

Konvolučné siete

Zadanie viz. prílohu 02-HW.ipynb

Dataset

Dataset je tvorený 5 triedami rôznych dopravných prostriedkov (autá, lode, lietadlá, bicykle, vlaky), obsahuje skoro 200 obrázkov, ktoré sú takmer rovnomerne rozložené do jednotlivých tried. Je vytvorený prehľadávaním a sťahovaním rôznych obrázkov z internetu. Chcel som, aby dataset obsahoval čo najviac prvkov s odlišnými veľkosťami a uhlami pohľadu klasifikovaných objektov a s rôznorodými pozadiami aby to model nemal až také jednoduché a aby sa čo najlepšie naučil rozpoznávať jednotlivé triedy. Preto som zvolil variant dopravných prostriedkov, pretože bolo pomerne jednoduché nájsť dostatok obrázkov, ktoré spĺňali môj zámer.

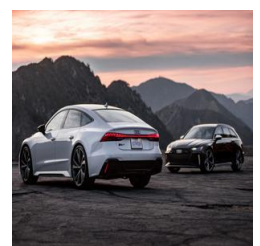
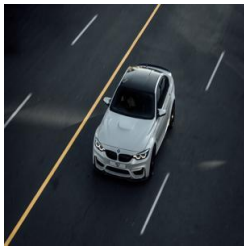
Veľkosti obrázkov boli rôzne v rozmedzí 500 až 2000 pixelov na jeden rozmer, ale v datasete sú kvôli veľkosti datasetu všetky resiznuté na 300x300 pixelov pomocou PIL knižnice.

Predspracovanie obrázkov zahrňuje len ich resiznutie a použitie funkcie `preprocess_image` ktorá je pri každom použití modely iná, a zahrňuje modifikáciu obrázkov, ktorú model vyžaduje.

Augmentácia dát nebola potrebná vďaka dostatočnému počtu obrázkov v datasete a aj po pár epochách bol pretrenovaný model schopný dosiahnuť dobré výsledky.

Ukážky obrázkov:

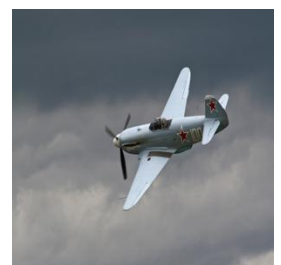
- autá



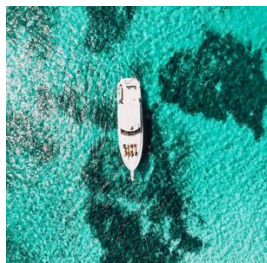
- vlaky



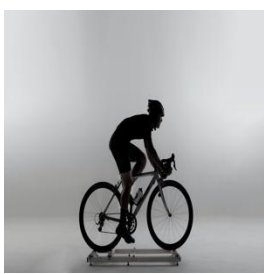
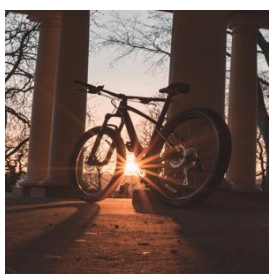
- lietadlá



- **Lode**



- **bicykle**



Modely

Použité modely:

- Xception
- MobileNet
- ResNet50

Nad výberom modelov som príliš nepremýšľal, vybral som 3 spomedzi najpopulárnejších modelov, ktoré sú zároveň implementované v knižnici tensorflow kvôli jednoduchosti.

Štruktúry modelov sa skladajú z konvolučných, pooling a dense vrstiev. Posledných pár vrstiev, ktoré slúžili v pôvodnom modeli na klasifikáciu boli vynechané a nahradené mojimi vlastnými. (v prípade bližšieho záujmu sa dá presná štruktúra pozrieť na pomocou funkcie `model.summary()` v priloženom kóde).

Základná štruktúra siete ostala nezmenená a nebola pretrénovaná, váhy boli už predtrénované na datasete imagenet, ktorý obsahuje 1000 tried s vysokým počtom rôznych obrázkov v každej triede.

Posledné vrstvy boli zamenené za tieto:

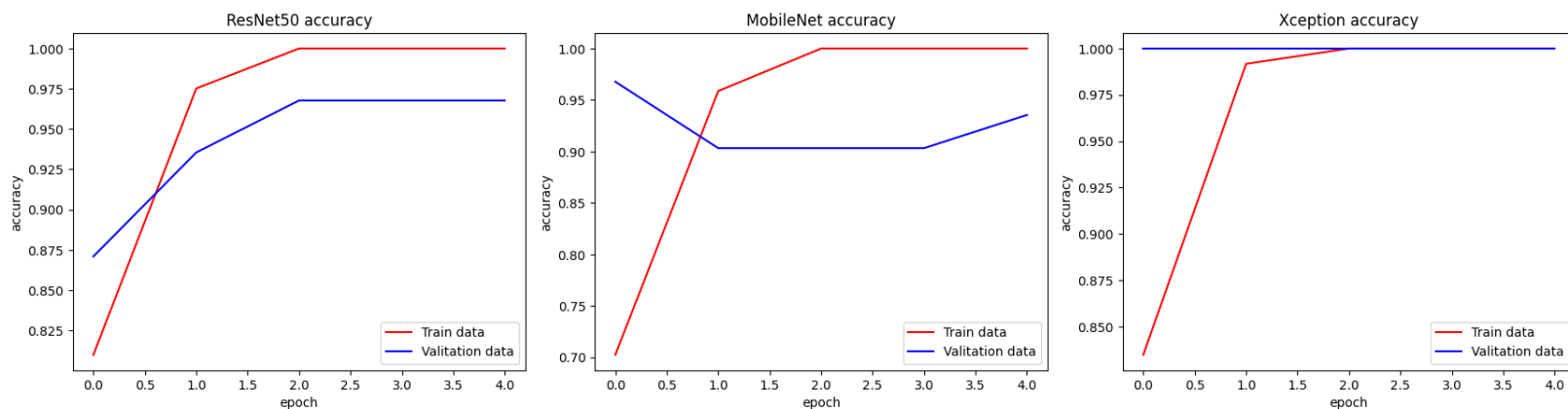
- Flatten()
- Dense(units=classes, activation=linear)
- Activation(activation=softmax)

V podstate som to nechal veľmi jednoduché, len flatten vrstva, ktorá zmrští maticu alebo tenzor do jedného rozmeru. Potom jedna dense vrstva s lineárnou aktiváciou, aby sme dosiahli počet výstupov, koľko máme tried a aktivačná vrstva softmax, ktorá nám vytvorí pravdepodobnosti pre jednotlivé triedy.

Výsledky

Model	Train accuracy	Test accuracy
ResNet50	99.34%	97.37%
MobileNet	98.68%	94.74%
Xception	100%	100%

Grafy pre jednotlivé modely:



Vyhodnotenie:

Všetky modely dosahovali veľmi podobné a dobré výsledky už po pár epochách. Trochu horší výsledok dosahoval MobileNet model. Môže to byť spôsobené tým, že ide o menší model ako zvyšné dva, asi by ešte potreboval viac epôch. Najlepšie výsledky dosahoval Xception model pravdepodobne pretože je najmohutnejší.

Často už po prvej alebo druhej epoche modely dosahovali na validačných dátach výborné výsledky a s počtom epôch už ostávajú na podobných hodnotách, alebo dokonca klesajú, čo je spôsobené miernym overfittingom.

Pretrénovanie už natrénovaných sietí fungovalo výborne, je to samozrejme spôsobené tým, že siete už sú predom natrénované na tisíckach obrázkoch a my len sieť naučíme klasifikáciu do iných tried. Časť siete ktorá sa skladá z konvolučných vrstiev (to je tá časť ktorú sa už nesnažíme pretrénovať) funguje ako feature engineering, sieť si v obrázkoch nájde zaujímavé tvary a farby, ktoré si potom posledná časť siete ktorú už trénujeme spojí s jednotlivými triedami.

Takže prvá časť siete, tým že už je natrénovaná vie dobre rozpoznávať tieto zaujímavé prvky na obrázkoch, podľa ktorých už potom druhá časť vie dobre rozpoznávať objekty, ktoré sa snažíme klasifikovať.