

Smart gallery

Matěj Boura

Faculty of Informatics and Management

University of Hradec Kralove,

Hradec Kralove, Czech Republic

bourama1@uhk.cz

Abstrakt—Tato práce se zabývá vývojem inteligentní galerie, využívající umělou inteligenci k automatické klasifikaci fotografií do specifických kategorií. Implementace této galerie by mohla sloužit k vytvoření uživatelsky příjemného prostředí, které umožňuje uživatelům snadno procházet a organizovat svou fotografickou sbírku, primárně u turistických fotografií. Nejdříve bylo nutné prozkoumat techniky hlubokého učení, datasety a metody strojového vidění k efektivnímu a přesnému rozdělování obrázků. Z důvodu vysoké výpočtové náročnosti bylo nutné z datasetu vybrat menší dataset kategorií, které se přímo týkají mého cíle. Výsledkem je model se zhruba o 9% horší přesností, ale skoro 4 násobně rychlejší během predikce oproti jeho konkurenci trénované na plném datasetu.

Klíčová slova—photography; scene classification; Convolutional neural network; TensorFlow; Transfer Learning

I. ÚVOD

A. OBSAH

V dnešní době, kdy obrazová data zaplavují digitální prostor, se stává stále obtížnějším a časově náročnějším úkolem efektivně organizovat a vyhledávat mezi tisíci fotografiemi. Kvůli tomuto narůstajícímu problému se tato práce zabývá vytvořením chytré galerie, která využívá pokročilé techniky umělé inteligence k automatické klasifikaci obrázků do relevantních kategorií. S nástupem pokročilých technologií v oblasti počítačového vidění a umělé inteligence vznikla řada efektivních modelů pro klasifikaci, avšak s tím přišly i nové výzvy spojené s vysokými nároky na výpočetní kapacity a přílišnou komplexností těchto modelů.

Tato práce se zaměřuje na analýzu a řešení této problematiky. Motivací je nejen optimalizace procesu klasifikace obrazů, ale zejména snaha o snížení výpočetní náročnosti a komplexity stávajících řešení. V dnešním prostředí, kde rychlost, efektivita a šetrnost s výpočetními zdroji hraje klíčovou roli, je naléhavě potřebné hledat inovativní přístupy k dosažení optimálního vyvážení mezi přesností a efektivitou klasifikace obrazu.

B. DEFINICE PROBLÉMU

Podobný cíl si kladla práce [5] z roku 2021. Cílem navrhovaného projektu bylo vytvořit inteligentní systém pro správu galerie s využitím technologie Google Vision pro zpracování a analýzu obrazu. Zahrnuje server, mobilní rozhraní a webové rozhraní pro nahrávání, zpracování a dotazování fotografií na základě objektů a OCR (Optical

Character Recognition). Tato práce problém s výpočetním výkonem obešla využitím Google Vision API. Což by ovšem mohlo být viděno jako výrazný problém pro lidi, kteří jsou více citliví na své soukromí, z důvodu nutnosti nahrávat své fotografie na servery Googlu. Dalším problémem je funkčnost takové aplikace jen při připojení s dostatečnou šíří pásma pro nahrání obrázků. Dalším problémem v této studii je velmi omezené testování a analýza, která byla prováděna na pouhých 100 obrázcích rozdělených do poměrně obecných 7 kategorií.

Další z aktuálních prací, která se snaží o klasifikaci scény u primárně turistických fotografií a destinací je z roku 2020. Tato práce [4] konkrétně pojednává o budoucnosti globálního cestovního ruchu a jeho ovlivnění technologickým pokrokem, včetně umělé inteligence, strojového učení a robotiky, které směřují toto odvětví k automatizovanější budoucnosti. Tato práce dosáhla velmi slušné přesnosti smíšením přístupu Transfer learningu a kombinací sítí Dense-Net169 a Xception, ale bohužel s ní nepůjde naše práce porovnat jelikož si autoři vytvořili vlastní dataset složený z méně kategorií a se zhruba pouze 31 000 obrázky, který není veřejně dostupný.

Poslední prací [3] kterou bych chtěl zde v úvodu zmínit je práce zabývající se real-time zpracováním obrazových dat. V této práci z února letošního roku se její autoři zabývají použitím techniky YOLO (You Only Look Once) pro detekci objektů. Tato technika skenuje celý obraz a odhaduje ohraničující boxy pomocí konvolučních sítí a pravděpodobnostních tříd. Tato metoda se ale na první pohled nezdá vhodná ke klasifikaci scény. Dalším již standardním problémem shodným s většinou prací je dataset. Článek totiž výslovně neuvádí konkrétní datovou sadu ani data použitá při experimentech nebo implementaci navrhované metody.

V následujících kapitolách budou detailně rozebrány metody, které byly použity při vytváření chytré galerie, včetně popisu algoritmů, datových sad, a výsledků analýz a testování. Nakonec proběhne diskuze o možných rozšířeních a zlepšeních tohoto systému. Cílem této práce je prezentovat jiný přístup k optimalizaci klasifikace obrazů, než v předchozích pracích. Tento přístup by měl překlenout současné výzvy spojené s vysokou náročností a komplexností stávajících modelů.

II. NOVÉ ŘEŠENÍ

A. DATASET

Pro práci byl zvolen dataset Places365[6] jako základní zdroj obrazových dat. Tento rozsáhlý dataset obsahuje širokou škálu scén z různých prostředí, což nám umožňuje vytvořit robustní a rozmanitý model klasifikace. Avšak, s ohledem na specifický cíl této práce, optimalizace rychlosti a zachování přesnosti, byla provedena selekce a redukce datasetu. Snížení původních 365 kategorií na 100 klíčových kategorií, přičemž každou z nich reprezentuje 1000 vybraných obrázků. Tato redukce nejenže snižuje náročnost výpočtů, ale také umožňuje zaměřit se na klíčové aspekty klasifikace a dosáhnout vyšší efektivity při zachování významného reprezentativního vzorku pro každou kategorii. Tímto způsobem vznikl optimalizovaný a vyvážený dataset, který slouží jako základ pro trénink, analýzy, testy a jejich výsledky.

B. CNN (CONVOLUTIONAL NEURON NETWORK)

Konvoluční neuronová síť (CNN) je hluboká, dopředná umělá neuronová síť, která zachovává hierarchickou strukturu tím, že se učí reprezentace interních funkcí a zobecňuje funkce v běžných obrazových problémech, jako je rozpoznávání objektů a počítačové vidění[7]. CNN se skládají z umělých neuronů uspořádaných do konvolučních vrstev, kde každá vrstva má výstupní matice spojené se vstupními maticemi prostřednictvím synapsí spojených s konvoluční maticí. CNN používají konvoluční operaci jako jednu ze svých vrstev místo plně připojené vrstvy[8]. Tyto Conv2D vrstvy používají filtry k postupnému skenování vstupních dat a extrakci lokálních vzorů a rysů. Jsou schopny zachytit hranice, textury a další důležité části obrázků. Tyto sítě provádějí operace podobné filtrům zpracování obrazu a používají se v úkolech, jako je klasifikace obrazu, detekce objektů a sémantická segmentace. CNN jsou široce používány v hlubokém učení pro rozpoznávání vzorů a analýzu obrazu[9].

Další důležitou součástí CNN jsou pooling vrstvy. Pooling vrstvy slouží ke snižování rozměrů dat tím, že redukuje rozlišení. Nejčastěji používaným typem je MaxPooling, který zachovává nejvyšší hodnoty z každé matice.

Předposlední základní vrstvou je vrstva Flatten. Ta je umístěna po konvolučních a pooling vrstvách a slouží k transformaci vstupních dat z dvourozměrného formátu (matice) na jednorozměrný vektor. Tím dochází k převedení prostorové struktury dat na lineární formu, což umožňuje předání těchto dat plně propojeným vrstvám.

Plně propojeným vrstvami jsou reprezentovány Dense vrstvy. Tyto vrstvy obsahují neurony spojené s každým neuronem z předchozí vrstvy, což vytváří plně propojenou strukturu. V těchto vrstvách se provádí operace vah a biasů, což umožňuje sítím se učit složitější vzory a vztahy mezi různými částmi dat. Dense vrstvy jsou obvykle umístěny na konci architektury sítě a často slouží k převedení extrahovaných rysů na výstupní formu odpovídající specifickému problému, jako je klasifikace nebo regrese.

C. AUGMENTACE DAT A DROPOUT

Augmentace dat je technika, která zahrnuje vytváření nových trénovacích příkladů pomocí různých transformací původních dat. To může zahrnovat rotace, horizontální či vertikální zrcadlení, změnu jasu a kontrastu, nebo základní geometrické transformace. Cílem augmentace dat je vytvořit rozmanitější trénovací sadu, což pomáhá modelu lépe se adaptovat na různorodé podmínky a snižuje riziko přeučení (overfitting) na konkrétních vzorcích v datech. Tato technika napomáhá zlepšit obecnost modelu, což v konečném důsledku může vést ke zvýšení jeho validační přesnosti, zejména v situacích, kde jsou dostupná trénovací data omezená. Augmentace dat tedy přináší výhody v posílení schopnosti modelu generalizovat a lépe vyhodnocovat nová, dosud neviděná data.

Kromě augmentace dat může být dalším efektivním nástrojem pro zlepšení generalizace modelu přidání dropout vrstvy. Dropout je technika regularizace, která náhodně deaktivuje („vypouští“) některé jednotky během trénování. Tímto způsobem se model naučí nezávislost různých částí sítě a snižuje se riziko přeučení. Dropout poskytuje robustnost vůči malým fluktuacím v datech a umožňuje modelu efektivněji se přizpůsobit novým vzorům. Jeho přidání do architektury neuronové sítě tak může významně přispět k lepšímu obecnému výkonu modelu a zvýšení jeho schopnosti dosáhnout vyšší validační přesnosti.

D. TRANSFER LEARNING

Přenosové učení je technika používaná při klasifikaci obrazových scén k využití předem trénovaných modelů z jiných domén. Pomáhá při řešení problémů s omezenými tréninkovými daty a různými rozměry obrazu. V kontextu neuronových sítí, to znamená převzít předtrénovaný model (tzv. pre-trained model), který byl natrénován na obecných datech, a adaptovat ho na nový úkol nebo specifický dataset. Výhody transfer learningu spočívají v tom, že předtrénovaný model již obsahuje naučené znalosti o obecných rysech a vzorech v datech, což může značně urychlit a zvýšit úspěšnost trénování nového modelu. Transfer learning je často využíván v oblasti vizuálního zpracování obrazu, kde jsou k dispozici velké předtrénované modely pro širokou škálu úkolů, například ImageNet dataset. Tento postup minimalizuje počet epoch a vrstev potřebných k trénování sady dat a urychluje CNN. Usnadňuje také trénování malých datových sad bez nadměrného přizpůsobení. Dle této studie[1] je s využitím techniky přenosového tréninku CNN, možné dosáhnout až 95% přesnosti predikce na nové datové sadě o velikosti pouhých 75 obrázků.

Studie[2] hodnotila výkon různých architektur CNN, včetně ResNet50, VGG16, VGG19, InceptionV3 a EfficientNet, na úlohu detekce subjektů. Model ResNet50 vykazoval nižší výkon ve srovnání s jinými architekturami, naopak model EfficientNet výrazně překonal očekávání vzhledem k nižšímu stupni složitosti, který lze vyčíst z velmi nízkého počtu parametrů oproti ostatním modelům CNN. Tato studie také

zkoumala dopad zkreslení obrazu, jako je rozostření a šum, na klasifikátory obrazu založené na CNN. Speciálně byly zkoumány Gaussovské rozostření a Salt-and-pepper šum a byly analyzovány jejich účinky na klasifikační výkon. U všech modelů kromě již dříve zmíněného ResNet50, nedošlo k nijak drastickému zhoršení testovaných metrik.

III. IMPLEMENTACE ŘEŠENÍ

Níže na Obrázku 1 lze vidět flowchart toho jak by aplikace a průchod uživatele skrz ni mohl vypadat.

A. NÁSTROJE

Pro implementaci modelu chytré galerie byly vybrány nástroje, které zajišťují efektivní a jednoduchý vývoj. Pro práci s hlubokým učení byl zvolen TensorFlow, populární framework, který poskytl potřebnou flexibilitu a výkon. Pro vizualizaci dat a analýzu výsledků knihovna Matplotlib, umožňující přehledné zobrazení výsledků a analýzu trendů. Pro přístupu k předtrénovaným modelům byl nainstalován TensorFlow Hub, usnadňující práci s existujícími modely a urychlující vývoj díky jeho jednoduché integraci s již nainstalovaným Tensorflow frameworkem. K implementaci některých částí modelů byly dále využity moduly z knihovny Keras Applications. Tato kombinace nástrojů umožnila efektivně vytvořit a optimalizovat model chytré galerie s ohledem na specifické požadavky na rychlost a přesnost klasifikace.

B. ARCHITEKTURA MODELU

Nejdříve byl vytvořen a trénován model, který lze vidět níže v ukázce kódu. Tento model kombinuje konvoluční vrstvy s plně propojenými vrstvami pro účely extrakce a interpretace hierarchických rysů ve vstupních datech s cílem dosáhnout správné klasifikace. První normalizační vrstva pouze mění rozměry obrázků aby se standardizovalo s čím model bude pracovat. O průběhu tvorby a tréninku informuje algoritmus 1.

```
model = Sequential([
    normalization_layer,
    layers.Conv2D(16, 3, padding='same',
                  activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same',
                  activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same',
                  activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

Tento trénink ovšem poukázal na problém s nízkou validační přesností, která oproti té trénovací nerostla stejnou rychlostí, naopak spíše stagnovala až klesala. Tento problém se podařilo vyřešit přidáním augmentační vrstvy, kterou můžete vidět v podobě níže, na první pozici tohoto modelu.

Data: Dataset

Result: Trained Model

```
1 Load the dataset;
2 Split the dataset into training and testing sets;
3 Preprocess the training and testing sets;
4 Create a normalization layer;
5 Build a model with the normalization layer;
6 // Train Model
7 Initialize number of epochs (num_epochs);
8 for epoch = 0 to num_epochs do
9     for each batch in training set do
10         Train the model on the training set;
11         Evaluate the model on the validation set;
12         Compute validation accuracy;
13     end
14 end
15 if validation accuracy does not improve then
16     Create a augmentation layer;
17     Build a model with the addition of augmentation
    layer;
18 end
19 // Train Second Model
20 for epoch = 0 to num_epochs do
21     for each batch in training set do
22         Train the model on the training set;
23         Evaluate the model on the validation set;
24         Compute validation accuracy;
25     end
26 end
```

Algorithm 1: Vytvoření a trénink modelů

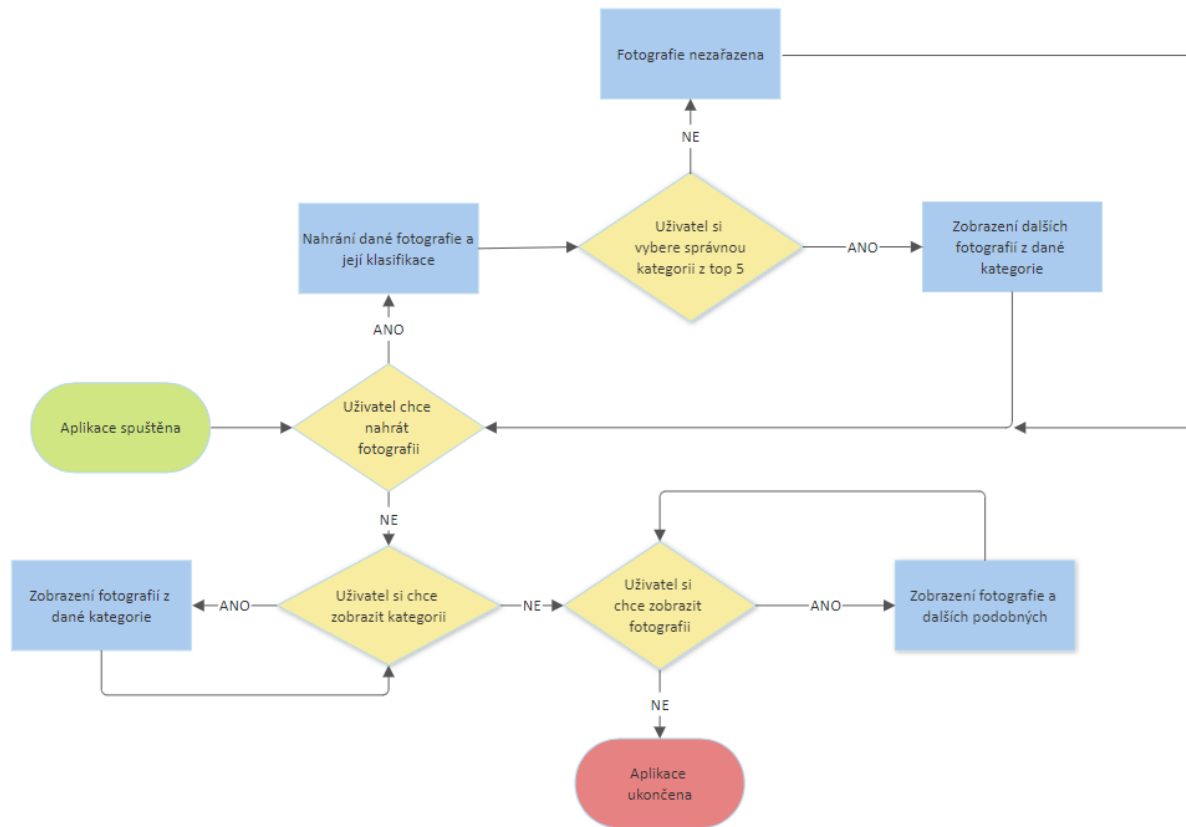
C. AUGMENTACE DAT

```
data_augmentation = keras.Sequential(
[
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
],
)
```

V ukázce kódu výše lze vidět specifické augmentační vrstvy, jejich výběr byl motivován snahou o dosažení vyvážené a realistické augmentace trénovacích dat. Tato kombinace augmentačních operací se zaměřuje na vytváření variací v datech, které jsou v souladu s přirozenými transformacemi, jimž jsou reálné fotografie obvykle vystaveny.

`layers.RandomFlip("horizontal")` simuluje horizontální zrcadlení, což odpovídá běžným scénářům, kdy jsou objekty nebo postavy ve scéně obráceny. Tato operace může zároveň přispět k větší robustnosti modelu vůči různým orientacím objektů.

`layers.RandomRotation(0.1)` přidává náhodné rotace do trénovacích dat. To je užitečné pro zajištění, že model je



Obrázek 1: Flowchart fungování aplikace

Zdroj: Autor

schopen pracovat s obrazy i v různých úhlech natočení, což zvyšuje jeho schopnost generalizace.

layers.RandomZoom(0.1) simuluje náhodné přiblížení, což odpovídá situacím, kdy jsou objekty ve scéně zobrazeny v různých velikostech. Tato operace pomáhá modelu lépe porozumět a zvládnout variabilitu velikosti objektů v datech.

Níže na Obrázku 2 můžete vidět příklad toho jaké změny takováto augmentace může generovat.

D. TRANSFER LEARNING MODELÝ

Pro zjednodušení práce byly použity modely skrze TensorFlow Hub. Zde byly vybrány 4 modely vyhovující požadavkům na použité nástroje (TensorFlow 2) a na typ úkolu (scene classification). Konkrétně se jednalo o modely MobileNetV2, EfficientNetV2, ResNetV2 a InceptionV3, které byly všechny přidány do Hubu společností Google. Všechny kromě Inception byly nastaveny na velikost obrázků z datasetu této práce, a u modelu Inception šlo pouze o drobný rozdíl. Poté byly podrobeny tréninku jak lze vyčíst z algoritmu 2. Nakonec proběhlo jejich porovnání o čemž se lze dočíst v kapitole o testování.

Důležitou součástí využití transfer learningu v CNN je nahrazení poslední plně propojené vrstvy modelu. Tato nahrazená vrstva by měla mít počet neuronů odpovídající počtu

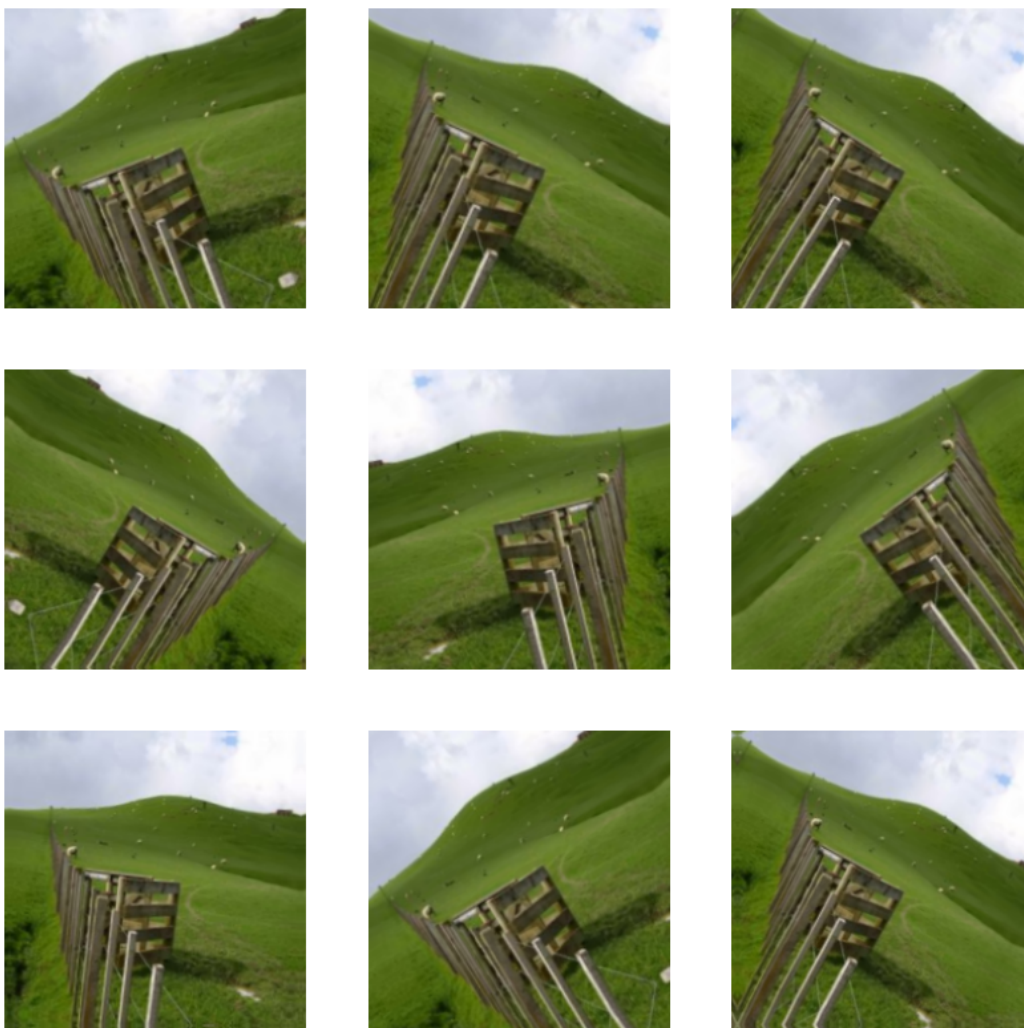
Data: Dataset, Pretrained Models M_i :

EffNet, MobNet, ResNet, Inception

Result: Best Trained Model

- 1 **Load** the dataset;
- 2 **Split** the dataset into training and testing sets;
- 3 **Preprocess** the training and testing sets;
- 4 **for each pretrained model** M_i **do**
- 5 **Load** pretrained model M_i ;
- 6 **Replace** the final classification layer(s) of M_i with a new layer for your task;
- 7 // Train Modified Model
- 8 **Train** the modified model on the training set for *num_epochs*;
- 9 // Evaluate Model
- 10 **Evaluate** the fine-tuned model on the testing set;
- 11 **Compute** performance metrics;
- 12 **Store** the model and it's performance metrics for comparison;
- 13 **end**
- 14 // Select Best Model
- 15 **Select** the model with the best performance on the testing set;

Algorithm 2: Využití Transfer Learningu s předtrénovanými modely



Obrázek 2: Příklad augmentace obrázku

Zdroj: Autor

tříd (num_classes), což je důležité pro adaptaci modelu na konkrétní klasifikační úkol. Proto všechny modely transfer learningu vypadaly po doplnění o potřebné vrstvy zhruba jako v kódu níže.

```
model = tf.keras.Sequential([
    data_augmentation ,
    normalization_layer ,
    feature_extractor_layer ,
    tf.keras.layers.Dense(num_classes)
])
```

Kde vrstva feature_extractor_layer značí samotné Keras vrstvy získané z TensorFlow Hubu již upravené na rozměry obrázků z datasetu.

IV. TESTOVÁNÍ VYVINUTÉ APLIKACE

A. DEFINICE TESTOVÁNÍ

V procesu testování a srovnání výkonu navržených modelů byl uplatněn systematický přístup za účelem získání objektivních výsledků. Prvním krokem bylo sestavení reprezentativního datového souboru obsahujícího rozmanitý

obrazový materiál ze všech kategorií. Tím bylo zajištěno, že testovací data odrážejí skutečné podmínky, se kterými by měly modely pracovat v reálném prostředí. Zároveň byly samozřejmě vybrány pouze data, na kterých modely nebyly trénovány, aby nebyly získané výsledky zkreslené.

V kontextu této práce je důležité zdůraznit, že přímé srovnání s modely od jiných autorů nebylo proveditelné z důvodu předchozí úpravy datasetu. Práce se zaměřila na vytvoření a optimalizaci modelů pro specifické požadavky této chytré galerie, což zahrnovalo kroky, jako byla selekce a augmentace dat. Důvodem, proč nebylo možné provést přímé srovnání s modely od jiných autorů, spočívá v zásadním zúžení množství kategorií v rámci této práce.

Důležitým aspektem této práce bylo snížit komplexnost modelů, které byly původně navrženy a testovány na plném datasetu. S cílem dosáhnout redukce časové a výpočetní složitosti, což by zvýšilo efektivitu a výkon v kontextu chytré galerie. Takže jedním z nejdůležitějších ukazatelů byla rychlost predikce.

B. SROVNÁNÍ MODELU

V rámci tohoto výzkumu bylo provedeno srovnání čtyř modelů získaných pomocí transfer learningu. Primárním zájmem bylo zhodnotit přesnost těchto modelů v kategoriích Top-5 a Top-1, přičemž byl kladen důraz na co nejlepší časovou efektivitu a minimalizaci výpočetních nároků. Níže na Obrázku 3 lze vidět srovnání přesností Top-1 a Top-5 kategorií a hned na dalším obrázku 4 lze vidět i srovnání rychlosti predikce u jednotlivých modelů.

Z výše zmíněných grafů lze vyčíst klíčové výsledky srovnání jednotlivých modelů. Z celkového pohledu je nejvýraznějším a nejúspěšnějším modelem EfficientNet, který dosáhl absolutně nejvyšší přesnosti jak v Top-1 tak v Top-5. Tento model se vyznačuje schopností poskytovat precizní klasifikace při současném udržení rozumné časové a výpočetní náročnosti. EfficientNet tak představuje optimální kombinaci přesnosti a efektivity, což ho činí ideálním kandidátem pro naše specifické požadavky.

C. SROVNÁNÍ PREDIKCÍ

Pro účely srovnání výkonu byly porovnány predikce vybraného modelu z naší práce s modelem, který byl trénován na plném datasetu. Výsledkem těchto srovnání bylo významné zkrácení časových a výpočetních nároků našeho modelu na přibližně jednu čtvrtinu v porovnání s modelem natrénovaným na plném datasetu. Toto srovnání potvrzuje efektivitu našeho přístupu k redukci komplexity modelů. Změny provedené v architektuře a trénovacím procesu vedly k značnému zvýšení rychlosti predikcí. Čímž bylo dosaženo klíčového cíle práce – optimalizace výkonu modelů s ohledem na časové a výpočetní omezení.

Dle oficiálního zdroje plného datasetu [6] dosáhl model VGG16, který byl vybrán k porovnání s naším modelem jelikož dosáhl nejlepších výsledků, přesností 55,24% v Top-1 a 84,91% v Top-5 přesnosti. Což je oproti našemu modelu, jenž dosáhl zhruba 46% přesnosti v kategorii Top-1 a 77% přesnosti v kategorii Top-5, lepší výsledek, ale jak bylo vysvětleno dříve je to za cenu čtyřnásobné doby predikce.

Níže na Obrázku 5 můžete vidět ukázkou predikce na modelu z této práce (vlevo) a modelu trénovaném na plném datasetu (vpravo). Včetně výše zmíněných časových rozdílů na prvním řádku obrázku.

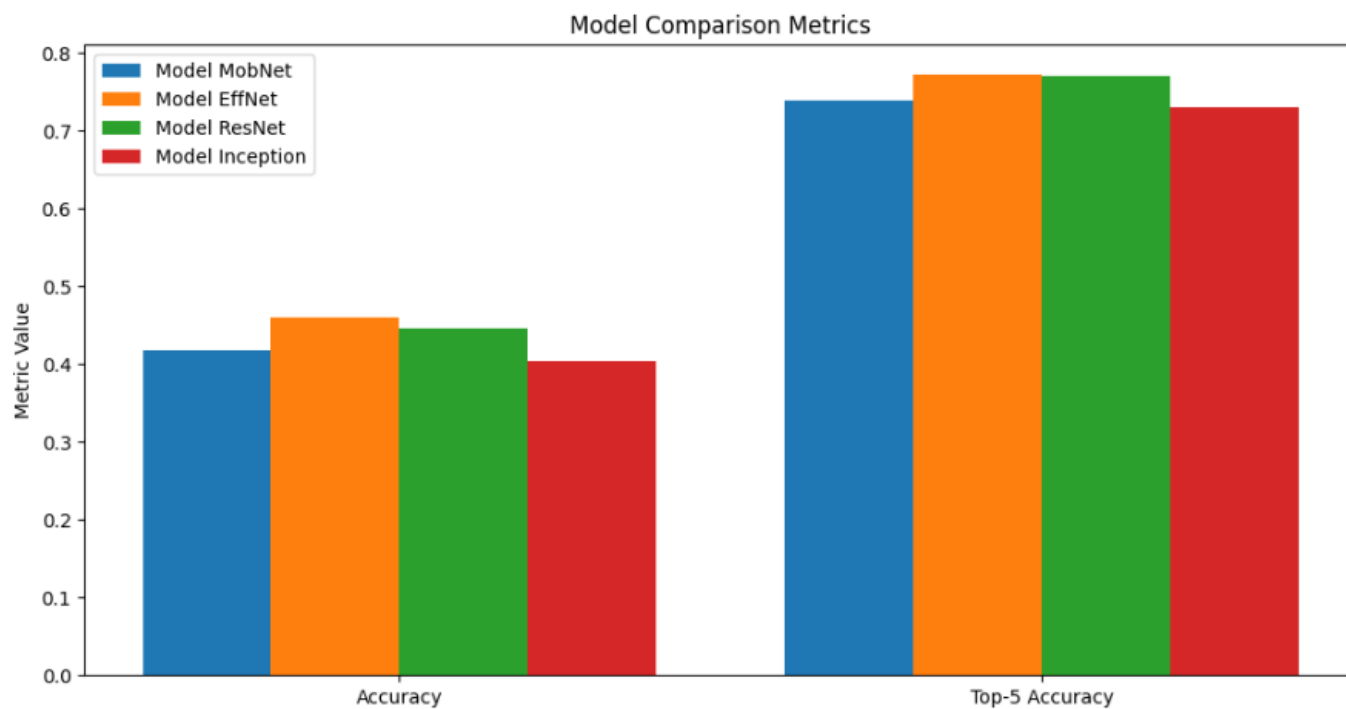
V. ZÁVĚRY

V rámci této práce byl zkoumán proces vytváření chytré galerie s využitím pokročilé klasifikace obrazů za účelem zlepšení efektivity. Cílem bylo dosáhnout zvýšení rychlosti kategorizace fotografií a současně minimalizovat snížení přesnosti na přijatelnou úroveň. Výsledky práce představují signifikantní zlepšení v oblasti efektivity, kdy bylo dosaženo až čtyřnásobného zvýšení rychlosti klasifikace, a to za cenu snížení přesnosti o zhruba 9%.

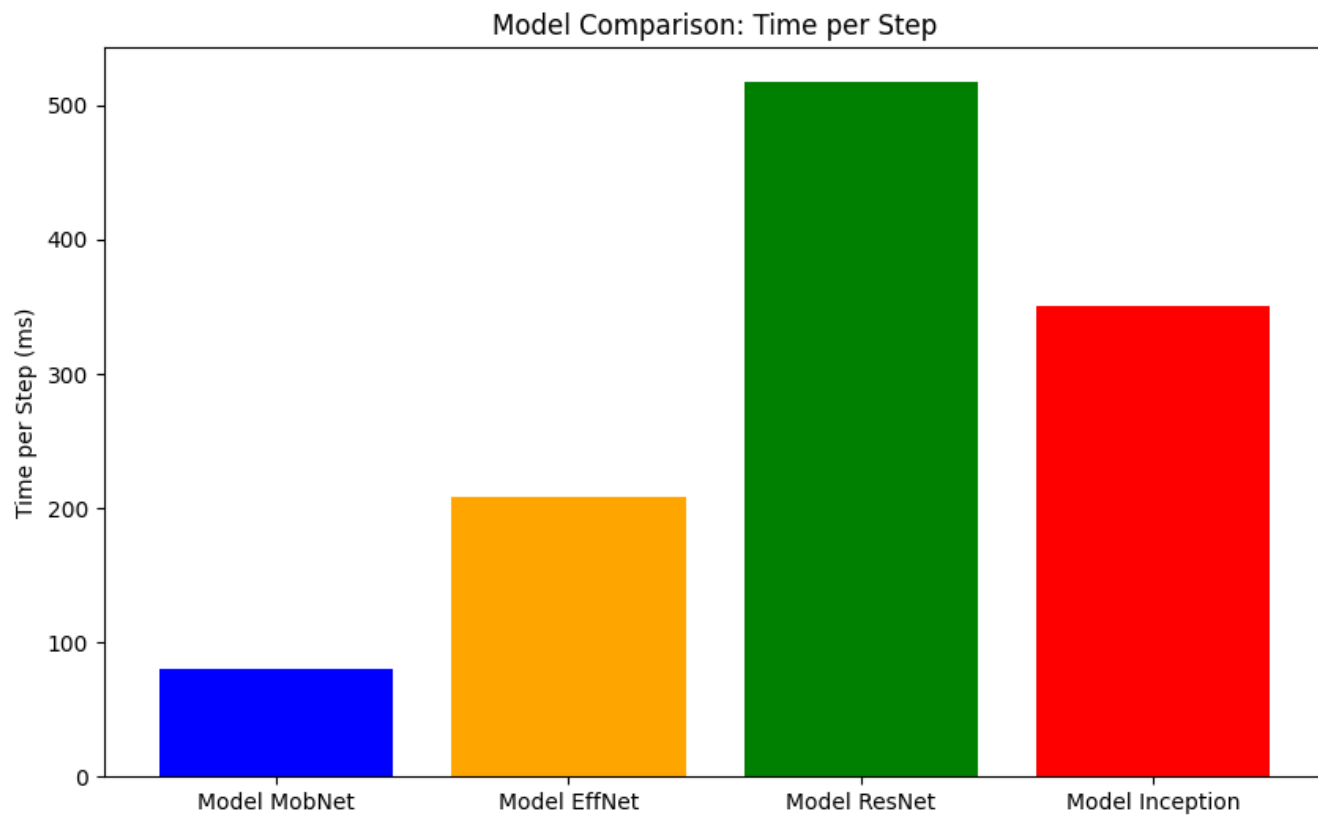
Jednou z možných cest vpřed by mohlo být zahrnutí detekce objektu a metody YOLO. Tato metoda by mohla efektivně pracovat s různými objekty na jedné fotografii a potenciálně poskytnout přesnější klasifikaci. Vzhledem k rychlosti YOLO by mohlo dojít ještě k dalšímu značnému zlepšení výkonu naší galerie.

REFERENCE

- [1] Emmanuel Seetam, Friday E. Onuodu. Object Detection Using Convolutional Neural Network Transfer Learning. 2022, www.semanticscholar.org/paper/Object-Detection-using-Convolutional-Neural-Network-Emmanuel-Onuodu/c61d07cdb15155c9b9fea4af42a3e917f3962a88.
- [2] Al-Shammery, Ali Abbas, et al. "Automatic Image Annotation System Using Deep Learning Method to Analyse Ambiguous Images." *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 11, no. 2, International University of Sarajevo, Apr. 2023, p. 176. <https://doi.org/10.21533/pen.v11i2.3517>.
- [3] Sharma, Tanya Goel Ujjwal. "Real-Time Image Processing Using Deep Learning With Opencv and Python." *Journal of Pharmaceutical Negative Results*, Medknow, Feb. 2023, pp. 1905–08. <https://doi.org/10.47750/pnr.2023.14.03.246>.
- [4] Wang, Renwu, et al. "Developing an Artificial Intelligence Framework for Online Destination Image Photos Identification." *Journal of Destination Marketing and Management*, vol. 18, Elsevier BV, Dec. 2020, p. 100512. <https://doi.org/10.1016/j.jdmm.2020.100512>.
- [5] Thakurdesai, Shalva, et al. "Smart Gallery Using Google Vision." *Social Science Research Network, RELX Group (Netherlands)*, Jan. 2021, <https://doi.org/10.2139/ssrn.3852766>.
- [6] Zhou, Bolei, et al. Places: A 10 Million Image Database for Scene Recognition — IEEE Journals & Magazine — IEEE Xplore. 2018, <https://doi.org/10.1109/TPAMI.2017.2723009>.
- [7] Manaswi, Navin Kumar. "Deep Learning with Applications Using Python." *Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras*, edited by Navin Kumar Manaswi, Apress, 2018, pp. 91–96, https://doi.org/10.1007/978-1-4842-3516-4_6.
- [8] Ketkar, Nikhil. "Deep Learning with Python." *Deep Learning with Python: A Hands-on Introduction*, edited by Nikhil Ketkar, Apress, 2017, pp. 63–78, https://doi.org/10.1007/978-1-4842-2766-4_5.
- [9] Teoh, Teik Toe, and Zheng Rong. "Artificial Intelligence with Python." *Artificial Intelligence with Python*, edited by Teik Toe Teoh and Zheng Rong, Springer, 2022, pp. 261–75, https://doi.org/10.1007/978-981-16-8615-3_16.



Obrázek 3: Výsledky srovnání přesností modelů
Zdroj: Autor



Obrázek 4: Výsledky srovnání rychlosti predikce modelů
Zdroj: Autor


```
--SCENE CATEGORIES for https://images.unsplash.com/photo-1607899917260-8075f6c6dfe8?ixid=M3wxMjA3fDB8MHxwaG90bylWYmVldHx8fGVufDB8cm9udGVudDQwMTB8bGljaXVCemRhdGE%3D&fm=jpg&q=80
museum-outdoor
tower
apartment_building-outdoor
plaza
temple-asia
```

```
1/1 [=====] - 0s 155ms/step
```

```
c --SCENE CATEGORIES for https://images.unsplash.com/photo-1517744014769-400000000000
rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8
skyscraper
office_building
downtown
tower
apartment_building/outdoor
```



Obrázek 5: Ukázka predikce modelů
Zdroj: Autor