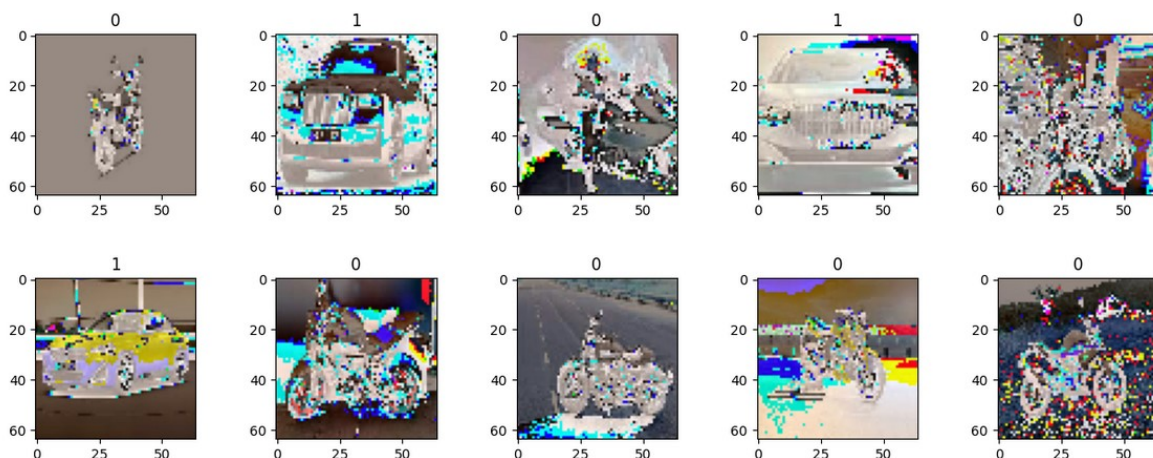
Předzpracování obrázků je důležitá část pro klasifikaci obrázků, obzvláště při využití metody jako konvoluční neuronové sítě. Tyto sítě pracují nejlépe s normalizovanými a uniformními vstupními daty, bez úpravy obrázků, aby splňovaly tyto požadavky může dojít k tomu, že náš výsledný model bude zaujatý (biased), bude komputačně neefektivní, či dokonce nebude daný model možno ani vytvořit, protože obrázky v daném formátu nepřijme. Dále je důležité, abychom původní dataset rozdělili na více částí, a to trénovací, validační a testovací. Model se bude trénovat na části trénovací. Při trénování nám k předběžné evaluaci poslouží část validační, na které se model kontroluje při tréningu. Tyto validační data však nevstupují do tréningu jako takového. Při finální evaluaci modelů nám pak poslouží část testovací.

Prvním krokem při předzpracování dat je rozdělení na již zmíněné tři části. Toto je provedeno pomocí balíčku os. Z původního zazipovaného datasetu se extrahují všechny obrázky, a následně jsou pro ně vytvořeny nové adresáře. Rozdělení je následující:

- Trénovací část: 80%
- Validací část: 10%
- Testovací část: 10%

Dalším krokem je převedení obrázků do jednotného formátu. Z důvodů relativně velkého a komplexního datasetu je zvolen formát 64x64 pixelů. Dále je důležité normalizovat barvy k jednotlivým obrázkům. K tomu je použito předzpracování na základě modelu VGG16 [SZ15]. Nejprve se vypočítá průměrná hodnota barev pixelů pro jednotlivé barevné složky ve všech obrázcích celého datasetu. Poté se jednotlivé pixely všech obrázků převedou do normalizované formy v rozmezí -1;1. V tomto rozsahu představuje hodnota 0 vypočítaný průměr barev pixelů pro jednotlivé barvy.

Finálně se obrázky uloží do jednotlivých částí po "várkách" (batches) o velikosti 10. To je provedeno pro zvýšení efektivity trénování modelu.



Obrázek 1: Příklady obrázků po předzpracování

2.2 Modelování konvoluční neuronové sítě

Modelování Konvoluční neuronové sítě probíhá v programovacím jazyku Python s využitím balíčků Keras a TensorFlow. Tyto balíčky poskytují nástroje pro vytváření a trénování neuronových sítí. Pomocí balíčku keras se vytváří struktura neuronové sítě jako takové, a následně balíček TensorFlow umožňuje efektivní nástroje pro výpočet výsledků sítě. Pomocí těchto nástrojů je vytvořen model s cílem rozpoznávat obrázky aut a motorek.

Konvoluční neuronové sítě jsou komplexní třída modelů hlubokého učení, která je obzvláště efektivní při rozpoznávání a klasifikaci obrázků díky jejich schopnosti zapamatování si lokální prostorové korelace v obrázcích.

2.2.1 Model 1

První vytvořený model byl trénován na pěti epochách a skládá se ze sedmi následujících vrstev:

1. Vstupní vrstva, která je nastavená aby přijala obrázek o rozlišení 64x64 ve třech základních barvách
2. První konvoluční vrstva: Tato vrstva je tvořena 32 filtry, neboli maticemi o velikosti 3x3. Dále je zde ReLU aktivační funkce, která negativní výsledek výpočtu filtru přepíše na 0. Dále je zde nastaveno, že výsledek této vrstvy musí zachovat rozměry vstupu. Po tomto bloku je MaxPooling2D, který nadále zmenší rozměry výsledku.
3. Druhá konvoluční vrstva: Má stejnou strukturu jako vrstva předchozí. Jediná změna je v počtu filtrů, a to 64.
4. Třetí konvoluční vrstva: Má stejnou strukturu jako předchozí. Jediná změna je v počtu filtrů, a to 128.
5. Vrstva zploštění: Zde se převádí 2D obrazovou mapu na 1D vektor.

6. Spojená vrstva: V této části je 128 neuronů a ReLU aktivační funkce. Dále se zde využívá L2 regularizace ve snaze předejít přeučení modelu na trénovacích datech.
7. Výstupní vrstva: Konečná vrstva modelu, ve které jsou 2 neurony s softmax aktivační funkcí, která má jako výstup vektor, který reprezentuje pravděpodobnosti výsledků pro jednotlivé obrázky.

Model: "sequential_14"

Layer (type)	Output Shape	Param #
conv2d_42 (Conv2D)	(None, 64, 64, 32)	896
max_pooling2d_42 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_43 (Conv2D)	(None, 32, 32, 64)	18496
max_pooling2d_43 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_44 (Conv2D)	(None, 16, 16, 128)	73856
max_pooling2d_44 (MaxPooling2D)	(None, 8, 8, 128)	0
flatten_11 (Flatten)	(None, 8192)	0
dense_15 (Dense)	(None, 128)	1048704
dense_16 (Dense)	(None, 2)	258

Total params: 1,142,210
 Trainable params: 1,142,210
 Non-trainable params: 0

Obrázek 2: Popis modelu 1 použitím metody summary

2.2.2 Model 2

Druhý vytvořený model byl trénován v osmi epochách a skládá se ze sedmi následujících vrstev:

1. Vstupní vrstva, která je nastavená aby přijala obrázek o rozlišení 64x64 ve třech základních barvách.
2. První konvoluční vrstva: Tato vrstva je tvořena 32 filtry, neboli maticemi o velikosti 3x3. Dále je zde LeakyReLU aktivační funkce. Tato aktivační funkce je podobná standardnímu ReLU, má však nevrací přímou 0, ale číslo blízko 0. Dále je opět zachováno, že výsledek této vrstvy musí zachovat rozměry vstupu. Po tomto bloku je MaxPooling2D, který nadále zmenší rozměry výsledku. Nakonec je přidána i vrstva Dropout, což je technika regularizace, která náhodně nastaví určitou část vstupních jednotek na 0 při trénování.
3. Druhá konvoluční vrstva: Má stejnou strukturu jako vrstva předchozí. Jediná změna je v počtu filtrů, a to 64.
4. Třetí konvoluční vrstva: Má stejnou strukturu jako předchozí. Jediná změna je v počtu filtrů, a to 128.
5. Vrstva zploštění: Zde se převádí 2D obrazovou mapu na 1D vektor.
6. Spojená vrstva: V této části je 128 neuronů a LeakyReLU aktivační funkce. Místo regularizace L2 je zde opět přítomna další vrstva Dropout.
7. Výstupní vrstva: Konečná vrstva modelu, ve které jsou 2 neurony s softmax aktivační funkcí, která má jako výstup vektor, který reprezentuje pravděpodobnosti výsledků pro jednotlivé obrázky.

Model: "sequential_16"

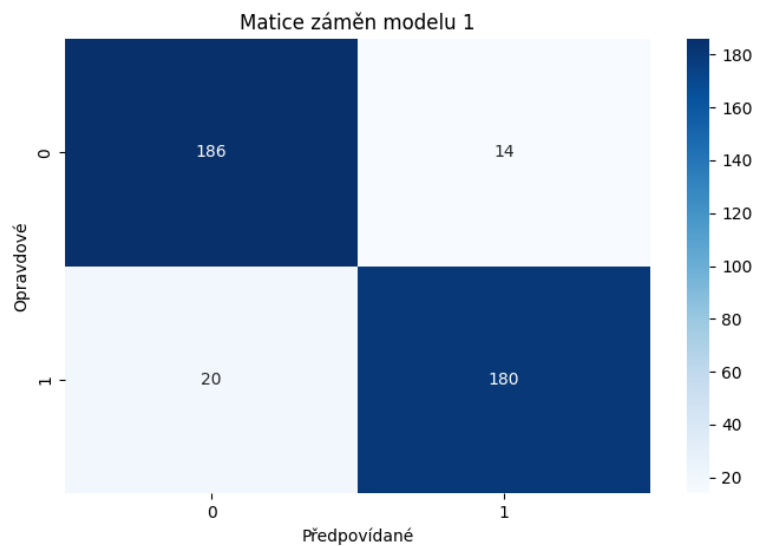
Layer (type)	Output Shape	Param #
conv2d_46 (Conv2D)	(None, 64, 64, 32)	896
leaky_re_lu (LeakyReLU)	(None, 64, 64, 32)	0
max_pooling2d_45 (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_47 (Conv2D)	(None, 32, 32, 64)	18496
leaky_re_lu_1 (LeakyReLU)	(None, 32, 32, 64)	0
max_pooling2d_46 (MaxPooling2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
conv2d_48 (Conv2D)	(None, 16, 16, 128)	73856
leaky_re_lu_2 (LeakyReLU)	(None, 16, 16, 128)	0
max_pooling2d_47 (MaxPooling2D)	(None, 8, 8, 128)	0
dropout_2 (Dropout)	(None, 8, 8, 128)	0
flatten_12 (Flatten)	(None, 8192)	0
dense_17 (Dense)	(None, 128)	1048704
leaky_re_lu_3 (LeakyReLU)	(None, 128)	0
dropout_3 (Dropout)	(None, 128)	0
dense_18 (Dense)	(None, 2)	258
Total params: 1,142,210		
Trainable params: 1,142,210		
Non-trainable params: 0		

Obrázek 3: Popis modelu 2 použitím metody summary

3 Výsledky

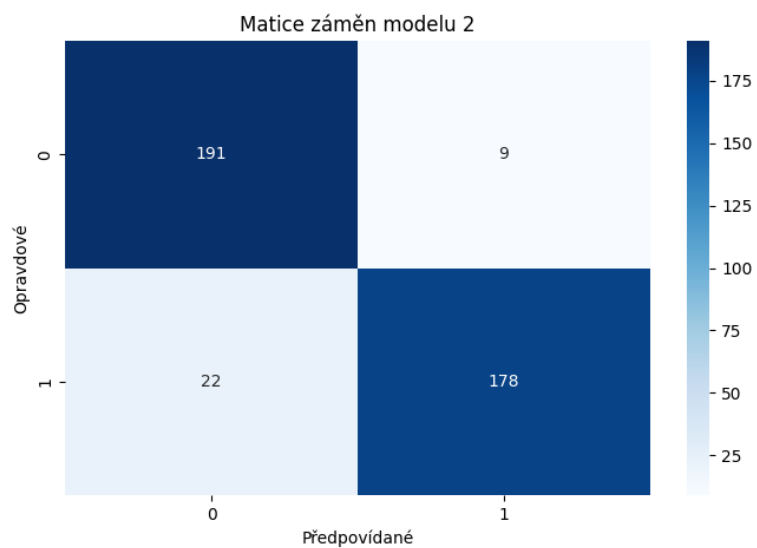
V modelování a evaluaci byla třída auto (Car) zvolena jako pozitivní a motorka (Bike) jako negativní.

Po provedení trénování a testování jednotlivých modelů byla provedena analýza výsledků na testovacích obrázcích. Modely byly analyzovány na základě přesnosti a ztrátové funkce.



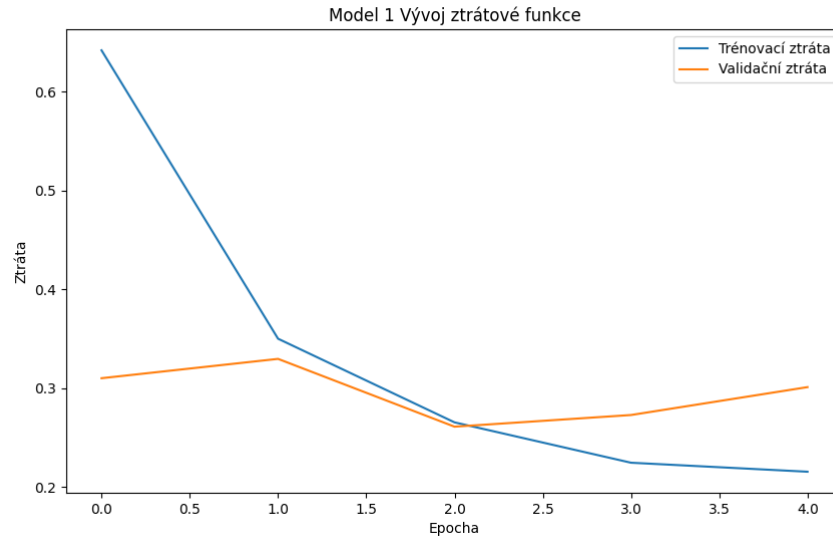
Obrázek 4: Model 1 Matice záměn

Následující matice záměn modelu 1 popisuje, jak náš model klasifikoval obrázky z testovacího setu. Můžeme vidět, že byl relativně úspěšný v rozpoznávání aut a motorek. Jeho přesnost je 91.5%



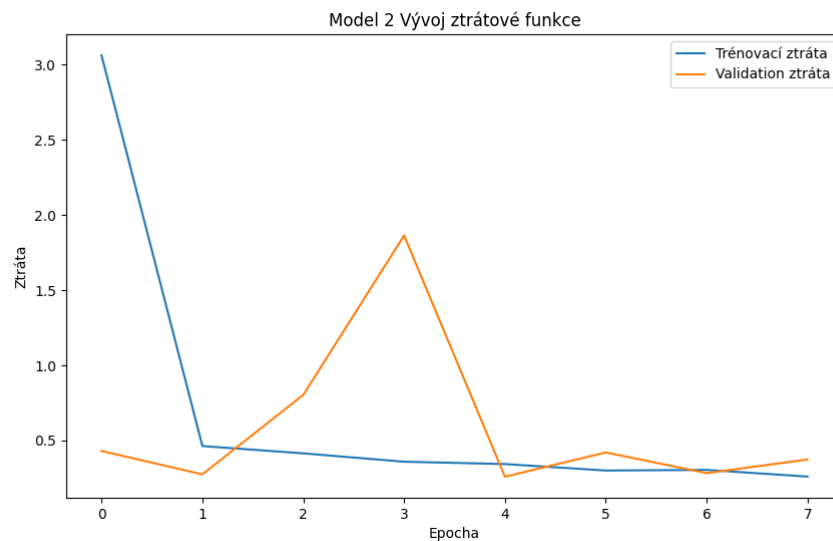
Obrázek 5: Model 2 Matice záměn

Následující matice záměn modelu 2 popisuje, jak náš model klasifikoval obrázky z testovacího setu. Můžeme vidět, že byl o trochu lepší, než model 1. Jeho přesnost je 92.25%



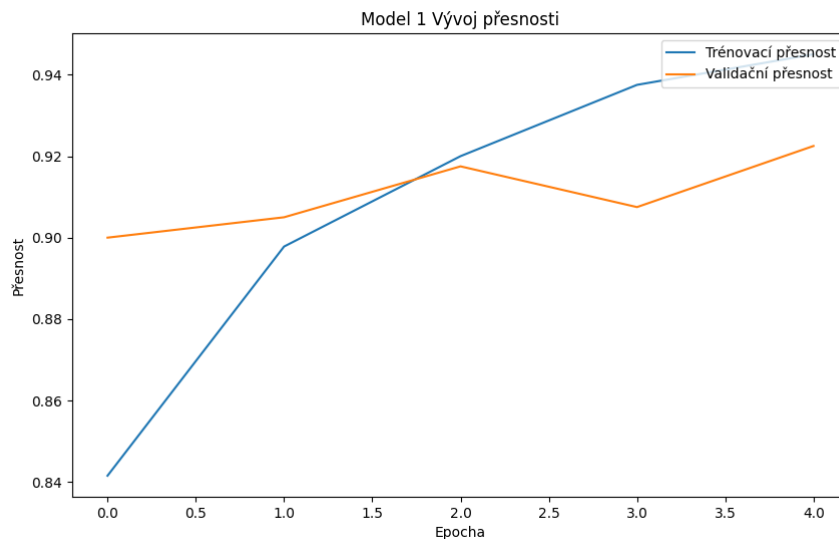
Obrázek 6: Vývoj ztrátové funkce modelu 1

Následující obrázek popisuje vývoj ztrátové funkce modelu 1 v průběhu jeho trénování. Můžeme vidět iniciálně rychlý trend zlepšení modelu, po 3. trénování modelu se zdá, že model se přetrénoval a již neprojevuje výrazné známky zlepšení, naopak se zhoršil na validačních datech.



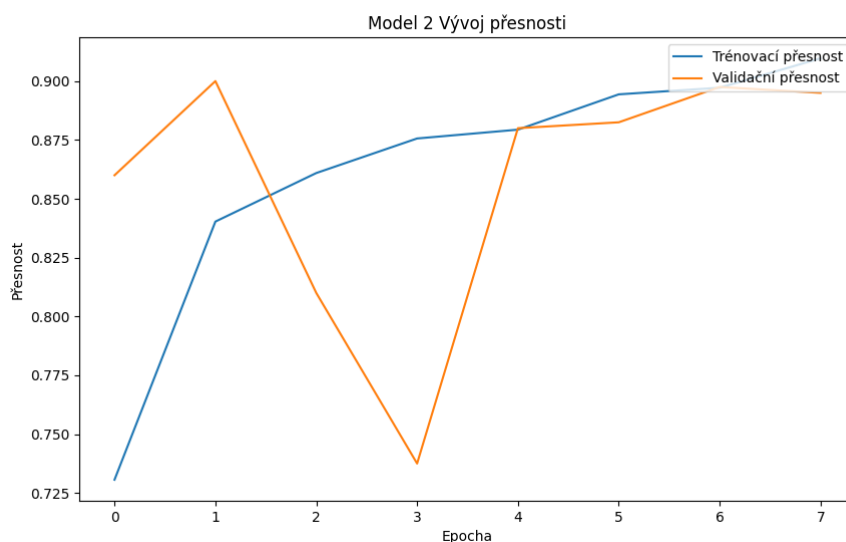
Obrázek 7: Vývoj ztrátové funkce modelu 2

Následující obrázek popisuje vývoj ztrátové funkce modelu 2 v průběhu jeho trénování. Můžeme také vidět iniciálně rychlý trend zlepšení modelu. Zvláštní je vývoj validační ztrátové funkce, která z neznámého důvodu náhle u 3. a 4. výrazně vzroste a následně klesne.



Obrázek 8: Vývoj přesnosti modelu 1

Na grafu vývoje přesnosti modelu 1 můžeme vidět rychlý růst přesnosti na trénovacích obrázcích. Přesnost na obrázcích validačních se zlepšuje pouze do 3. epochy, kde pravděpodobně dochází k přetrénování modelu a následná přesnost upadá.



Obrázek 9: Vývoj přesnosti modelu 2

Na grafu vývoje přesnosti modelu 2 můžeme vidět rychlý růst přesnosti na trénovacích obrázcích. U přesnosti na validačních obrázcích opět dochází k zvláštnímu jevu, kde u 3. a 4. epochy přesnost výrazně klesne, a následně se vrátí opět zpět. Je zde také evidentní, že se model pravděpodobně přetrénuje na trénovacích obrázcích, kde nejpřesnější je na validačních obrázcích na konci 2. etapy.

4 Diskuze

Tato studie prezentuje účinný způsob, jak rozlišovat obrázky do binárních tříd pomocí konvolučních neuronových sítí. Modely ukázaly svou přesnost, i když byly pravděpodobně přetrénovány na trénovacích datech.

Studie otevírá možnost dalšího zlepšení, a to více experimentovat s epochami nebo jinými optimalizačními algoritmy a hodnotami učení. Dále by se dal zvětšit původní dataset a experimentovat s více obrázky, či více třídami. Víde obrázků by mohlo pomoci modelu lépe generalizovat. Dále by mohlo být zajímavé prozkoumat i další architektury neuronových sítí a porovnat je se sítěmi konvolučními.

5 Závěr

V průběhu naší studie jsme se zabývali binární klasifikací motorových vozidel, konkrétně klasifikací aut a motorek. Použili jsme dataset z Kaggle sestávající se z 4000 obrázků. Proces zahrnoval předzpracování obrázků a následné vytváření konvolučních neuronových sítí pro klasifikaci.

Naše výsledky ukazují, že tyto modely dokáží efektivně klasifikovat motorová vozidla.

Tato práce tak představuje slibný krok směrem k automatickému rozpoznávání a klasifikaci obrázků motorových vozidel.

Citace

- [dee23] Deeplizard - building collective intelligence, 2023. Available at: <https://deeplizard.com/> (Accessed: 19. 05. 2023).
- [Sax22] Utkarsh Saxena. Car vs bike classification dataset. Available at: <https://www.kaggle.com/datasets/utkarshsaxena/car-vs-bike-classification-dataset> (Accessed: day month year), 2022.
- [SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2015.

6 Figury

- 1: Příklady obrázků po předzpracování
- 2: Popis modelu 1 použitím metody summary
- 3: Popis modelu 2 použitím metody summary
- 4: Model 1 Matice záměn
- 5: Model 2 Matice záměn
- 6: Vývoj ztrátové funkce modelu 1
- 7: Vývoj ztrátové funkce modelu 2
- 8: Vývoj přesnosti modelu 1
- 9: Vývoj přesnosti modelu 2