

Česká zemědělská univerzita v Praze  
Provozně ekonomická fakulta  
Katedra informačního inženýrství



**Teze diplomové práce**  
**Rozpoznávání a klasifikace obrazu konvoluční neuronovou sítí**

autor: Bc. Jakub Pekárek  
vedoucí práce: doc. Ing. Arnošt Veselý, CSc.

© 2021 ČZU v Praze

## Abstrakt

Tato diplomová práce v teoretické části prezentuje základní principy, fungování a možnou aplikaci konvolučních neuronových sítí na frameworku Keras a knihovně TensorFlow konfigurovaných pomocí programovacího jazyka Python. Dále charakterizuje jejich přednosti a omezení při plnění různých úkolů jako je rozeznávání znaků, obrázků a jiných vizuálních dat.

V praktické části je testování a využití konvolučních neuronových sítí s různým určením na obvyklých datových souborech a následně jejich použití na jiných datových souborech, pro které nebyly trénovány a naučeny.

V závěru pak popisuje srovnání jejich výsledků dosažených na datových souborech, které nejsou nativně obsaženy v knihovnách Keras a TensorFlow.

**Klíčová slova:** Hluboké učení, Python, TensorFlow, Keras, CUDA, CNN, strojové učení, multimédia, transfer-learning.

# 1 Úvod

V této práci se zabývám problematikou rozpoznávání obrazu počítačem. V současné době je velice efektivní technikou, jak tuto věc s velkou mírou úspěšnosti řešit, využití konvoluční neuronové sítě.

Máme dvě základní možnosti, jak řešit rozpoznávání obrazového materiálu a to: trénovat konvoluční neuronovou síť od začátku, nebo využít již naučenou síť a použít ji na datovém souboru, který chceme zpracovat. Trénování vlastní sítě může být časově velmi náročné zvláště pokud není k dispozici velký datový soubor na trénink. Využití sítě natrénované na velkém objemu trénovacího materiálu, který je dostupný volně na internetu je možností, kterou nesmíme v žádném případě opomenout při rozhodování o tom, jak řešit úkol v oblasti rozpoznávání a klasifikace obrazu. Přináší totiž zpravidla velice kvalitní výsledky.

Toto téma považuji za důležité neboť v současné době je již možné, aby každý, kdo se chce věnovat řešení úloh z oblasti hlubokého učení, tomuto věnoval na svém osobním počítači. K tomuto stavu přispěl rozvoj vykonného hardware a vývoj jednoduše použitelných knihoven jako je Keras.

V této práci jsou předvedeny techniky využití již naučených sítí obsažených v knihovně Keras na jednom z veřejně dostupných datových souborů a je zjištěno v jaké míře jsou konvoluční sítě opakovaně využitelné a jaké výsledky to přináší.

## 2 Cíl práce a metodika

Cílem této práce je pomocí nabytých teoretických znalostí použít natrénované konvoluční neuronové sítě k řešení problému, ke kterému nebyly přímo trénovány s jiným modelem konvoluční neuronové sítě, který byl trénovaný od začátku a tyto výsledky, kterých bylo dosaženo na stejném datovém souboru porovnat.

Pro přípravu a úpravu dat je použit programovací jazyk Python 3.8 v IDE PyCharm a jeho knihovny, zejména NumPy pro matematické úlohy, matplotlib pro zobrazení výsledků grafy.

Samotná konfigurace a následná kompilace modelů konvolučních neuronových sítí je provedena na platformě Keras s využitím funkcí knihovny pro strojové učení Tensorflow.

## 2.1 Teoretická část

V teoretické části jsou popsány základní prvky a funkce a principy neuronových sítí a hlubokého učení a také krátké shrnutí vývoje v posledních dekadách.

## 2.2 Praktická část

Popisuje způsob vývoje vlastního modelu konvoluční neuronové sítě ve frameworku Keras, tak aby dosahoval uspokojivých výsledků na datasetu Dogs vs Cats, který obsahuje 25000 fotografií kočky nebo psa. Poté je datový soubor zmenšen na 2000 trénovacích obrázků a je znovu prováděno ladění s důrazem na co nejlepší kvalitu modelu, s využitím technik pro snížení míry přeučení. S ním jsou poté porovnány výsledky, kterých bylo dosaženo využitím již naučených sítí. Pro testování byly využity tyto naučené sítě obsažené v Kerasu: VGG16, VGG19, InceptionV3, Xception, Densenet121. Výběr proběhl tak, aby byly zastoupeny různorodé architektury, od starších sekvenčních modelů VGG16, až po složité a sofistikované plně propojené architektury jako je Densenet121. Tyto všechny naučené sítě byly použity k testu, jakých výsledků dosahují při nenáročném technice extrakce vah z předtrénované sítě, poté náročnější technice spočívající v použití celé natrénované báze na trénovacím souboru dat. Nakonec následovalo trénování s přizpůsobováním vah, spočívající v odemčení několika vrchních vrstev sítě naučeného modelu, které umožňuje jemné přizpůsobení vah datům, které sítí prochází. Tato technika jemného ladění je z těchto tří nejnáročnější z hlediska využití zdrojů a měla by přinášet nejlepší výsledky při řešení problému binární klasifikace.

## 3 Výsledky a diskuse

Při trénování sítě od začátku bylo zapotřebí vyšší znalosti problematiky a daleko více experimentování s parametry, než při použití naučeného modelu. Docházelo ke změnám v kapacitě plně propojených vrstev, dále celého počtu vrstev v celém modelu, tak aby bylo dosaženo, co nejlepšího výsledku. Někdy model ani při sebevětším ladění nedosáhne přesnosti nad 70%. Naproti tomu transfer learning obsahuje velmi málo možností nastavení ve srovnání s modelem trénovaným od nuly. Často ale dosahuje řádově lepších výsledků. Všechny použité techniky v této práci, extrakce příznaků, extrakce příznaků s rozšířením, a jemné ladění několika vrchních vrstev, mají svoje místo při praktickém používání technologie konvolučních sítí.

Obecně lze říci, že trénování sítě od nuly má největší potenciál v případě příhodných podmínek a opravdové nutnosti získat ty nejlepší výsledky, s tím, že problém budeme řešit opakovaně a máme dostatek času připravit řešení na míru konkrétnímu problému. V opačném případě při řešení jedné úzce definované nebo specializované úlohy, pro příklad hledání jedné značky automobilu mezi ostatními, lze velmi dobře a rychle použít již naučenou síť a pomocí již známých technik dosáhnout výsledků mnohem rychleji při stejné nebo obdobné kvalitě.

Při práci na malém datovém souboru je třeba používat prostředky proti přeučení. Jak bylo zjištěno, rozdíl v použití nebo nepoužití vrstvy Dropout je signifikantní, její použití zlepší kvalitu modelu, tím že omezí přeučení o polovinu nebo více. Naproti tomu tyto techniky nepomohou, pokud je špatně zvolena kapacita modelu, tedy velikost plně propojených vrstev. Dalším poznatkem je, že na datových souborech o velikosti 2000 a 22000 obrázků, lze dosáhnout velice různých výsledků. Zdá se, že s 22000 trénovacích vzorků lze natrénovat velmi dobrý a prakticky použitelný model. V případě řešení problému s velmi malým datovým souborem, tedy například 2000 obrázků, by první kroky měly směřovat k využití již naučeného modelu a prověření, jakých výsledků může být dosaženo.

Testováním bylo dosaženo těchto výsledků:

**Densenet121** dosahovala 98% tréninkové úspěšnosti při validační ztrátě 0.07%.

**Xception** dosahovala 100% tréninkové úspěšnosti při validační ztrátě 0.1%.

**InceptionV3** dosahovala 100% tréninkové úspěšnosti při validační ztrátě 0.2%.

**VGG16** dosahovala 97% tréninkové úspěšnosti při validační ztrátě 0.23%.

**VGG19** dosahovala 97% tréninkové úspěšnosti při validační ztrátě 0.32%.

**Sekvenční model** s malým datasetem 77% tréninkové úspěšnosti při validační ztrátě 43%.

Za nejdůležitější kritérium považujeme validační ztrátu, ukazuje schopnost sítě zobecňovat naučené vzory. Kvalitativně nejlepším modelem je poté Densenet121 a hned za ním Xception s validační ztrátou blížící se nule a testovací přesností až 98%. Pokud však vezmeme v úvahu technologické rozdíly a pokročilou architekturu Densenet121, nevedou si mnohem starší VGG16 a VGG19 vůbec špatně.

## 4 Závěr

Hlavní myšlenkou této práce a otázkou, na kterou bylo třeba odpovědět je prozkoumání výhod a nevýhod využití naučené sítě na řešení obdobného problému (transfer learning) a také toho, jak je použitelné v praktických úlohách strojového učení při rozpoznávání objektů. V tomto případě odlišení kočky od psa na obrazovém materiálu různé kvality, za podmínek, které nejsou pro trénink neuronové konvoluční sítě ideální. Také byly nastaveny další podmínky, tak aby byla situace komplikovanější, nebyl k dispozici výkonný hardware a datový soubor byl ještě razantně zmenšen, tak aby více odpovídal reálnému použití této technologie.

Pro mě osobně mělo toto téma veliký přínos a jen samotný úvod do této problematiky bude mít velké využití i v mém profesním životě. Problematiku "vidění počítačů" považuji za fascinující s velkým potenciálem do budoucna, byť je pravděpodobné že dojde na další dobu stagnace, jako se již v historii několikrát stalo. Vzhledem k tomu, že problematika rozpoznávání statických obrazových dat, je již na pomyslném vrcholu a pravděpodobně ještě chvíli bude, než dojde k dalšímu mezníku v technologii.

Pokračování této práce by mohlo řešit již skutečný problém, rozpoznávání fotografií kočky domácí od psa domácího by bylo užitečné leda pro myš nebo jiného hlodavce obývajícího nějaké lidské obydlí. Zajímavým problémem by mohla být klasifikace obrazového materiálu, zdali obsahuje obraz zvířete podléhající ochraně CITES. Nebo posun k rekurentním sítím a využití jejich paměti k rozpoznávání objektů ve videosouborech.

## Seznam použité literatury

- [1] CHOLLET, François: Deep learning v jazyku Python: knihovny Keras, Tensorflow. Praha: Grada Publishing, 2019. Knihovna programátora, ISBN 978-80-247-3100-1.
- [2] What is Transfer Learning?, [online], [cit. 2020-03-25], Dostupné z WWW: <https://towardsdatascience.com/what-is-transfer-learning-8b1a0fa42b4>
- [3] Keras vs Tensorflow: Must Know Differences!, [online], [cit. 2020-04-27] Dostupné z WWW: <https://www.guru99.com/tensorflow-vs-keras.html>
- [4] TensorFlow Core v2.6.0 dokumentace [online], [cit. 2021-10-21], Dostupné z WWW: [https://www.tensorflow.org/api\\_docs/python/tf/keras/utils/image\\_dataset\\_from\\_directory](https://www.tensorflow.org/api_docs/python/tf/keras/utils/image_dataset_from_directory)
- [5] Francois Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, [online], (PDF), [cit. 2020-10-21] (PDF) Dostupné z WWW: [https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Chollet\\_Xception\\_Deep\\_Learning\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf)