

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

SYSTÉM PRO IDENTIFIKACI A KLASIFIKACI  
VYBRANÝCH OBJEKTŮ V LETECKÝCH MAPÁCH

Bakalářská práce

Autor práce: Jiří Kalina

Vedoucí práce: doc. Dr. Ing. Jan Voráček, CSc.

Jihlava 2024

# Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	<b>Jiří Kalina</b>
Studijní program:	Aplikovaná informatika
Obor:	Aplikovaná informatika
Garant studijního programu:	Ing. Lenka Kuklišová Pavelková, Ph.D.
Název práce:	<b>Systém pro identifikaci a klasifikaci vybraných objektů v leteckých mapách</b>
Vedoucí práce:	doc. Dr. Ing. Jan Voráček, CSc.
Cíl práce:	Cílem práce je návrh systému, který v leteckých mapách vyhledá a kategorizuje vybrané grafické objekty. Konkrétně půjde o nalezení budov a klasifikaci typů jejich střech. Splnění tohoto cíle předpokládá teoretické osvojení a praktickou implementaci následujících znalostních oblastí: a) práce s mapovými formáty, b) optimální segmentace a standardizace zdrojových obrazových dat, c) vytvoření vlastních tréninkových a klasifikačních množin, d) volba vhodného paradigmatu strojového učení a jeho strukturní i parametrická adaptace, e) ověření výkonnosti výsledného řešení, diskuse dosažených výsledků a formulace závěrečných doporučení. Výsledné softwarové řešení bakalářské práce bude prezentováno interaktivní formou s využitím specializovaných knihoven jazyka Python.

## Abstrakt

Současný rozvoj metod a nástrojů umělé inteligence ovlivnil i problematiku počítačového zpracování obrazové informace. Cílem práce je proto navrhnout systém pro identifikaci a klasifikaci typů střech z leteckých snímků pomocí technik hlubokého učení. Získané výsledky lze využít například pro účinnější rozmisťování alternativních zdrojů energie nebo při návrhu 3D modelů terénu. Navržený systém inovativním způsobem řeší dvě hlavní úlohy, kterými jsou automatické pořízení datové sady a její užití k účinnému natrénování zvolených modelů. Práce navazuje na dosavadní poznání v této oblasti a zaměřuje se na zjednodušení procesů vedoucích k výsledku. Stávající metody jsou náročné v části přípravy datové sady, která je buď závislá na ručním zpracování snímků, nebo sestává z komplikovaného expertního systému, jež ruční zpracování částečně doplňuje. Presentovaná metoda přináší nový přístup k řešení problému. Po odstranění nejnáročnější fáze, ručního odvozování obrazů střech, zůstane poslední nevyhnutelný manuální krok a tím je zařazení snímku do příslušné kategorie. Práce prokázala, že navrhovaná metoda automatizace procesu extrakce střechních obrazů za cenu nepřesností však nemá zásadní vliv na výkon testovaných modelů. Nejlepší dosažená celková přesnost testovaných modelů rovnocenná publikovaným výsledkům činí 84 % (Qian et al., 2023, s. 33), 85 % (Buyukdemircioglu, Can a Kocaman, 2021, s. 58), 86 % (Meng, Soleimani-Babakamali a Taciroglu, 2023, s. 15) a v případě námi navrhovaného řešení 89 %. Popsané řešení je tedy aplikovatelné na výše uvedenou problematiku, kde může snížit podíl lidské práce při zachování potřebné přesnosti. Zároveň nabízí i prostor k dalšímu výzkumu v oblasti automatického získávání obrazů střech a odstraňování nepřesností, inženýringu hyperparametrů nebo vývoje vlastních klasifikačních modelů.

## Klíčová slova

Hluboké učení; konvoluční sítě; přenosové učení; klasifikace střech; dálkový průzkum země; PyTorch; EfficientNetB1; ResNet 101; VGG-16.

## Abstract

The current development of artificial intelligence methods and tools has also influenced the issue of computer processing of image information. Therefore, the aim of this paper is to propose a system for identification and classification of roof types from aerial images using deep learning techniques. The obtained results can be used, for example, for more efficient deployment of alternative energy sources or in the design of 3D terrain models. The proposed system solves two main tasks in an innovative way, which are the automatic acquisition of the dataset and its use to efficiently train the selected models. The work builds on existing knowledge in this area and focuses on simplifying the processes leading to the result. Existing methods are challenging in the dataset preparation part, which either depends on manual image processing or consists of a complicated expert system that partially replaces the manual processing. The presented method brings a new approach to solve the problem. After removing the most challenging stage, the manual derivation of roof images, the last unavoidable manual step remains and that is the classification of the image into the appropriate category. The work demonstrated that the proposed method of automating the roof image extraction process at the cost of inaccuracies, however, does not have a significant impact on the performance of the tested models. In both cases, the best achieved overall accuracy of the tested models is equivalent to the published results of 84 % (Qian et al., 2023, p. 33), 86 % (Buyukdemircioglu, Can a Kocaman, 2021, p. 58), 86 % (Meng, Soleimani-Babakamali a Taciroglu, 2023, p. 15) and 89 % in the case of our proposed solution. Thus, the described solution is applicable to the above-mentioned problem where it can reduce the proportion of human work while maintaining the required accuracy. At the same time, it also offers scope for further research in the field of automatic roof image retrieval and inaccuracy removal, hyperparameter engineering or development of custom classification models.

## Keywords

Deep learning; convolutional networks; transfer learning; roof type classification; remote sensing; PyTorch; EfficientNetB1; ResNet 101; VGG-16.

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užití své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 21. června 2024

.....

podpis studenta

## Poděkování

*Na tomto místě bych rád vyjádřil vděčnost všem, kteří mi pomohli s vypracováním této bakalářské práce. Především bych rád poděkoval své manželce za neutuchající podporu, trpělivost a důvěru po celou dobu studia. Bez ní by se mi tato práce jen těžko podařila dokončit. Dále bych rád srdečně poděkoval panu doc. Dr. Ing. Janu Voráčkovi, CSc., vedoucímu mé bakalářské práce. Za jeho cenné rady, vstřícnost a ochotu věnovat mi nadstandardní čas i mimo konzultačních hodiny. Jeho odborné vedení a připomínky mi pomohly posunout tuto práci na mnohem vyšší úroveň.*

*Děkuji Vám oběma.*

# Obsah

<b>Seznam obrázků.....</b>	<b>8</b>
<b>Seznam tabulek .....</b>	<b>9</b>
<b>Seznam zkratk.....</b>	<b>10</b>
<b>Úvod .....</b>	<b>11</b>
<b>1 Teoretický úvod do problematiky.....</b>	<b>13</b>
1.1 Datové podklady.....	13
1.2 Hluboké učení.....	16
<b>2 Metody .....</b>	<b>18</b>
2.1 Datové zdroje a získání podkladů .....	18
2.2 Návrh experimentu.....	23
<b>3 Výsledky a diskuse experimentu .....</b>	<b>27</b>
3.1 Způsoby hodnocení modelu .....	27
3.2 Model EfficientNetB1 .....	29
3.3 Model ResNet-101.....	32
3.4 Model VGG-16 .....	37
3.5 Celkové vyhodnocení experimentu .....	41
<b>Závěr.....</b>	<b>44</b>
<b>Seznam použité literatury .....</b>	<b>46</b>

## Seznam obrázků

Obr. 1: Ukázka zvolených šesti kategorií.....	14
Obr. 2: Ukázka zvolených čtyř kategorií.....	15
Obr. 3: Ukázka zvolených šesti kategorií.....	15
Obr. 4: Ukázka WMS služby Ortofotomapa ČR připojené do softwaru QGIS .....	18
Obr. 5: Ukázka automatické linky v softwaru FME Workbench .....	19
Obr. 6: Ukázka segmentace střechy – zleva lokalizace budovy pomocí BBOX, uprostřed obraz budovy s polygonem RÚIAN (modrá) a buffer 2,5 m (červená), vpravo segmentovaná střecha budovy.....	20
Obr. 7: Ukázka zvolených typů střech.....	21
Obr. 8: Ukázka vrstev architektury VGG-16 .....	24
Obr. 9: Ukázka části zdrojového kódu pro transformaci obrazů .....	25
Obr. 10: Příklady ztráty modelu, zleva: podtrénování, přetrénování a konvergence.....	28
Obr. 11: Graf ztráty a přesnosti modelu EfficientNetB1 .....	30
Obr. 12: Confusion matrix modelu EfficientNetB1, vlevo 9 epoch s celkovou přesností modelu 78,91 %, vpravo 21 epoch s celkovou přesností modelu 81,84 % .....	31
Obr. 13: Grafy ztráty a přesnosti modelu ResNet-101.....	34
Obr. 14: Confusion matrix modelu ResNet-101, vlevo 9 epoch s celkovou přesností modelu 81,00 %, vpravo 21 epoch s celkovou přesností modelu 81,84 % .....	35
Obr. 15: Grafy ztráty a přesnosti modelu VGG-16 .....	38
Obr. 16: Confusion matrix modelu VGG-16, vlevo 9 epoch s celkovou přesností modelu 88,10 %, vpravo 21 epoch s celkovou přesností modelu 88,94 % .....	39
Obr. 17: Aktivační funkce, zleva ReLU, ReLU6 a Swish (v PyTorch jako SiLU).....	42

## Seznam tabulek

Tab. 1: Distribuce snímků do trénovací a validační sady .....	14
Tab. 2: Distribuce snímků do trénovací, validační a testovací sady .....	16
Tab. 3: Vybrané kategorie dle využití objektu RÚIAN .....	22
Tab. 4: Četnost jednotlivých typů střech v podmnožinách .....	23
Tab. 5: Vybrané příklady záměn klasifikace modelu EfficientNetB1 .....	32
Tab. 6: Vybrané příklady záměn klasifikace modelu ResNet-101 .....	36
Tab. 7: Vybrané příklady záměn klasifikace modelu VGG-16 .....	40
Tab. 8: Výsledky testovaných modelů .....	42
Tab. 9: Celkové porovnání napříč pracemi .....	43

## Seznam zkratek

BBOX	Bounding box
CNN	Convolutional neural network (konvoluční neuronová síť)
ČR	Česká republika
ČSN	Česká státní norma
ČSU	Český statistický úřad
ČÚZK	Český úřad zeměměřický a katastrální
GDAL	Geospatial data abstraction library
GIS	Geografický informační systém
GPU	Graphic processing unit
LoD2	Level of Detail 2
MO	Ministerstvo obrany
MZ	Ministerstvo zemědělství
OGS	Open Geospatial Consortium
PNG	Portable network graphics
QGIS	Quantum geographic information system
RGB	Red-green-blue (aditivní barevný model, červená-zelená-modrá)
RÚIAN	Registr územní identifikace, adres a nemovitostí
VFR	Výměnný formát RÚIAN
VŠPJ	Vysoká škola polytechnická Jihlava
WMS	Web map service

# Úvod

Jako analytik geografických informačních systémů (GIS) se stále častěji setkávám s komplexnějšími požadavky na GIS systém, který umožňuje efektivní analýzu a správu geografických dat. Tyto požadavky se týkají mnoha aspektů, včetně energetického plánování, rozpočtování a udržitelnosti. Zvláště v oblasti státní správy a byznysu rostou nároky na důkladnou analýzu geografických objektů a zdrojů.

Cílem bakalářské práce je reagovat na tyto rostoucí požadavky tím, že navrhnu a otestuji systém, který umožní identifikaci a klasifikaci vybraných objektů v leteckých mapách. Konkrétně se budeme zaměřovat na identifikaci budov a klasifikaci typů jejich střech. To má význam zvláště v souvislosti s rozhodováním týkajícím se solární energie, alokace rozpočtů a udržitelného rozvoje.

Velmi častým požadavkem soukromých subjektů je možnost v globálním měřítku strojově identifikovat objekty budov a klasifikovat typ jejich střech a zjištěné informace využít pro zvýšení konkurenceschopnosti. Nejčastěji jde o vytipování vhodného objektu pro umístění solární elektrárny nebo o automatický výpočet předběžných nákladů na pořízení nové střešní krytiny. Další velmi aktuální oblastí jsou 3D mapy zastavěných oblastí dle standardu OGC CityGML, kde velmi častým požadavkem je konstrukce tvaru střechy pro LoD2 (Biljecki et al., 2015, s. 27). Zde lze právě klasifikaci typů střech využít.

V současné době prožíváme boom v oblasti umělé inteligence, zejména v oblasti počítačového vidění a hlubokého učení. Tato technologická revoluce přináší nové možnosti pro analýzu a klasifikaci geografických dat, což zvyšuje tlak na implementaci těchto inovativních metod do GIS systémů.

Práce bude založena na analýze existujících postupů a implementaci vlastního softwarového řešení v jazyce Python. Výsledné řešení bude prezentováno interaktivní formou s využitím specializovaných knihoven pro analýzu dat a strojového učení. Je třeba zvládnout dvě hlavní úlohy, a to pořídit datovou sadu a s její pomocí úspěšně natrénovat a ověřit účinnost zvolených modelů. Jedná se o komplexní multioborovou záležitost, kde je potřeba znalostí různých mapových formátů a jejich zpracování a zároveň znalostí hlubokého učení.

Úlohu první, pořízení datové sady, lze zobecnit do tří kroků, kdy poslední je nevyhnutelný pro libovolnou metodu. Prvním krokem je získání snímků, na kterých se nacházejí obrazy budov. Zde se nabízí možnost využít bezpilotní prostředky, letecké snímkování nebo družice dálkového průzkumu Země. Každá metoda vyžaduje jiný druh expertízy. Druhým krokem, který je časově nejvíce náročný, je odvození střešních ploch ze snímků budov. Třetím a posledním krokem je zařazení obrazu odvozené střešní plochy do zvolené kategorie. Uvedené metody si popíšeme v teoretickém úvodu do problematiky. V metodické části pak představíme inovativní řešení, jež výrazně minimalizuje náročnost kroků jedna a dva.

Úloha druhá, natrénování zvolených modelů a ověření jejich účinnosti, je obvykle řešena sestavením série experimentů za použití přetrénovaných modelů. I my se budeme držet tohoto osvědčeného scénáře a použijeme techniku zvanou *transfer learning*, kterou taktéž popíšeme v teoretické a následně metodické části při návrhu vlastního experimentu.

Na závěr budeme prezentovat a diskutovat dosažené výsledky ve srovnání s uvedenými publikacemi a nastíníme další možnosti a směr vývoje prezentovaného řešení. Práce má zodpovědět tyto výzkumné otázky: Je datová sada RÚIAN vhodná pro lokalizaci budov a segmentaci střech? Lze automatizovaně pořídit datovou sadu z veřejně dostupných zdrojů? Mají nepřesnosti automatické metody v segmentaci střechy vliv na výkonnost modelu?

# 1 Teoretický úvod do problematiky

Hluboké učení, zejména prostřednictvím konvolučních sítí, přináší revoluční změnu ve zpracování informací. Na rozdíl od tradičního přístupu, kde je program explicitně definován s cílem dosáhnout požadovaného výsledku, konvoluční sítě se učí (vytvářejí program) na základě obrovského množství označených dat (výsledku). Tyto sítě identifikují a extrahují relevantní vzory a vlastnosti přímo ze vstupních dat, což znamená, že není nutné ručně definovat každý krok zpracování.

Tímto způsobem se konvoluční sítě adaptují na širokou škálu problémů od rozpoznávání obrazů až po překlad jazyka. Klíčovým prvkem úspěchu konvolučních sítí je však kvalita datové sady a správná příprava těchto dat. Bez vhodného a rozmanitého tréninkového souboru dat by konvoluční sítě neměla dostatečný základ pro naučení se relevantních vzorů, a to by mohlo vést k nízké úspěšnosti modelu. Proto je důkladná příprava datových sad klíčovým krokem při použití konvolučních sítí.

V dalších kapitolách vybudujeme minimální teoretický základ pro pochopení úlohy před nás postavené a představíme si publikace, které se problematikou klasifikace typů střech zabývají. Uvedené publikace nám budou základem pro vývoj vlastního řešení.

## 1.1 Datové podklady

Důkladně zachytit nuance prostředí je nezbytné pro úspěšnou klasifikaci střech, jelikož tyto charakteristiky jsou inherentní jednotlivým vzorkům. Přehlednutí těchto nuancí by mohlo vést k neúplnému zachycení informací a ztrátě potenciálně cenných poznatků, které jsou klíčové pro vytvoření optimálního reprezentativního vzorku. Cílem naší práce je získání takového optimálního vzorku dat.

Některá řešení se omezují jen na určitý typ zástavby a vyrobí specializovanou datovou sadu pro tyto účely. Takovým případem je klasifikace, která se zaměřuje na solitérní rodinné domy (Meng, Soleimani-Babakamali a Taciroglu, 2023). Sofistikovanější řešení, které se snaží o vytvoření optimálního reprezentativního vzorku dat, jež je schopné generalizovat napříč rozdílnými druhy zástavby, představuje Qian et al. (2023). Jedná se o velmi robustní řešení s automatickým získáváním obrazů s budovami i automatickým odvozením obrazu střechy. Standardní metodu s nejvyšší mírou manuální činnosti při přípravě datové sady nabízí Buyukdemircioglu, Can a Kocaman (2021).

### 1.1.1 Získání obrazů s budovami a odvození obrazů střech

V roce 2021, kdy byla tato práce zahájena, existovaly publikované studie zabývající se získáním obrazů budov výhradně manuálními metodami. Jedním z příkladů je práce Buyukdemirciogla z roku 2021. V uvedené práci je popsána metoda získání snímků pomocí leteckého snímkování tím, že se určí zájmová oblast s budovami, která se nasnímá. Získané obrazy je nutno manuálně projít a pomocí GIS nástroje segmentovat obraz střechy, což je časově náročný úkol.

V průběhu roku 2023 se objevily práce, které se snaží tyto kroky automatizovat. Dobrým příkladem je metoda (Meng, Soleimani-Babakamali a Taciroglu, 2023), která pomocí informací z realitní databáze v kombinaci s geolokační službou získává obrazy budov z mapové služby

Google Maps. Protože se soustřeďuje pouze na solitérní rodinné domy, rozhodli se autoři střechy nesegmentovat a obrazy použít i s rezidui okolí, jako jsou např. okolní stromy, chodníky či travnaté plochy.

Nejvíce komplexní systém, který je schopen získat obrazy budov a dále segmentovat hrany střech, představuje publikace Qiana et al. (2023), v níž autorský tým využívá kombinaci technik hlubokého učení a několika konvolučních architektur k získání obrazů s budovami a segmentací střech budov. Tento systém klasifikuje budovy do urbanistických kategorií a také klasifikuje příslušné typy střech v těchto kategoriích.

### 1.1.2 Zařazení obrazu střechy do kategorie

Proces zařazování obrazů střech do kategorií je nevyhnutelnou součástí klasifikačního systému a je společný všem metodám. Konvoluční síť potřebuje informaci o tom, na jakou kategorii se dívá, aby se mohla efektivně učit. V praxi to znamená, že každý snímek musí vidět člověk a zapsat informaci o daném typu střechy. Konvoluční sítě jsou známy tím, že potřebují datové sady v řádech tisíců až statisíců obrazů, jedná se tedy o časově náročnou úlohu.

Pro klasifikaci solitérních rodinných objektů za účelem modelování zranitelnosti vůči větru (Meng, Soleimani-Babakamali a Taciroglu, 2023) bylo zvoleno šest kategorií. Zleva: jednoduchá sedlová, křížová sedlová, komplexní křížová sedlová, jednoduchá valbová, křížová valbová a neznámá.



Obr. 1: Ukázka zvolených šesti kategorií

Zdroj: Qian et al. (2023, s. 21)

Tab. 1: Distribuce snímků do trénovací a validační sady

Roof type	Number of images	
	Training	Validation
Simple gable	1820	455
Simple cross-gable	1805	451
Complex cross-gable	1160	290
Simple hip	1037	259
Cross-hip	1823	456
Unknown	1822	456

Zdroj: Meng, Soleimani-Babakamali a Taciroglu (2023, s. 8)

Pro mapování rozsáhlých zastavěných oblastí a vzhledem k pixelovému rozlišení satelitního snímkování zvolili Qian et al. (2023) čtyři kategorie: střechu plochou, sedlovou, valbovou a komplexní. Zároveň byl každý snímek spolu s informací o typu střechy označen i kategorií urbanistické zástavby (bydlení, občanská, průmyslová).

Roof architectural type	Plan view	High-resolution satellite images		
Flat				
Gable				
Hip				
Complex				
	...	...	...	...

**Obr. 2: Ukázka zvolených čtyř kategorií**

*Zdroj: Qian et al. (2023, s. 21)*

Ve své práci rozděluje Buyukdemircioglu, Can a Kocaman (2021) získané snímky do šesti kategorií, jmenovitě zleva se jedná o tyto kategorie střech: komplexní, pyramidová, sedlová, půlvalbová, valbová a plochá. Účelem je generování zjednodušených modelů budov.

complex	pyramid	gable	half-hip	hip	flat

**Obr. 3: Ukázka zvolených šesti kategorií**

*Zdroj: Buyukdemircioglu, Can a Kocaman (2021, s. 57)*

**Tab. 2: Distribuce snímků do trénovací, validační a testovací sady**

<b>Roof Type</b>	<b>Training (72 %)</b>	<b>Validation (18 %)</b>	<b>Test (10 %)</b>	<b>Total</b>
Complex	1620	405	225	2250
Flat	1260	315	175	1750
Gabble	1260	315	175	1750
Hip	1260	315	175	1750
Pyramid	1080	270	150	1500
Halfhip	720	180	100	1000

*Zdroj: Buyukdemircioglu, Can a Kocaman (2021, s. 56)*

## 1.2 Hluboké učení

Architektury hlubokých konvolučních sítí (CNN) používají k extrakci vzorů ze vstupních dat tzv. filtry *kernels*, v překladu jádra nebo konvoluční jádra (Howard a Guggen, 2020, s. 404). Tyto filtry nebo také váhy jsou aplikovány na lokální oblasti vstupního obrazu, čímž umožňují identifikaci různých vizuálních prvků. Princip extrakce vzorů spočívá v adaptivním nastavení vah filtrů tak, aby co nejlépe odpovídaly charakteristikám vstupních dat. V geografickém kontextu mohou tyto filtry identifikovat charakteristiky, jako jsou hrany, valby, křížení, což je důležité pro přesnou klasifikaci objektů.

Při vytváření a trénování konvolučních neuronových sítí je vhodné rozdělit dostupná data na trénovací, validační a testovací sady. Jedná se o standardní postup, jak ověřit výkonnost modelu a zajistit jeho schopnost generalizace na nová, dosud neviděná data (Goodfellow, Bengio a Courville, 2016, s. 110). Trénovací data slouží k aktualizaci vah filtrů tak, aby co nejlépe odpovídaly požadovaným vlastnostem vstupních dat.

Validace probíhá na separátní sadě dat, která není použita během trénování a slouží k optimalizaci hyperparametrů sítě, jako jsou například rychlost učení či velikost filtrů. Nakonec jsou testovací data použita k posouzení výkonnosti modelu na nezávislých datech a poskytují objektivní měřítko jeho schopnosti generalizace. Tento postup umožňuje vytvoření spolehlivého a robustního modelu, který je schopen efektivně klasifikovat střechy a zachytit jejich různorodé charakteristiky.

### 1.2.1 Transfer learning

Trénování konvolučních neuronových sítí na omezených datových sadách, jako je ta naše, může být náročné a časově neefektivní. Existuje zde riziko, že model se nebude schopen naučit dostatečné množství vzorů z omezené datové sady a nebude schopen dostatečné klasifikace. Zaměříme se proto na metodu zvanou *transfer learning*.

Transfer learning je technika, která využívá znalosti naučeného modelu na jedné úloze k urychlení učení na jiné, obdobné úloze (Russell a Norvig, 2021, s. 832). V kontextu konvolučních sítí to znamená použití předtrénovaných modelů na rozsáhlých datových sadách, jako je například ImageNet (Chollet, 2019, s. 36), a aplikaci těchto znalostí na novou úlohu, například klasifikaci typů střech. Jedná se o široce používanou techniku, neboť jak uvádí Fridman (2019), dnes je již výzkum a vývoj v oblasti konvolučních sítí ztrátou času a pozornost by měla být věnována oblasti transfer learningu.

Svůj experiment staví Meng, Soleimani-Babakamali a Taciroglu (2023) právě na této technice a použití modelu VGG-19, jenž je rozšířením hloubky architektury VGG-16 (Simonyan a Zisserman, 2014). Oproti tomu Buyukdemircioglu, Can a Kocaman (2021) používá několik modelů – VGG-16, EfficientNetB4, Resnet-50 a jeden vlastní model, který trénuje od nuly, tedy bez použití transfer learningu.

Výjimkou je pak Qian et al. (2023), kteří používají celou škálu různých metod hlubokého učení. Jejich komplexní víceúčelový systém je navržen tak, aby se vypořádal s omezeními jednoúlohových řešení popsaných výše. Jedním z výstupů pak je i klasifikace typu střechy, do jejíž predikce je zahrnuto více faktorů. Jedním z nich je i typ urbanistické zástavby, ve kterém se střecha nachází (pro bydlení, průmyslová, občanská).

V našem řešení se budeme věnovat pouze transfer learningu, neboť je sám o sobě velmi výkonný a ve většině případů předčí modely trénované od nuly. Jedním z již uvedených důvodů je malé množství snímků a dalším důležitým faktorem jsou i dostupné hardwarové prostředky. Učení hlubokých modelů, jako např. ResNet-101, od nuly je časově a hardwarově mnohem náročnější než použití téhož modelu s již naučenými vzory.

## 2 Metody

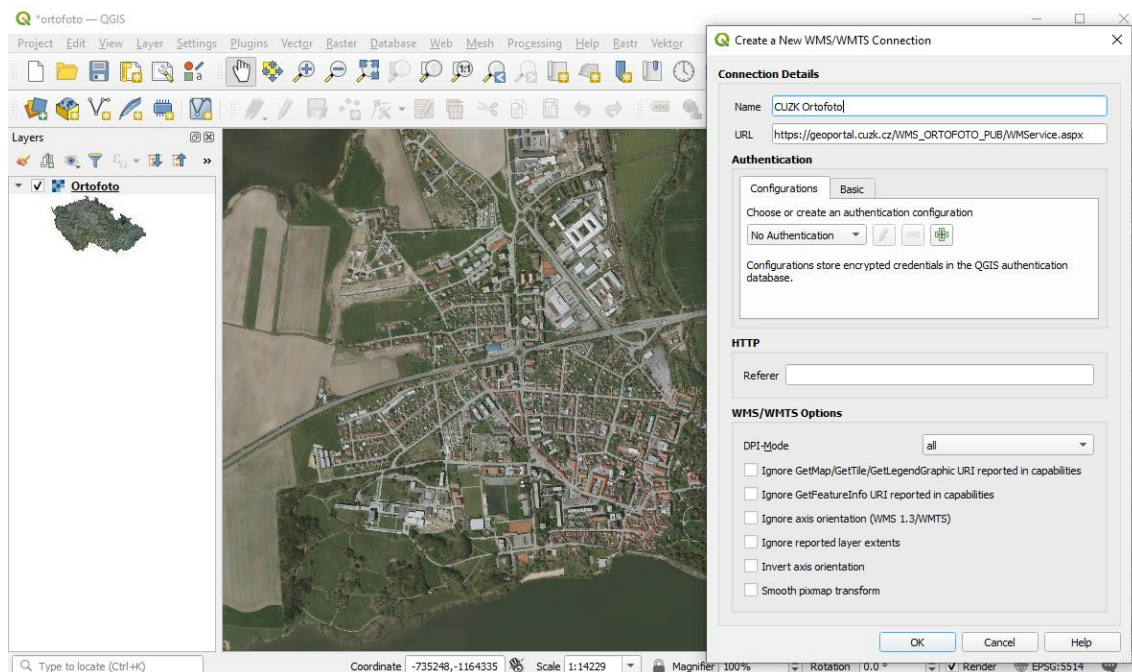
Kapitola s návrhem vlastního řešení je rozdělena do dvou hlavních tematických bloků. Prvním blokem je samotná příprava datové sady, popis dostupných zdrojových dat a způsob získání snímků. V druhém bloku je pak popsáno nastavení experimentu a potřebného vývojového prostředí s hardwarovými nároky.

### 2.1 Datové zdroje a získání podkladů

Je nutno podotknout, že navrhovaná metoda má své lokální specifikum a tím je její využitelnost omezená na území ČR. Metoda počítá s využitím veřejně dostupných geografických zdrojů, a to zejména veřejného registru státní správy RÚIAN (ČÚZK, 2023) a kartografického díla Ortofotomapa České republiky (910, 2024). Variace tohoto přístupu za použití jiných podkladů (Meng, Soleimani-Babakamali a Taciroglu, 2023, s. 4) je popsána v kapitole 1.1.1 teoretické části.

#### 2.1.1 Ortofotomapa ČR

Jedná se o na našem území již dobře etablovaný veřejný mapový podklad, který vzniká ve spolupráci MO, MZ a ČÚZK. Při zpracování a shromažďování leteckých snímků je pro každé místo v ČR dostupná alespoň jedna stereodvojice (Hron, 2019, s. 16) a mnoho dalších metadat a dat ze senzorů. Veřejnosti je pak dostupná bezešvá mozaika s velikostí pixelu 12,5 cm jako obraz zemského povrchu, který prošel zpracováním nasbíraných snímků ve formě WMS služby na adrese: <https://ags.cuzk.cz/arcgis1/services/ORTOFOTO/MapServer/WMSServer>.



Obr. 4: Ukázka WMS služby Ortofotomapa ČR připojené do softwaru QGIS

Zdroj: vlastní zpracování

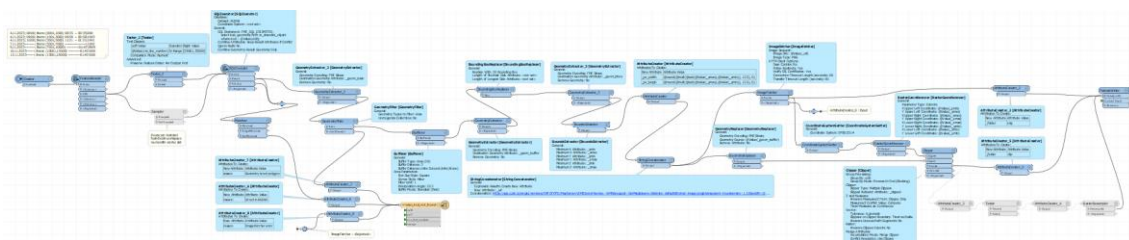
### 2.1.2 Registr územní identifikace, adres a nemovitostí (RÚIAN)

Jedná se o základní registr veřejné správy, který obsahuje adresy a údaje o pozemcích a budovách (ČÚZK, 2023). Geometrické informace o budovách lze z této datové sady získat dle specifikace výměnného formátu VFR (ČÚZK, 2023). Ve vrstvě *stavební objekty* se nachází většina stavebních objektů na území ČR. V současné době eviduje RÚIAN více než čtyři miliony stavebních objektů. Navrhované řešení předpokládá schopnost práce s VFR. K identifikaci stavebních objektů je použita právě tato veřejná sada, která poměrně přesně vytyčuje polygon stavebního objektu a tím ho i lokalizuje.

Pro tyto účely je potřeba mít naimportovanou sadu VFR pro celé území ČR do databáze, která podporuje zobrazování geografických dat. Jak uvádí Klíma (2015), existuje vícero *open source* možností, např. použití pluginu do GIS softwaru QGIS nebo použití GDAL. Popis metod pro získání tohoto podkladu není předmětem práce.

### 2.1.3 Automatizace kroku získání obrazu budov a odvození obrazu střechy

Obrazy budov jsou získány z veřejné WMS služby Ortofotomapa ČR, a to automatickou linkou typu PNG RGBA s pixelovým rozlišením 12,5 cm/px. Jedná se tedy o snímky s vysokým rozlišením, srovnatelné s kvalitou bezpilotních snímků 10 cm/px (dle Buyukdemircioglu, Can a Kocaman, 2021, s. 56). Snímky jsou oříznuty dle polygonu stavebního objektu RÚIAN, zvětšeného o buffer 2,5 m. Takto získáme poměrně přesné hranice střechy bez použití složitých metod jejich určení nebo ručního odvozování. Obraz má jen některé reziduální informace z okolí.



**Obr. 5: Ukázka automatické linky v softwaru FME Workbench**

*Zdroj: vlastní zpracování*

Automatizace získání obrazů budov a následná segmentace střechy je zpracována v softwaru FME Form (Safe Software, 2024). Procesní linka je napojena na databázi s importovanou datovou sadou RÚIAN. Z příslušné vrstvy náhodně vybere stavební objekty z kategorií popsaných v kapitole 2.1.5. Ty odpovídají urbanistickému typu zástavba pro bydlení a občanská zástavba, průmyslové objekty byly záměrně vynechány.

Na základě polygonu stavebního objektu je spočítán *bounding box*, dále jen BBOX. Takto získané souřadnice BBOXu slouží k lokalizaci budovy a sestavení dotazu na WMS službu Ortofotomapa ČR, ze které je následně stažen obraz budovy s jejím okolím.



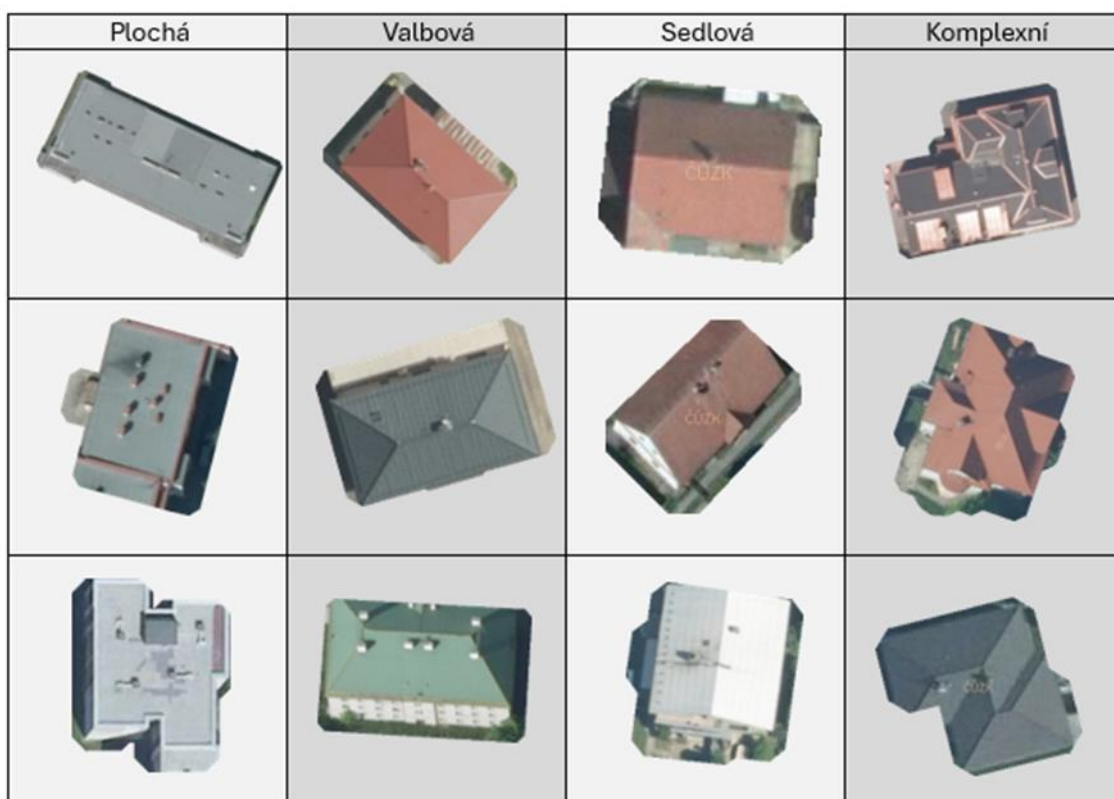
**Obr. 6: Ukázka segmentace střechy – zleva lokalizace budovy pomocí BBOX, uprostřed obraz budovy s polygonem RÚIAN (modrá) a buffer 2,5 m (červená), vpravo segmentovaná střecha budovy**  
*Zdroj: vlastní zpracování*

Polygon stavebního objektu je zvětšen o 2,5 m, jedná se o hodnotu zjištěnou experimentováním. Velikost bufferu je dostatečná, aby kompenzovala nepřesnosti při kombinaci datové sady RÚIAN s WMS ČR, kdy WMS není snímána ortogonálně. Uvedený buffer tento nedostatek kompenzuje za cenu méně přesné segmentace střech a okolních reziduí.

#### 2.1.4 Zařazení snímků do kategorií

Cílem vývoje klasifikátoru střech je usnadnit identifikaci objektů na území České republiky, které jsou vhodné pro umístění solárních panelů. Klasifikace slouží k získání základní informace o tvaru střechy, který ovlivňuje způsob instalace panelů a jejich optimální počet. Na základě znalosti tvaru střechy z klasifikátoru a polygonu RÚIAN je možné zkonstruovat přibližný 3D model střechy a určit její orientaci ke světovým stranám. V kombinaci uvedených datových sad je informace z klasifikátoru o typu střechy poslední informací, která umožní v globálním měřítku analyzovat vhodnost střech pro instalaci FTV panelů.

Základní rozdělení střech pro klasifikátor by bylo na ploché a sklonité. Toto rozdělení je relevantní, jelikož instalace panelů se v těchto dvou případech výrazně liší. Dále se sklonité střechy dělí na sedlové, valbové a komplexní, čímž se celkový počet kategorií zvyšuje na čtyři. Toto rozdělení zohledňuje vliv sklonu a tvaru střechy na optimální umístění panelů z hlediska orientace ke světovým stranám. Z hlediska instalace panelů na sklonitých střechách je nevhodnější sedlová střecha, dále valbová a nakonec komplexní.



**Obr. 7: Ukázka zvolených typů střech**

*Zdroj: vlastní zpracování*

Proces zařazování obrazů střech do kategorií je nezbytnou součástí klasifikačního systému založeného na konvolučních sítích a z hlediska našeho systému se jedná o jediný manuální krok v přípravě datové sady. Existuje více metod, jak snímku přiřadit informaci o příslušné kategorii. V našem případě použijeme metodu přesunutí snímku do patřičné souborové složky, abychom v implementaci mohli použít metodu knihovny PyTorch *DataLoader*, která právě počítá s označením vzorků pomocí rozdělení do pojmenovaných souborových složek.

### 2.1.5 Charakteristika získané datové sady

Datová sada pro tuto práci obsahuje snímky střech stavebních objektů náhodně vybraných z celého území České republiky. Objekty byly vybrány na základě specifických kategorií využití dle číselníku datové sady RÚIAN s názvem CE\_ZPUSOB\_VYUZITI\_OBJEKTU.

Tyto číselníkové kategorie zhruba odpovídají urbanistickým kategoriím *bydlení* (Rozmanová et al., 2013), *občanská* (Šindlerová et al., 2020) a *průmyslová* (Rozmanová et al., 2016). Toto rozdělení dále zohledňuje práci Qiana et al. (2023), jehož systém kromě typu střechy klasifikuje i druh zástavby, ve které se objekt nachází. Tímto způsobem doplňuje informaci, kterou lze na našem území získat z územních plánů.

Druh zástavby je klíčový z hlediska územního plánování, které definuje plochy s různým způsobem využití (Vyhláška č. 501/2006 Sb.). Tyto plochy následně vymezují, jaký typ objektu lze na daném území umístit, což nepřímo ovlivňuje i typ střechy daného objektu.

Územní plánování v ČR tak ovlivňuje typy budov umísťovaných na našem území a s nimi i typy jejich střech. Tyto informace se pak v rámci realizace staveb promítají do datové sady RÚIAN prostřednictvím datového modelu do příslušných atributů.

Následný výběr objektů pro tuto práci se proto zaměřuje na objekty z číselníku RÚIAN *způsob využití objektu*, jenž volně sdružuje urbanistické kategorie *bydlení* a *občanská*. Názvy vybraných kategorií jsou uvedeny v tabulce č. 2. Kompletní číselník je dostupný na webu Českého úřadu zeměměřického a katastrálního: <https://www.cuzk.cz/ruian/Poskytovani-udaju-ISUI-RUIAN-VDP/Ciselniky-ISUI/Atributy-stavebniho-objektu/Zpusoby-vyuziti-stavebniho-objektu.aspx>.

Z datové sady byly záměrně vyjmuty objekty kategorie *průmyslové*. V tomto ohledu existuje prostor pro rozšíření datové sady o tento typ objektů a pro případný další výzkum zaměřený na to, jakým způsobem se informace z územních plánů přenášejí do datové sady RÚIAN a jaké další informace z této sady lze využít ke zpřesňování klasifikace střech.

**Tab. 3: Vybrané kategorie dle využití objektu RÚIAN**

<i>Kód</i>	<i>Název</i>
3	objekt k bydlení
6	bytový dům
7	rodinný dům
8	stavba pro rodinnou rekreaci
10	stavba pro obchod
11	stavba ubytovacího zařízení
15	stavba občanského vybavení

*Zdroj: vlastní zpracování*

Získané obrazy střech jsou ve formátu PNG se třemi barevnými kanály RGB. Datová sada obsahuje přibližně 3000 snímků a byla rozdělena na tři podmnožiny: trénovací, validační a testovací v poměru 70 : 15 : 15. Toto rozdělení má několik důležitých důvodů:

- **Trénovací sada (70 %):** Tato sada slouží k trénování modelu strojového učení. Model se učí rozpoznávat různé typy střech na základě obrazů v této sadě.
- **Validační sada (15 %):** Tato sada se používá k ladění hyperparametrů modelu. Hyperparametry jsou nastavení modelu, která ovlivňují jeho výkonnost. Validační sada umožňuje optimalizovat hyperparametry tak, aby model dosahoval co nejlepšího výkonu na nezávislých datech.
- **Testovací sada (15 %):** Tato sada se používá k hodnocení konečného výkonu modelu. Testovací sada by měla obsahovat data, která model nikdy předtím neviděl, jak je tomu i v našem případě. Výsledky na testovací sadě poskytují realistický obraz o tom, jak dobře bude model fungovat v praxi.

Rozdělení datové sady na trénovací, validační a testovací sadu se obvykle provádí náhodně. To znamená, že obrazy střech jsou náhodně vybrány a přiděleny do jedné z podmnožin. Existuje několik různých metod náhodného rozdělení, například rozdělení v poměru 70 : 15 : 15, které bylo použito v tomto případě.

**Tab. 4: Četnost jednotlivých typů střech v podmnožinách**

	<i>Plochá</i>	<i>Valbová</i>	<i>Sedlová</i>	<i>Komplexní</i>
Trénovací	749	515	780	186
Validační	165	94	182	38
Testovací	165	94	182	38
Celkem v kateg.	1079	703	1144	262
Celkem	3188			

*Zdroj: vlastní zpracování*

## 2.2 Návrh experimentu

Experiment je sestaven tak, aby se v některých parametrech shodoval s publikovanými pracemi a v závěru jsme mohli výkonnosti modelů nad datovými sadami alespoň přibližně porovnávat.

### 2.2.1 Vývojové prostředí

Pro vývoj a běh experimentu jsem zvolil programovací jazyk Python s knihovnou pro strojové učení PyTorch (Pytorch Foundation, 2023). Grafické výstupy jsou realizovány pomocí knihovny Matplotlib (Hunter, 2007). Pro sledování experimentu a výkonnosti jednotlivých nastavení hyperparametrů jsem zvolil interaktivní vizualizační nástroj TensorBoard (TensorFlow, 2023).

Pro prototypování a prvotní ověření hypotézy bylo zvoleno cloudové prostředí Google Colab (Google, 2024), které má všechny potřebné knihovny a nástroje pro práci s konvolučními modely. Jedná se o interaktivní rozhraní přímo v prohlížeči, ze kterého lze za poplatek využívat výkonného hardwaru, v tomto případě GPU.

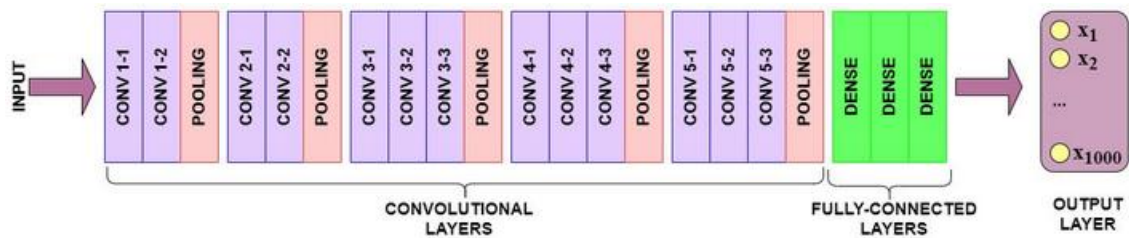
Po prvním úspěšném testování byl vývoj přesunut na vlastní PC pro extenzivní experimentování s jednotlivými modely. Běhové prostředí sestává z linuxového stroje s OS Ubuntu 23.10. (Smith, 2023) s GPU Nvidia RTX1080 8 GB. Vývojové prostředí je sestavené z kontejneru Docker (Merkel, 2014) s využitím přizpůsobeného docker image pytorch. Pro paralelní využití GPU bylo využito knihoven CUDA a Nvidia Container Toolkit.

### 2.2.2 Modularita

Experiment je postaven modulárně a je rozdělen do několika komponent, které lze nezávisle na sobě modifikovat. Tím lze experiment opakovat např. s jinak popsányými daty nebo modifikovat metriku modelu či grafické výstupy. Stačí modul, který se stará o jednotlivé části sestavení experimentu, modifikovat nebo vyměnit.

Příprava dat: **data\_setup.py** transformuje obrazy střech na požadovaný rozměr 224 x 224, převádí je na tenzory a normalizuje barvy. Dále dle zadání parametru random seed náhodně rozděljuje data do trénovací, validační a testovací sady v poměru 70 : 15 : 15 %. Výstupem je dataloader s tenzory, které přijímá zvolený model.

Příprava modelu: **model\_builder.py** je modul pro zvolené architektury, připravené pro transfer learning. Vybrané předtrénované architektury konvolučních modelů jsou připravené pro metodu transfer learning. Jedná se o tyto tři architektury: EfficientNetB1, ResNet-101 a VGG-16. Konvoluční vrstvy (convolutional layers) modelů jsou zmrazené a hustě propojené vrstvy (fully-connected layers) jsou nahrazené vrstvami, které budeme učit klasifikovat typy střech do čtyř kategorií.



Obr. 8: Ukázka vrstev architektury VGG-16

Zdroj: McDermont (2024)

Trénování modelu: **engine.py** je modul, který se stará o trénování zvolených architektur, a tedy o učení modelu. V tomto modulu je i metoda SummaryWriter, která se stará o zaznamenávání všech experimentů. Sledované experimenty se pak dají vizualizovat pomocí nástroje TensorBoard (TensorFlow, 2023). Tento nástroj také shromažďuje špatně klasifikované obrazy střech v jednotlivých experimentech, které jsou pak ukázány v příslušných kapitolách u vyhodnocení jednotlivých modelů

Testování modelu: **eval\_model.py** je modul nasazený po natrénování a validování jednotlivých architektur, aby na testovací sadě vyhodnotil výkonnost jednotlivých modelů. Sleduje všechny dostupné metriky: confusion matrix, celkovou přesnost, přesnost pro jednotlivé třídy, precision, recall a F1 skóre.

Grafické výstupy modelu **plot\_data.py** slouží ke generování grafů a statistik pro jednotlivé modely. Pro každý model generuje statistiku metrik popsaných v modulu eval\_model.py a dále graf confusion matrix a grafy trénovací, resp. validační přesnosti a ztráty každého modelu.

Hlavní funkce pro spuštění navrženého experimentu **main\_model\_4.py** trénuje zvolené modely, každý po dobu 9 a 21 epoch. Lze ji spouštět pro libovolný počet náhodných zamíchání dat – *random seed*. Hlavní funkce používá všechny výše popsané komponenty a díky tomu lze experiment a jeho výstupy snadno modifikovat.

### 2.2.3 Příprava datové sady pro parametry modelu

Pro účely naší práce plánujeme využít obrazy o rozměrech  $224 \times 224$  pixelů, což je záměrná volba s ohledem na použití stávajících architektur, které pracují s touto velikostí. Tyto obrazy budou ve formátu PNG a obsahují informace o geografických objektech ve formě RGB kanálů. Pro implementaci v PyTorch je důležité, aby vstupní data byla vhodně upravena podle požadovaných parametrů modelu. RGB hodnoty barev jsou obvykle v rozmezí 0 až 255, a proto je nezbytné provést jejich normalizaci pro práci s váhami neuronové sítě. Standardní postup zahrnuje převod hodnot na desetinná čísla v rozsahu 0 až 1. Tento proces spočívá v dělení hodnot každého kanálu maximální hodnotou (255). Funkcionalitu zajišťuje modul data\_setup.py.

```

14  data_transform = transforms.Compose([
15      transforms.Resize((224, 224)),
16      ConvertToRGB(),
17      transforms.ToTensor(),
18      transforms.Normalize([0.485, 0.456, 0.406],[0.229, 0.224, 0.225])
19  ])

```

**Obr. 9: Ukázka části zdrojového kódu pro transformaci obrazů**

*Zdroj: vlastní zpracování*

Při vytváření a trénování konvolučních neuronových sítí je důležité rozdělit dostupná data na trénovací, validační a testovací sady. Tento postup umožňuje trénování modelu na jedné sadě dat, ověřování jeho výkonu na validační sadě a testování na zcela nezávislé testovací sadě. Tím se zajišťuje, že model má schopnost generalizace na nová, neviděná data, aniž by byl příliš specializovaný na trénovací sadu.

Pro zajištění opakovatelnosti experimentů a výsledků je doporučeno použít náhodný seed při rozdělení dat. Tím zajistíme, že data budou vždy rozdělena do trénovací, validační a testovací sady stejným způsobem při každém běhu kódu, což je klíčové pro reprodukovatelnost výsledků a porovnání různých experimentů. Náš experiment pracuje s rozdělením 70 % trénovací data, 15 % validační data, 15 % testovací data a hodnotou seedu 13. Při volbě jiné hodnoty seedu, tedy jiného náhodného rozdělení nebo jiného poměru rozdělení, bude model dávat jiné výsledky. Toto nastavení lze změnit v modulu `data_setup.py` a jedná se o standardní náhodné vzorkování (Kuhn a Johnson, 2013, s. 68).

#### 2.2.4 Nastavení hyperparametrů

Při trénování modelu jsem musel optimalizovat řadu hyperparametrů, abych dosáhl co nejlepších výsledků. Níže popisuji nastavení a zdůvodnění pro každý z důležitých hyperparametrů.

**Počet epoch:** Trénoval jsem model na 9 a 21 epoch. Obě tato nastavení se ukázala jako dostatečná a model v obou případech konvergoval. Počet epoch je důležité nastavit tak, aby model stihl konvergovat a dosáhnout optimálního výkonu.

**Dávka:** Díky transfer learningu a zmrazení vrstev vyžadujících gradient jsem mohl nastavit vysokou hodnotu dávky na `BATCH_SIZE=128`. Kdybych model učil od nuly, paměť by mi nepostačila a musel bych použít dávku `BATCH_SIZE=8`, čímž by se trénování i na GPU značně prodloužilo. Velikost dávky ovlivňuje rychlost a stabilitu trénování. Větší dávky vedou k rychlejšímu trénování, ale mohou také vést k nestabilitě. Menší dávky vedou k pomalejšímu trénování, ale jsou obvykle stabilnější. Optimální velikost dávky závisí na konkrétním modelu a datové sadě.

**Rychlost učení:** Použil jsem standardní rychlost učení s hodnotou 0,001 (Bengio, 2012, s. 17). Rychlost učení je důležité nastavit tak, aby model konvergoval v rozumném čase a dosáhl optimálního výkonu. Pokud je rychlost učení nastavena příliš vysoko, model může přeskočit optimum a nikdy nekonvergovat. Pokud je rychlost učení nastavena příliš nízko, trénování bude trvat velmi dlouho (Goodfellow, Bengio a Courville, 2016).

Dropout: V tomto modelu jsem dropout nepoužil. Dropout (Chollet, 2019, s. 109) je užitečná technika pro velké modely s velkým počtem parametrů. V mém případě jsem použil relativně malý model a transfer learning, takže dropout nebyl nutný.

Hodnota seed: Testoval jsem model s 3 různými seedy (13, 23, 42) a nejlépe si vedl seed 13. Výsledky s tímto seedem budu prezentovat v práci. Hodnota seed je důležitá pro zajištění reprodukovatelnosti výsledků. Pokud by se hodnota seed změnila, model by mohl produkovat různé výsledky, i když by se hyperparametry a trénovací data nezměnily.

Optimalizační algoritmus: Použil jsem optimalizační algoritmus ADAM (Kingma a Ba, 2017). ADAM je univerzální optimalizační algoritmus, který se osvědčil v mnoha různých úlohách strojového učení. Je spolehlivý a efektivní a obvykle představuje dobrou volbu pro úlohy strojového učení.

Cost funkce: Jako cost funkci jsem použil Cross-Entropy (Mao et al., 2023). Cross-Entropy je běžná cost funkce pro úlohy klasifikace. Je vhodná pro modely s více třídami jako v mém případě.

Je důležité poznamenat, že optimální nastavení hyperparametrů se může lišit v závislosti na konkrétním modelu, datové sadě a úkolu. Je důležité experimentovat s různými nastaveními a sledovat výkon modelu na validační sadě, aby byla nalezena nejlepší konfigurace.

## 3 Výsledky a diskuse experimentu

Následující kapitola se zabývá výsledky experimentu, tak jak byl definován v kapitole 2. Pro shrnutí uvádíme, že jsme testovali tři známé modely, a to EfficientNetB1, ResNet-101 a VGG-16. Každý model byl testován na stejné datové sadě s rozdělením 70 % trénovací data, 15 % validační data a 15 % testovací data (modelem neviděná data). Rozdělení obrazů střech bylo náhodným seedem s hodnotou 13 kvůli replikovatelnosti experimentu. Každý model byl trénován po 9 a 21 epoch.

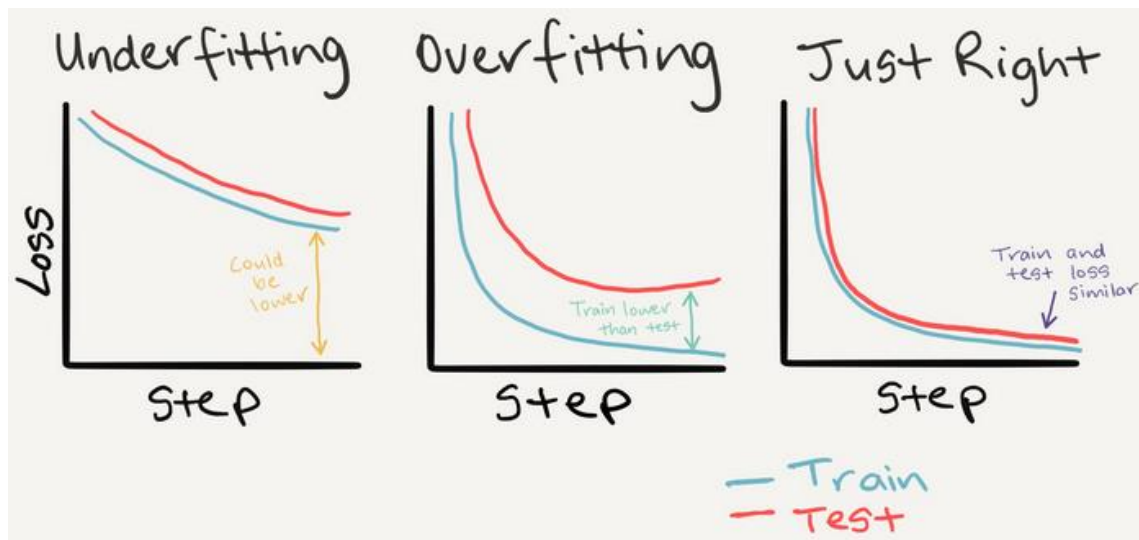
### 3.1 Způsoby hodnocení modelu

Metrikou pro hodnocení modelu nám bude jeho celková přesnost, dále to budou křivky ztráty (*loss*) a přesnosti (*accuracy*) a posledním měřítkem bude *confusion matrix* a další metriky odvozené z *confusion matrix*, které si charakterizujeme níže.

#### 3.1.1 Metrika přesnost a ztráta modelu

Z těchto křivek můžeme vyčíst několik klíčových informací o chování modelu:

- **Konvergence:** Ideální situací je, když křivka ztráty klesá a křivka přesnosti stoupá během tréninku. To naznačuje, že model se učí a stále zlepšuje svou schopnost predikce.
- **Přetrénování, *overfitting*** (Kuhn a Johnson, 2013, s. 62): Pokud se křivka ztráty pro trénovací data stále snižuje, zatímco křivka ztráty pro validační data začne stoupat, může to být známka přetrénování modelu. To znamená, že model se příliš naučil trénovací data a nemusí dobře generalizovat na nová data.
- **Podtrénování, *underfitting*** (Chollet, 2019, s. 106): Pokud jsou obě křivky, ztráty i přesnosti, stabilní nebo se jen mírně mění, model se pravděpodobně nedokázal naučit dostatečně dobře ani trénovací data. To může být známkou podtrénování, kdy je model příliš jednoduchý na to, aby efektivně modeloval data.
- **Optimální epocha:** Optimalizace modelu často zahrnuje vybrání epochy, ve které dosáhne nejlepšího kompromisu mezi ztrátou a přesností na validačních datech. Toto rozhodnutí může být založeno na průběhu křivek ztráty a přesnosti během tréninku (Goodfellow, Bengio a Courville, 2016, s. 426).



Obr. 10: Příklady ztráty modelu, zleva: podtrénování, přetrénování a konvergence

Zdroj: [learnpytorch.io](https://learnpytorch.io)

Vyčištění těchto křivek může zahrnovat různé postupy, jako jsou například optimalizace hyperparametrů (např. změna learning rate), rozšíření dat nebo změna architektury modelu. Cílem je dosáhnout stabilního tréninku modelu a minimalizovat případné přetrénování nebo podtrénování.

### 3.1.2 Metrika confusion matrix (matice záměny)

Jedná se o užitečný nástroj pro vizualizaci výkonu klasifikačních modelů, včetně konvolučních neuronových sítí. Tato matice (Pawan\_Dubey a arvindpdmn, 2019) organizuje predikce modelu a skutečné třídy do tabulky, která umožňuje snadnou interpretaci jeho výkonu.

- True Positives (TP): Počet případů, kdy model správně klasifikoval instanci do třídy, která je skutečně pozitivní.
- True Negatives (TN): Počet případů, kdy model správně klasifikoval instanci do třídy, která je skutečně negativní.
- False Positives (FP): Počet případů, kdy model mylně klasifikoval instanci do třídy, která je ve skutečnosti negativní, ale model ji klasifikoval jako pozitivní.
- False Negatives (FN): Počet případů, kdy model mylně klasifikoval instanci do třídy, která je ve skutečnosti pozitivní, ale model ji klasifikoval jako negativní.

Na základě těchto čtyř oblastí lze odvodit různé metriky, které poskytují informace o výkonu modelu:

- Přesnost (*accuracy*): Poměr správně klasifikovaných případů ke všem případům. Vypočítá se jako  $(TP + TN) / (TP + TN + FP + FN)$ .
- Přesnost pozitivní třídy (*precision*): Poměr správně klasifikovaných pozitivních případů ke všem případům, které model klasifikoval jako pozitivní. Vypočítá se jako  $TP / (TP + FP)$ .

- Úplnost (*recall*): Poměr správně klasifikovaných pozitivních případů ke všem skutečným pozitivním případům. Vypočítá se jako  $TP / (TP + FN)$ .
- F1 skóre (*F1-score*) je harmonický průměr mezi přesností (*precision*) a úplností (*recall*) a je často používanou metrikou pro vyhodnocení výkonu klasifikačních modelů, zejména v případech, kdy existují nevyvážené třídy.  $F1 = 2 * \frac{Precision * Recall}{(Precision + Recall)}$ .

Confusion matrix poskytuje komplexní pohled na výkon modelu, což je užitečné zejména pro identifikaci oblastí, ve kterých model nedosahuje optimálního výkonu a může být zlepšen.

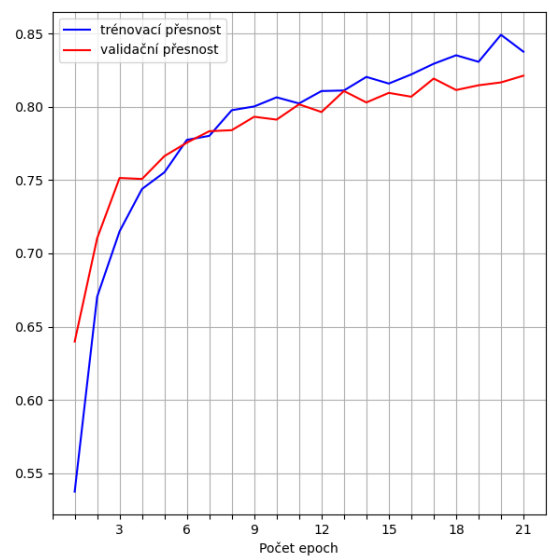
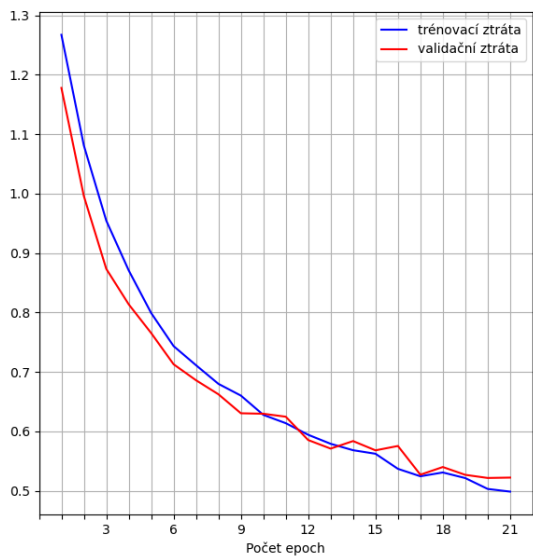
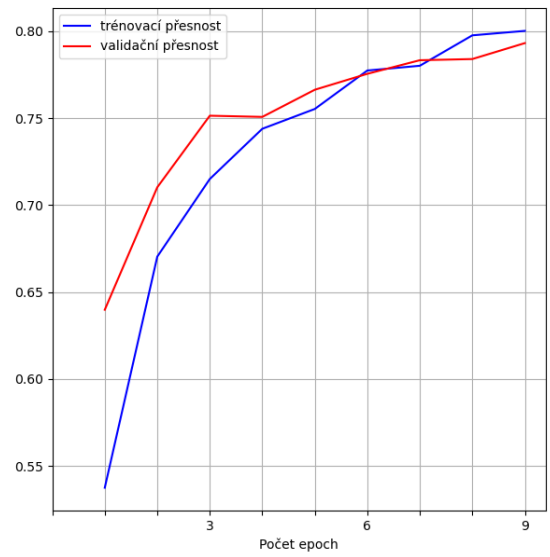
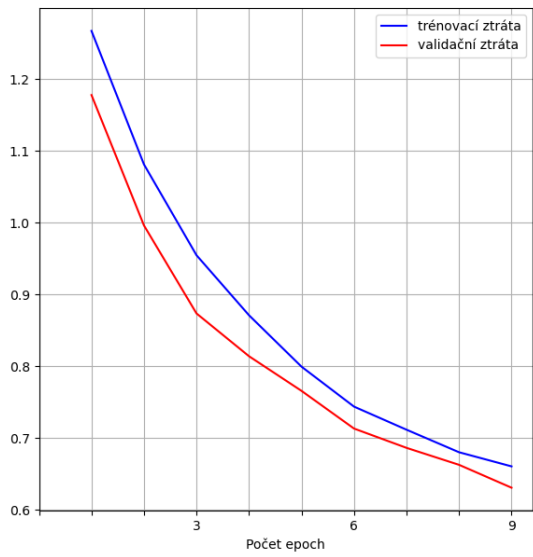
## 3.2 Model EfficientNetB1

Byl navržen tak, aby dosahoval srovnatelných výsledků s klasickými modely, jako je ResNet-50, s menším počtem parametrů a nižší výpočetní náročností. Díky tomu je vhodný pro širokou škálu úloh, včetně rozpoznávání objektů, segmentace obrazu a klasifikace obrázků.

Model EfficientNetB1 (Tan a Le, 2019) je založen na kombinované architektuře s opakujícími se bloky MBConv a mobilní invertovanou zbytkovou konvolucí. Tyto bloky umožňují modelu efektivně využívat dostupné výpočetní zdroje a zároveň dosahovat vysoké přesnosti. Model byl předtrénován na datové sadě ImageNet s více než 14 miliony obrázků (Russakovsky et al., 2015, s. 3) a díky tomu je schopen rozpoznat širokou škálu objektů a kategorií. Model lze dále doladit pro specifické úlohy.

### 3.2.1 Přesnost a ztráta modelu

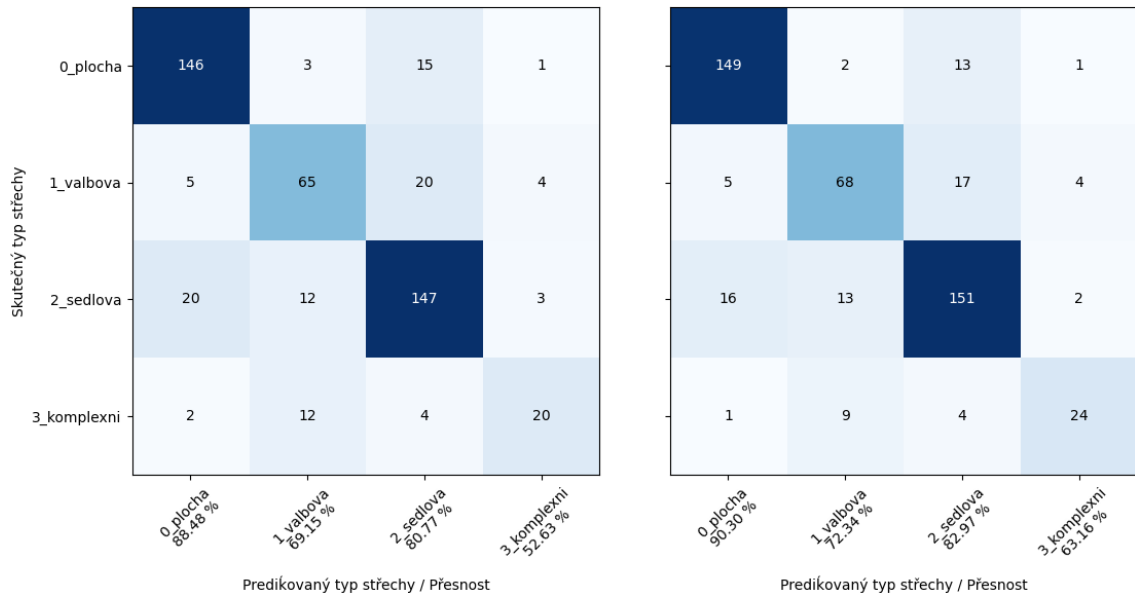
- Trénovací přesnost (*training accuracy*): Přesnost trénování se s počtem epoch zvyšuje, je znakem toho, že se model učí. Objevují se výkyvy, které jsou pravděpodobně způsobeny stochastickou povahou trénování a omezeným souborem dat.
- Validační přesnost (*validation accuracy*): Podobně jako přesnost trénování se zvyšuje s počtem epoch. Malý rozdíl mezi tréninkovou a validační přesností naznačuje, že model může dobře zobecňovat.
- Trénovací ztráta (*training loss*): Ztráta při trénování se s přibývajícím epochami neustále snižuje, což znamená, že model odpovídá tréninkovým datům.
- Validační ztráta (*validation loss*): Validační ztráta se také snižuje a přesně kopíruje křivku ztráty při trénování. Tento minimální rozdíl naznačuje, že model pravděpodobně není výrazně přehnaně přizpůsoben.



**Obr. 11: Graf ztráty a přesnosti modelu EfficientNetB1**  
*Zdroj: vlastní zpracování*

### 3.2.2 Confusion matrix

Z matice je patrné, že učení po více epoch má vliv na zlepšení celkové přesnosti modelu. Chceme-li se ale soustředit na přesnější klasifikaci jednotlivých kategorií, zjistíme, že pro některé typy střechy se s rostoucím počtem epoch zvyšuje více než pro jiné, např. přesnost pro typ *plochá* se zvýší z 88,48 % na 90,30 %, pro typ *komplexní* se zvýší z 52,63 % na 63,16 %.

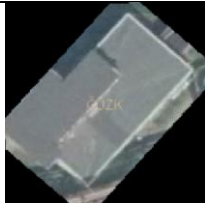
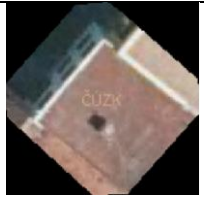

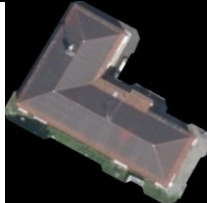

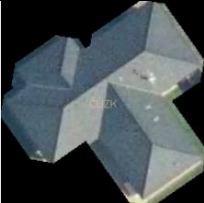
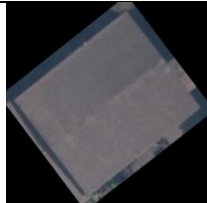



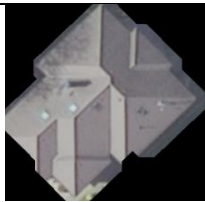



**Obr. 12: Confusion matrix modelu EfficientNetB1, vlevo 9 epoch s celkovou přesností modelu 78,91 %, vpravo 21 epoch s celkovou přesností modelu 81,84 %**

*Zdroj: vlastní zpracování*

Níže uvedená tabulka ukazuje, jak vypadají záměny obrazově, abychom mohli posoudit, s jakými kategoriemi má model potíže.

Tab. 5: Vybrané příklady záměn klasifikace modelu EfficientNetB1

		<i>Předpověď</i>			
		<i>Plochá</i>	<i>Valbová</i>	<i>Sedlová</i>	<i>Komplexní</i>
<i>Skutečnost</i>	<i>Plochá</i>	X			
	<i>Valbová</i>		X		
	<i>Sedlová</i>			X	
	<i>Komplexní</i>				X

Zdroj: vlastní zpracování

### 3.2.3 Celkové hodnocení modelu

Architektura EfficientNetB1, která je známá svou efektivitou, by mohla být pro tuto úlohu a velikost datového souboru vhodná a vést k dobrému zobecnění bez výrazného přetrénování (*overfitting*). Vývoj křivek je ukázkovým příkladem konvergence, kdy se model neustále zlepšuje. Trénování dle experimentu skončilo 21. epochou a je evidentní, že model by bylo možné trénovat po více epoch a dále zlepšovat jeho výkonnost. V tom případě by bylo potřeba pozorně sledovat další vývoj validační ztráty. Mírný nárůst validační ztráty při pokračujícím poklesu tréninkové ztráty by mohl naznačovat počáteční známky přetrénování. Pokud se validační ztráta začne příliš odchylovat od ztráty trénování, je potřeba zvážit možnost zastavení trénování.

## 3.3 Model ResNet-101

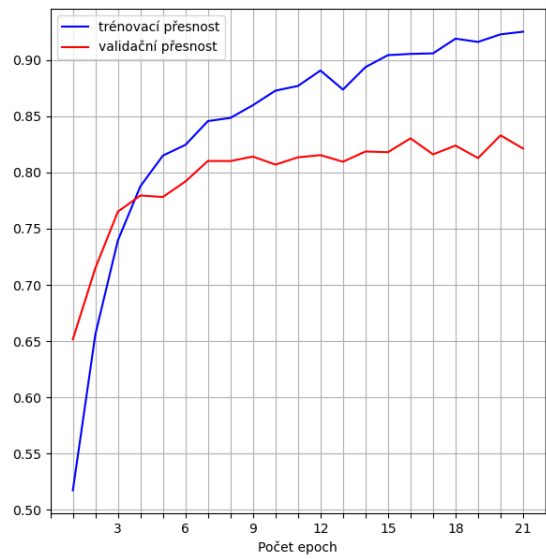
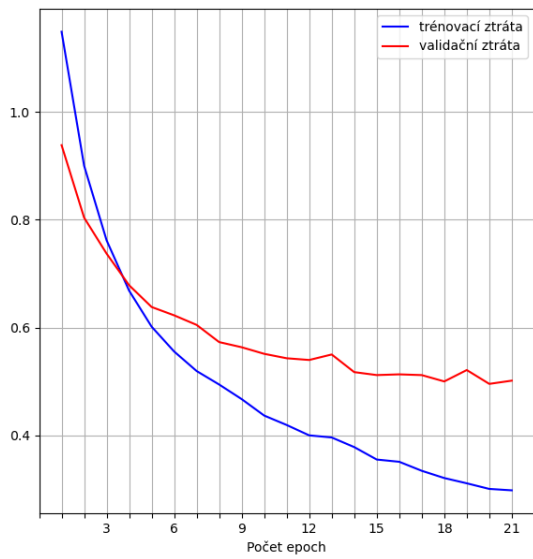
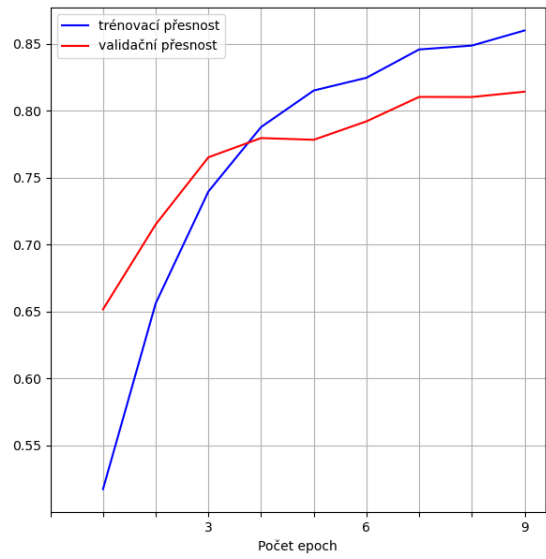
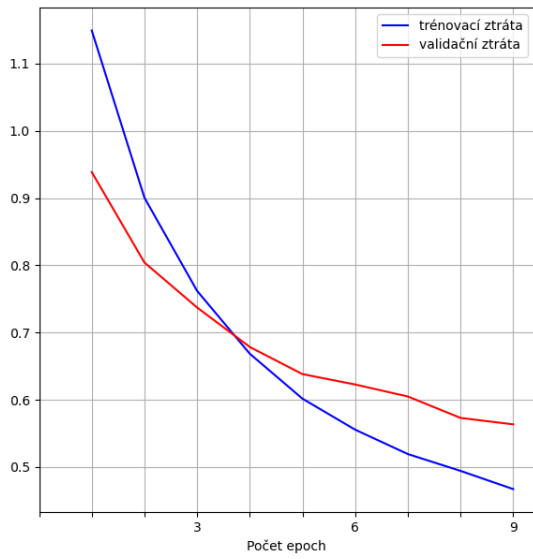
Modely s architekturou ResNet (He et al., 2015) jsou klasické architektury v tomto oboru, kdy číselný příznak udává hloubku architektury. Její inovativní *shortcut* spojení (He et al., 2015, s. 3) umožňuje efektivnější učení a snižují riziko přetrénování, čímž zajišťují vysokou přesnost v rozpoznávání objektů a klasifikaci obrázků. Model byl předtrénován na masivní datové sadě ImageNet a je schopen rozpoznat širokou škálu objektů a kategorií.

I když ResNet-101 vyniká v přesnosti, jeho 44,6 milionů parametrů a vysoká výpočetní náročnost ho omezují v mobilních a vestavěných systémech. V takových případech se jako efektivnější alternativa jeví model EfficientNetB1 s menším počtem parametrů (7,8 milionu) a srovnatelnou, ne-li vyšší přesností.

Při výběru mezi ResNet-101 a EfficientNetB1 je klíčové zvážit specifické požadavky dané úlohy. Pokud je prioritou vysoká přesnost a nevadí nám vyšší výpočetní náročnost, pak je ResNet-101 dobrou volbou. Pokud je prioritou efektivita a kompaktnost, pak je EfficientNetB1 vhodnější (Tan a Le, 2019, s. 6).

### 3.3.1 Přesnost a ztráta modelu

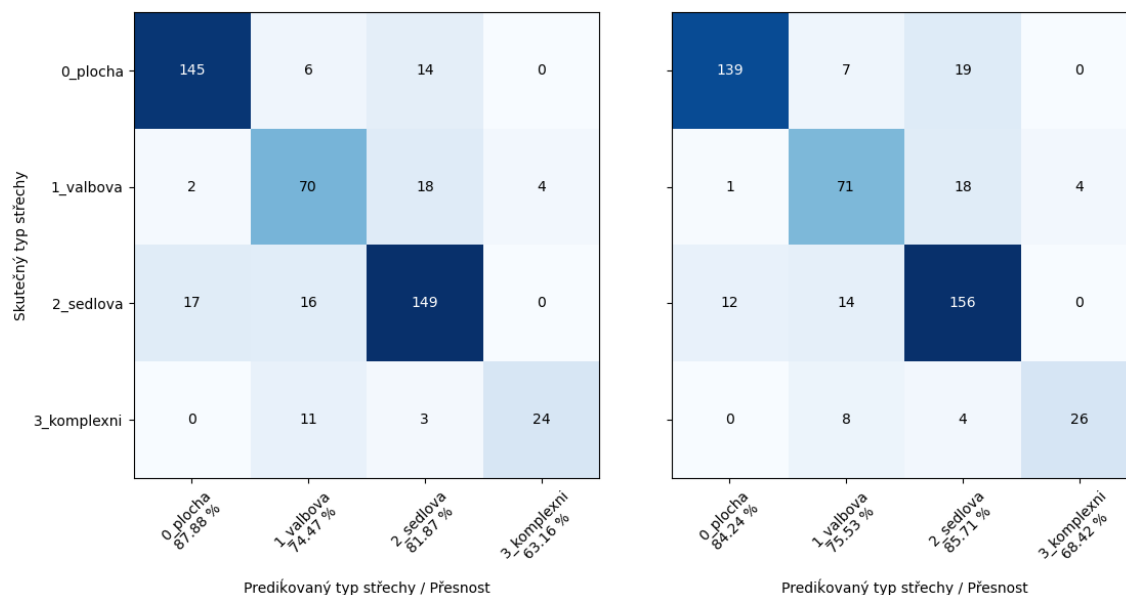
- **Trénovací přesnost (*training accuracy*):** Přesnost trénování obecně roste s počtem epoch, což znamená, že se model učí z trénovacích dat. V přesnosti dochází k určitým výkyvům, které mohou být způsobeny stochastickou povahou procesu trénování nebo omezenou velikostí souboru dat.
- **Validační přesnost (*validation accuracy*):** Přesnost validace se také zvyšuje s počtem epoch, ale zdá se, že je nižší než přesnost trénování. To naznačuje, že model se možná příliš přizpůsobuje trénovacím datům, dochází k přetrénování.
- **Trénovací ztráta (*training loss*):** Ztráta při trénování obecně klesá s počtem epoch, což naznačuje, že model lépe vyhovuje tréninkovým datům. Podobně jako u tréninkové přesnosti dochází ke kolísání křivky ztráty.
- **Validační ztráta (*validation loss*):** Validační ztráta se také snižuje s počtem epoch, ale ve srovnání s tréninkovou ztrátou se ustálí na vyšší hodnotě. To dále potvrzuje možnost nadměrného přizpůsobení.



**Obr. 13: Grafy ztráty a přesnosti modelu ResNet-101**  
*Zdroj: vlastní zpracování*

### 3.3.2 Confusion matrix

Zatímco u valbové, sedlové a komplexní střechy došlo po 21 epochách k mírnému zlepšení přesnosti, u ploché střechy došlo k poklesu z 87,88 % na 84,24 %. Možné příčiny poklesu mohou být způsobeny přetrénováním modelu tak, jak ukazují křivky ztráty. Pokles přesnosti u ploché střechy po 21 epochách naznačuje, že model ResNet-101 není pro tento typ dat optimálně vyladěn. Důkladná analýza dat a další experimentování s hyperparametry by měly vést k vylepšení jeho výkonu.

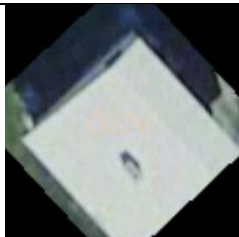
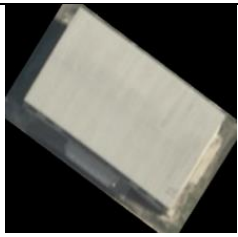


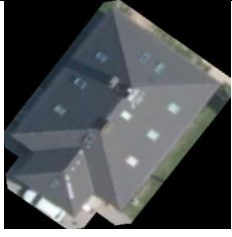
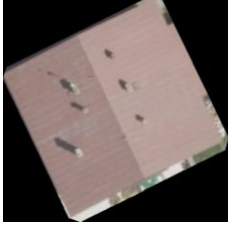





**Obr. 14: Confusion matrix modelu ResNet-101, vlevo 9 epoch s celkovou přesností modelu 81,00 %, vpravo 21 epoch s celkovou přesností modelu 81,84 %**

*Zdroj: vlastní zpracování*

Kromě analýzy matice záměny je důležité prozkoumat i vizuální stránku chyb, které model dělá. To nám pomůže lépe porozumět, s jakými kategoriemi má model potíže a jak můžeme zlepšit jeho výkon. U ploché střechy, kde došlo k poklesu přesnosti, můžeme vizuálně porovnat správně a nesprávně klasifikované obrázky. To nám může napovědět o specifických vlastnostech, které model nesprávně interpretuje. Například můžeme zjistit, že model zaměňuje ploché střechy s valbovými kvůli podobnému úhlu sklonu. Velmi častou záměnou je plochá střecha za sedlovou. U některých plochých střech se objevuje středová linie, která může naznačovat sedlo.

Tab. 6: Vybrané příklady záměn klasifikace modelu ResNet-101

		Předpověď			
		Plochá	Valbová	Sedlová	Komplexní
Skutečnost	Plochá	X			0
	Valbová		X		
	Sedlová			X	0
	Komplexní	0			X

Zdroj: vlastní zpracování

### 3.3.3 Celkové hodnocení modelu

Zdá se, že model se učí z trénovacích dat, protože přesnost i ztráta se zlepšují s počtem epoch. Rozdíl mezi křivkami přesnosti/ztráty při trénování a ověřování však naznačuje, že se model možná příliš přizpůsobuje tréninkovým datům. To znamená, že model funguje dobře na datech, na kterých byl vycvičen, ale nemusí dobře zobecňovat na nezkoumaná data.

Zde se jedná o *overfitting* modelu, tedy jeho přetrénování. Trend tohoto přetrénování lze zachytit poměrně brzy, a to již v šesté epoše. Je zde patrné, že za stávajícího nastavení hyperparametrů dalšího zlepšení nedosáhneme. Pokud bychom se rozhodli použít stávající nastavení, bylo by vhodné trénování modelu zastavit v 18. nebo ve 20. epoše. Tím by si mohl model lépe vést u kategorie *plochá* a nemuselo by docházet u této kategorie ke snížení přesnosti klasifikace.

Pro řešení potenciálního přeučení lze využít některé doporučené techniky. Nabízí se možnost snížení kapacity modelu. To v praxi znamená použití menší architektury a jako v případě Buyukdemirciogla, Can a Kocaman (2021) použít model ResNet-50, který je o polovinu menší než model ResNet-101.

Posledním a obvykle účinným řešením je zvětšení objemu a rozmanitosti trénovací a validační datové sady.

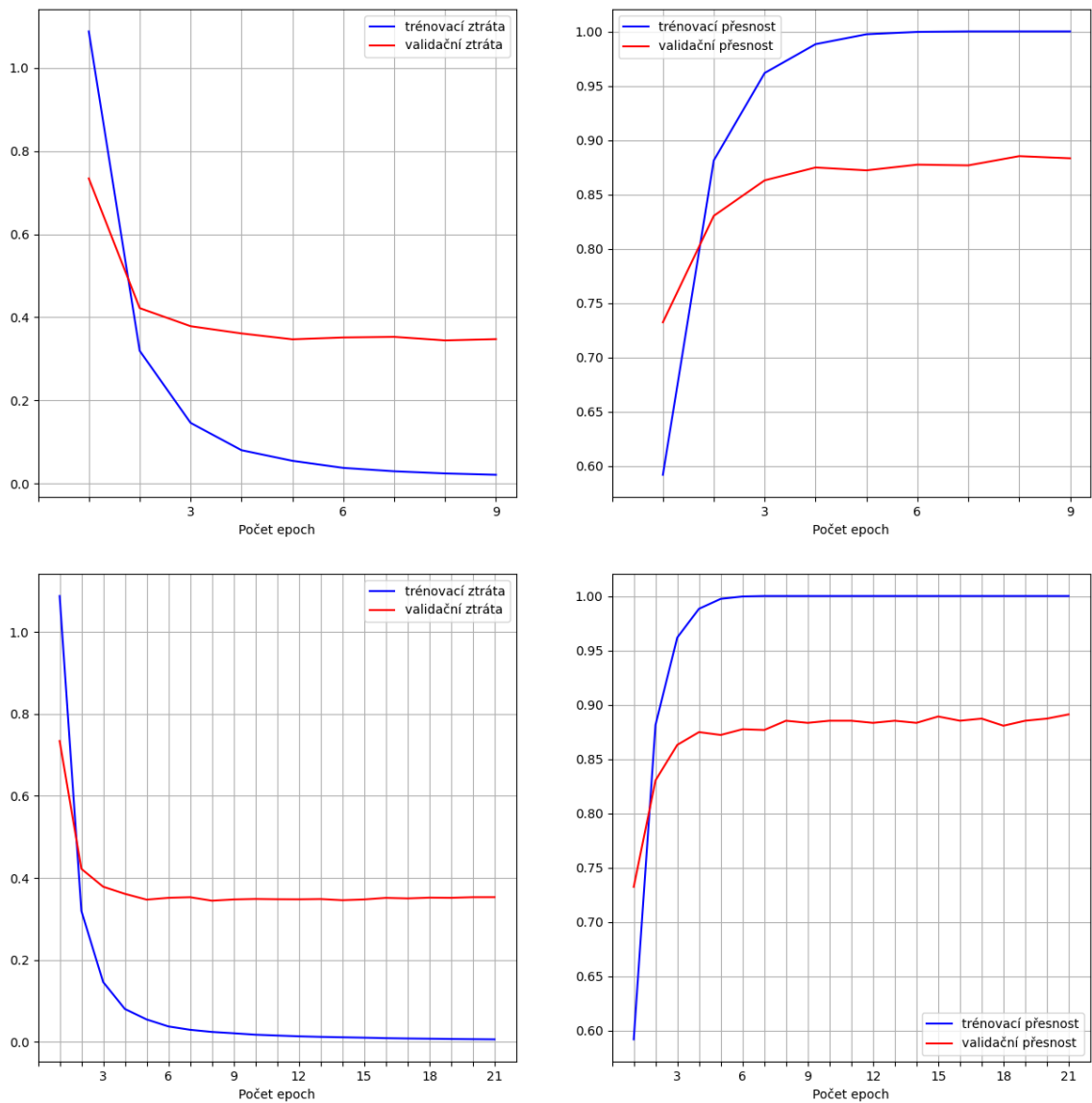
## 3.4 Model VGG-16

Průkopník v oblasti hlubokého učení s 16 konvolučními a plně propojenými vrstvami je pionýrským modelem v oblasti hlubokého učení pro počítačové vidění. I když byl představen v roce 2014, stále se těší velké oblibě a slouží jako základ pro mnoho dalších modelů.

Model VGG-16 (Simonyan a Zisserman, 2014) se vyznačuje jednoduchou architekturou s opakujícími se konvolučními bloky a max-pooling vrstvami. Jeho síla tkví v hloubce sítě, která umožňuje extrahovat z obrázků komplexní a abstraktní vlastnosti. Díky předtrénování na datové sadě ImageNet je VGG-16 schopen rozpoznat širokou škálu objektů a kategorií.

### 3.4.1 Přesnost a ztráta modelu

- Trénovací přesnost (*training accuracy*): Křivka tréninkové přesnosti prudce stoupá v prvních epochách a poté se stabilizuje na vysoké úrovni okolo 98 %. To naznačuje, že model se efektivně učí z tréninkových dat.
- Validační přesnost (*validation accuracy*): Křivka validační přesnosti se ale zvedá mnohem pomaleji než křivka tréninkové přesnosti a dosahuje mnohem nižší úrovně než tréninková přesnost, a to s maximem okolo 88 %. Rozdíl mezi oběma křivkami naznačuje, že model se může potýkat s přetrénováním.
- Trénovací ztráta (*training loss*): Klesá s rostoucím počtem epoch, což potvrzuje, že model se učí a optimalizuje parametry.
- Validační ztráta (*validation loss*): Křivka validační ztráty se zpočátku snižuje, ale v pozdějších epochách stagnuje a mírně se zvedá. Toto chování podporuje hypotézu o přetrénování.



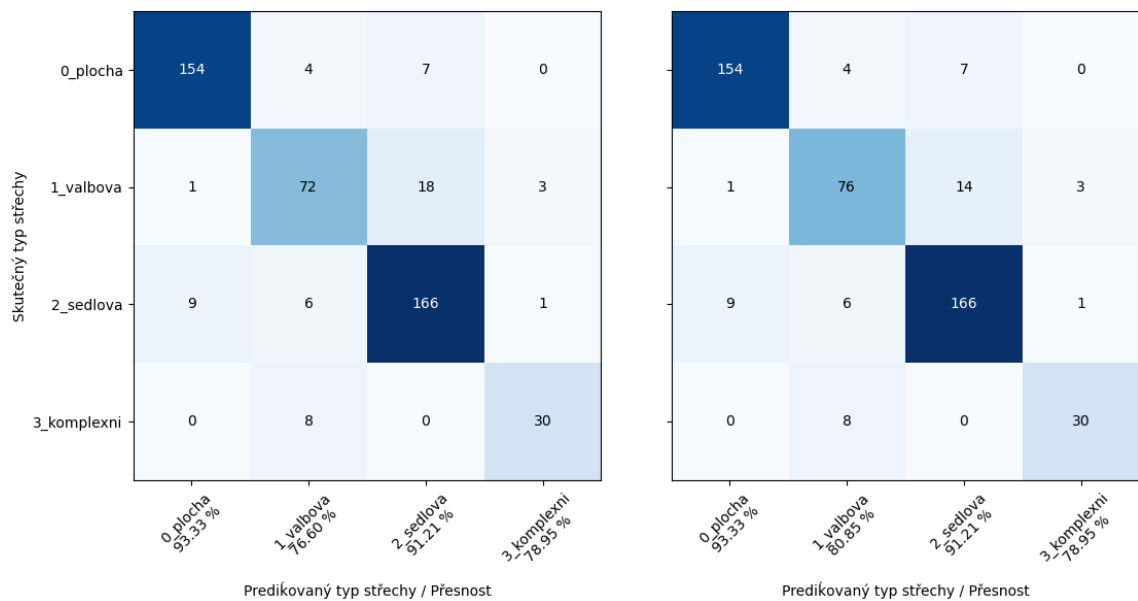
**Obr. 15: Grafy ztráty a přesnosti modelu VGG-16**

*Zdroj: vlastní zpracování*

### 3.4.2 Confusion matrix

Výkonnost modelů VGG-16 v tomto testu je překvapivě dobrá. I když se jedná o nejstarší ze všech testovaných modelů, dosahuje nejlepší přesnosti. To naznačuje, že architektura VGG-16 je stále relevantní a efektivní pro úkoly klasifikace obrázků. Její větší počet parametrů hraje v přesnosti pravděpodobně klíčovou roli. Modely s více parametry obecně dokáží extrahovat z obrázků komplexnější funkce, což vede k lepšímu rozlišování mezi různými kategoriemi. Vzhledem ke svému dobrému výkonu a jednoduchosti by VGG-16 mohl být dobrou volbou pro klasifikaci střech.

Model dosahuje vynikající přesnosti u všech typů střech s nejvyššími hodnotami u plochých a sedlových střech. To ukazuje, že model je schopen spolehlivě klasifikovat různé typy střech. Vzhledem k jeho dobrému výkonu se VGG-16 jeví jako dobrá volba pro úkoly klasifikace střech. Jeho jednoduchost také usnadňuje jeho implementaci a nasazení.





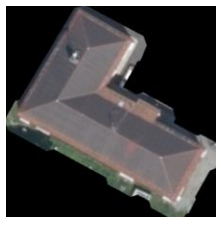
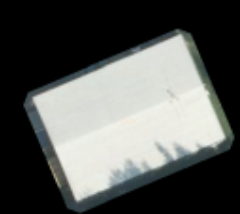


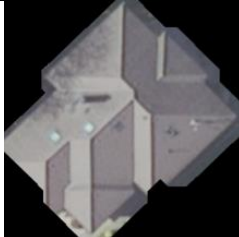


**Obr. 16: Confusion matrix modelu VGG-16, vlevo 9 epoch s celkovou přesností modelu 88,10 %, vpravo 21 epoch s celkovou přesností modelu 88,94 %**

*Zdroj: vlastní zpracování*

I když VGG-16 dosahuje vynikající přesnosti, je důležité provést vizuální analýzu chyb, abychom lépe porozuměli jeho slabinám a zjistili, zda existují specifické typy střech, které model klasifikuje nesprávně. Jeho největší slabinou je 14 valbových střech klasifikovaných jako sedlové. Jedná se o nejslabší místo všech testovaných modelů. Za uspokojivé považujeme snížení chybovosti u plochých střech, které byly klasifikovány jako sedlové. Zde je model značně lepší než předchozí dva modely.

Tab. 7: Vybrané příklady záměn klasifikace modelu VGG-16

		<i>Předpověď</i>			
		<i>Plochá</i>	<i>Valbová</i>	<i>Sedlová</i>	<i>Komplexní</i>
<i>Skutečnost</i>	<i>Plochá</i>	X			0
	<i>Valbová</i>		X		
	<i>Sedlová</i>			X	
	<i>Komplexní</i>	0		0	X

*Zdroj: vlastní zpracování*

### 3.4.3 Celkové hodnocení modelu

Křivky VGG-16 se odlišují od křivek ostatních modelů v několika ohledech. Tréninková a validační přesnost VGG-16 dosahuje v porovnání s ostatními modely nejvyšších hodnot. Rozdíl mezi tréninkovou a validační přesností VGG-16 je větší než u ostatních modelů, což naznačuje silnější přeučení. Křivka validační ztráty VGG-16 stagnuje a mírně se zvedá, zatímco u ostatních modelů klesá.

Zdá se, že model VGG-16 se efektivně učí z tréninkových dat a dosahuje vysoké tréninkové přesnosti. Nicméně značný rozdíl mezi tréninkovou a validační přesností a stagnující validační ztráta naznačují, že model se potýká s přetrénováním. I tak ale dosahuje nejvyšší celkové přesnosti klasifikace i klasifikace napříč třídami.

Výsledky experimentu naznačují, že VGG-16 je stále relevantní a efektivní model pro klasifikaci střech. Jeho větší počet parametrů a jednoduchost pravděpodobně přispívají k jeho vynikající přesnosti. Je však důležité dále testovat a porovnávat VGG-16 s jinými modely, abychom získali komplexní obraz jeho výkonu a možností.

Pro odstranění přetrénování tohoto modelu platí stejná pravidla jako pro model ResNet-101 v kapitole 3.3.3.

## 3.5 Celkové vyhodnocení experimentu

Celkové vyhodnocení experimentu se skládá ze dvou částí. V první části proběhlo srovnání tří modelů, EfficientNetB1, ResNet101 a VGG-16, trénovaných po dobu 21 epoch s náhodným seedem 13. Výsledky pro 9 epoch zde neuvádíme, jelikož u všech modelů dochází k lepším výsledkům v pozdějších epochách. Pro srovnání výkonu byly použity standardní metriky klasifikace pro čtyři kategorie střech: *plochá*, *valbová*, *sedlová* a *komplexní*. Hodnoceny byly přesnost klasifikace v jednotlivých kategoriích, precision, recall, F1 skóre a celková přesnost napříč kategoriemi.

V druhé části experimentu budou naše výsledky porovnány s výsledky prezentovanými v referenčních pracích. Je nutné zdůraznit, že toto srovnání bude pouze orientační, jelikož se liší sledované parametry i použité datové sady. I tak však toto srovnání dává vhled do problematiky a ukazuje, že námi navrhovaná metoda klasifikace typů střech založená na odvození obrazu střechy z veřejně dostupných mapových služeb dosahuje srovnatelných a v některých případech i lepších výsledků než metody prezentované v referenčních pracích z této oblasti.

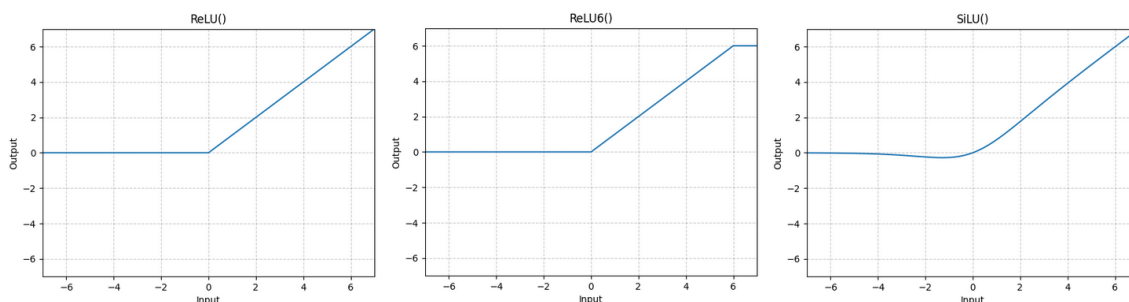
### 3.5.1 Celkové hodnocení testovaných modelů

Na základě výsledků v níže uvedené tabulce můžeme s jistotou prohlásit, že nejlepším modelem pro danou úlohu se stala architektura VGG-16, a to i přes to, že se jedná o nejstarší testovaný model. VGG-16 dosáhla nejvyššího průměrného skóre přesnosti (93,9 %), precision (93,9 %) a F1 skóre (93,6 %) napříč všemi typy střech.

Naproti tomu modely ResNet-101 a EfficientNetB1 dosáhly nižších průměrných hodnot. ResNet-101 exceloval v precision (91,4 %) u plochých střech, ale v ostatních kategoriích za VGG-16 zaostal. EfficientNetB1 se ukázal jako nejméně spolehlivý model s nejnižší průměrnou přesností (79,0 %) a F1 skóre (83,3 %).

Existuje několik faktorů, které mohou přispívat k nadřazenosti VGG-16 v tomto experimentu:

- Hloubka sítě: VGG-16 má hlubší architekturu než ResNet-101 a EfficientNetB1, což mu umožňuje extrahovat z obrázků komplexnější rysy. Tyto komplexnější rysy mohou být pro klasifikaci typů střech užitečnější.
- Počet parametrů: VGG-16 má více parametrů než ResNet-101 a EfficientNetB1. To mu umožňuje ukládat více informací o datech, což může vést ke zlepšení přesnosti.
- Aktivační funkce: VGG-16 používá aktivační funkci ReLU, zatímco ResNet-101 a EfficientNetB1 používají funkce ReLU6 a Swish (Ramachandran, Zoph a Le, 2017). ReLU se ukázala jako efektivnější pro úkoly klasifikace obrazu než ReLU6 a Swish.



**Obr. 17: Aktivační funkce, zleva ReLU, ReLU6 a Swish (v PyTorch jako SiLU)**

Zdroj: vlastní zpracování dle informací a obrázku uvedených na adrese <https://pytorch.org/docs/stable/generated/torch.nn>

Je důležité poznamenat, že se jedná pouze o spekulace a pro definitivní závěr by bylo potřeba provést další testy. Nicméně výsledky v tabulce jasně ukazují, že VGG-16 je pro danou úlohu klasifikace typů střech efektivnější architekturou než ResNet-101 a EfficientNetB1.

**Tab. 8: Výsledky testovaných modelů**

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>	<i>Overall accuracy</i>
<b>EfficientNetB1</b>					
Plochá	90,3 %	87,1 %	90,3 %	88,7 %	81,8 %
Valbová	72,3 %	73,9 %	72,3 %	73,1 %	
Sedlová	83,1 %	81,6 %	83,1 %	82,3 %	
Komplexní	63,2 %	77,4 %	63,2 %	69,6 %	
<b>ResNet-101</b>					
Plochá	84,2 %	91,4 %	84,2 %	87,7 %	81,8 %
Valbová	75,5 %	71,0 %	75,5 %	73,2 %	
Sedlová	85,7 %	79,2 %	85,7 %	82,3 %	
Komplexní	68,4 %	86,7 %	68,4 %	76,5 %	
<b>VGG-16</b>					
Plochá	93,9 %	93,9 %	93,3 %	93,6 %	88,9 %
Valbová	80,9 %	80,9 %	80,9 %	80,9 %	
Sedlová	91,2 %	88,8 %	91,2 %	90,9 %	
Komplexní	79,0 %	88,2 %	78,9 %	83,3 %	

Zdroj: vlastní zpracování

Kromě výše uvedených bodů je důležité zmínit, že výběr optimálního modelu pro danou úlohu závisí na mnoha faktorech, jako jsou dostupný výpočetní výkon, požadovaná přesnost a dostupná data. V tomto případě se VGG-16 ukázala jako nejlepší volba, ale pro jiné úlohy se můžou hodit lépe modely ResNet-101 nebo EfficientNetB1.

### 3.5.2 Celkové hodnocení napříč prezentovanými pracemi

V tabulce níže jsou uvedeny výsledky navrhované metody a výsledky prezentované v referenčních pracích. Z provedeného srovnání s výsledky prezentovanými v referenčních pracích vyplývá, že navrhovaná metoda klasifikace typů střech dosahuje srovnatelných výsledků. Všechny čtyři metody (včetně naší) dosáhly nejlepších výsledků s architekturou VGG-16, ať už v modifikované (VGG-19), nebo v původní verzi. Výjimkou je Qian, který nepoužívá klasickou neuronovou síťovou architekturu, ale expertní systém MTBR složený z více architektur. Tento

system dosahuje srovnatelných výsledků s VGG, avšak s nižší přesností, ale vzhledem k pixelovému rozlišení a pokrytí urbanistických kategorií se jedná o velmi kvalitní výsledek.

**Tab. 9: Celkové porovnání napříč pracemi**

	<i>Navrhovaná metoda</i>	<i>Buyukdemircioglu, Can a Kocaman (2021)</i>	<i>Meng, Soleimani-Babakamali a Taciroglu (2023)</i>	<i>Qian et al. (2023)</i>
Celková přesnost klasifikace	89 %	86 %	86 %	84 %
Vítězná architektura	VGG-16	VGG-16	VGG-19	MTBR
Počet kategorií	4	6	6	4
Původ snímku	Ortofotomapa ČR	Letecké snímkování	Google mapy	Družice SkySat
Velikost pixelu [px/cm]	12,5	10	15	50
Přibližná velikost datasetu	3 000	4 500	11 500	18 000
Lokalita snímků	celá ČR	Cesme (Turecko)	North Carolina & Miami-Dade County (USA)	Šanghaj (Čína)
Urbanistické kategorie	bydlení, občanská	nezjištěno	bydlení (solitérní domy)	bydlení, občanská, průmyslová

*Zdroj: vlastní zpracování*

Je důležité poznamenat, že se sledované parametry a použité datové sady u jednotlivých metod liší. Buyukdemircioglu a Meng klasifikovali snímky do šesti kategorií, zatímco Qian do čtyř. Buyukdemircioglu používal snímky s nejvyšším rozlišením (10 px/cm), zatímco Qian s nejnižším (50 px/cm). Meng použil snímky s rozlišením 15 px/cm. Počet snímků v datových sadách se pohybuje přibližně od 3 000 do 18 000. Lokality a urbanistické kategorie střech jsou také rozdílné. I přes tyto rozdíly v parametrech a datech dosahují všechny metody srovnatelných výsledků.

## Závěr

V rámci práce byla navržena a otestována metoda pro identifikaci typů střech budov za použití technik hlubokého učení. Pro testování byla použita vlastní datová sada, jejíž pořízení bylo komentováno a diskutováno jak v teoretické, tak i praktické části. Jádro této metody tkví v integraci dvou veřejně dostupných služeb: Registru územní adresy a nemovitostí (RÚAIN) a Ortofotomapy České republiky. RÚAIN poskytuje detailní informace o budovách, včetně jejich typu, adresy a dalších relevantních údajů. Ortofotomapa nabízí vysoce kvalitní letecké snímky, které slouží jako vizuální podklad pro identifikaci typů střech. Kombinace těchto dvou zdrojů dat umožňuje metodě dosahovat vysoké přesnosti při klasifikaci typů střech. Výsledky praktické části jasně ukázaly, že způsob segmentace střech je zcela dostačující a jako takový ho lze označit za průlomový. Navržená metoda tedy šetří čas a náklady na pořizování datové sady pro konvoluční síť. Z hlediska přípravy podkladů pro strojové učení zůstala jen část ručního zařazení snímků do jednotlivých kategorií a tento krok je u každého modelu zatím nevyhnutelný.

Pro produkční nasazení navržené metody bude nutné znovu prodiskutovat kategorie střech, a to s ohledem na požadavky urbanismu a architektury. Kategorie definované v rámci této práce sloužily primárně pro účely umisťování FTV panelů, avšak postrádají mezioborový přesah nezbytný pro širší využití datové sady a metody strojového učení jako celku. Je nutné precizněji klasifikovat typy střech v kontextu urbanistických kategorií volně definovaných v této práci jako *bydlení* (Rozmanová et al., 2013), *občanská* (Šindlerová et al., 2020) a *průmyslová* (Rozmanová et al., 2016), které by měly odpovídat plochám s rozdílným způsobem využití (Vyhláška č. 501/2006 Sb.). Tím by mohlo dojít ke spojení těchto kategorií s informacemi o způsobu využití objektu dle kapitoly 2.1.5. a bylo by možné do klasifikace zapojit další údaje dle RÚIAN, jako jsou druh budovy a další relevantní metadatové údaje. Takto rozšířená datová sada naleznе uplatnění i v jiných oblastech, zejména v tvorbě 3D městských map dle standardu CityGML. Může tak komplementovat metodu pro tvorbu 3D modelů budov dle Hrona (2019), která se opírá o princip extrakce střešních hran.

Samotná klasifikace do čtyř kategorií střech se ukázala jako opodstatněná a dosáhla srovnatelných výsledků s relevantními publikacemi. V tomto ohledu se ukázalo jako velmi efektivní využití zavedené architektury VGG-16 s náhodným seedem 13. Výsledky jasně demonstřují potenciál transfer learningu pro danou problematiku. Zmíněné výsledky otevírají cestu pro další výzkum v oblasti klasifikace střech budov. Konkrétně by bylo možné se zaměřit na rozšíření datové sady o všechny urbanistické kategorie nebo další kategorie střech.

Výzkum na úrovni nastavení hyperparametrů vytyčil směr, který může být výchozím bodem k dalšímu experimentování a zvyšování přesnosti klasifikace. Zde můžeme jen doporučit rozšířit datovou sadu, vzít vítězný model a zkusit jej automatizovaně v různých kombinacích hyperparametrů trénovat na výkonném clusteru za použití jemného doladění (*fine tuning*) některých horních konvolučních bloků (Chollet, 2019, s. 148).

Závěrem lze shrnout, že práce splnila všechny cíle stanovené v úvodu. V praktické části se podařilo vyvinout průlomovou metodu segmentace střech, která se vyznačuje nenáročností na pořizování dat, a ukázalo se, že datová sada RÚAIN je vhodná pro lokalizaci budov a segmentaci střech.

Vlastní pořízení datové sady ukázalo, že lze uvedeným způsobem získat sadu z veřejných zdrojů. Výsledky experimentální části zřetelně prokázaly, že rezidua vzniklá při segmentaci nemají vliv na výkonnost použitých modelů.

V části zaměřené na strojové učení byly implementovány a otestovány osvědčené hluboké architektury, které potvrdily efektivitu principu transfer learningu pro danou problematiku.

Dosavadní výsledky otevírají cestu pro další výzkum v oblasti klasifikace střech budov. V budoucnu se zaměříme na rozšíření datové sady o další urbanistické kategorie střech, preciznější definování a klasifikaci typů střech v kontextu urbanismu a na optimalizaci hyperparametrů modelu.

## Seznam použité literatury

- 910, 2024. Ortofoto České republiky – úvod. In: *Geoportal.cuzk.cz* [online]. 2. 4. 2024 [cit. 2024-06-02]. Dostupné z: [https://geoportal.cuzk.cz/\(S\(on5w4h3qf15gbxxshyxmxvya\)\)/default.aspx?mode=TextMeta&text=ortofoto\\_info&side=ortofoto&menu=23](https://geoportal.cuzk.cz/(S(on5w4h3qf15gbxxshyxmxvya))/default.aspx?mode=TextMeta&text=ortofoto_info&side=ortofoto&menu=23).
- BENGIO, Yoshua, 2012. Practical recommendations for gradient-based training of deep architectures In: *Arxiv.org* [online]. 24. 6. 2012 [cit. 2024-06-02]. Dostupné z: <http://arxiv.org/abs/1206.5533>
- BILJECKI, Filip, et al., 2015. Raise the Roof: Towards Generating Lod2 Models Without Aerial Surveys Using Machine Learning. *International archives of the photogrammetry, remote sensing and spatial information sciences* [online]. IV-4/W8, 27–34 [cit. 2024-06-02]. ISSN 2194-9034. Dostupné z: <https://doi.org/10.5194/isprs-annals-IV-4-W8-27-2019>
- BUYUKDEMIRCIOGLU, Mehmet, Recep CAN a Sultan KOCAMAN, 2021. Deep learning based roof type classification using very high resolution aerial imagery. *International archives of the photogrammetry, remote sensing and spatial information sciences* [online]. XLIII-B3-2021, 55–60 [cit. 2024-06-02]. ISSN 2194-9034. Dostupné z: <https://doi.org/10.5194/isprs-archives-XLIII-B3-2021-55-2021>
- ČÚZK, 2023. Registr územní identifikace, adres a nemovitostí (RÚIAN). *Cuzk.cz* [online]. © 2023 [cit. 2024-06-02]. Dostupné z: <https://www.cuzk.cz/ruian/RUIAN.aspx>.
- FRIDMAN, Lex, 2019. Most Research in Deep Learning is a Total Waste of Time - Jeremy Howard | AI Podcast Clips | Summary and Q&A. In: *Glasp.co* [online]. 10. 9. 2019 [cit. 2024-06-02]. Dostupné z: <https://glasp.co/youtube/p/most-research-in-deep-learning-is-a-total-waste-of-time-jeremy-howard-ai-podcast-clips>
- GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE, 2016. *Deep learning. Adaptive computation and machine learning series*. Cambridge: MIT press. ISBN 978-0262035613.
- GOOGLE, 2024. Google Colaboratory. *Colab.research.google.com* [online]. © 2024 [cit. 2024-06-02]. Dostupné z: <https://colab.research.google.com/>
- HE, Kaiming et al., 2015. Deep Residual Learning for Image Recognition. In: *Arxiv.org* [online]. 10. 12. 2015 [cit. 2024-06-02]. Dostupné z: <https://doi.org/10.48550/ARXIV.1512.03385>
- HOWARD, Jeremy a Sylvain GUGGER, 2020. *Deep learning for coders with Fastai and PyTorch: Ai applications without a PhD. Sebastopol*. Sebastopol: O'Reilly. ISBN 978-1-492-04552-6.
- HRON, Vojtěch, 2019. *Využití leteckých dat pro aktualizaci budov v ZABAGED®*. Disertační práce. Praha. České vysoké učení technické, Fakulta stavební, Katedra geomatiky.
- HUNTER, D. John, 2007. Matplotlib: A 2D graphics environment, Computing in Science & Engineering. *Computing in Science & Engineering* [online]. 9(3), 90–95 [cit. 2024-06-02]. ISSN 2676-542X. Dostupné z: <https://ieeexplore.ieee.org/document/4160265>
- CHOLLET, François, 2019. *Deep learning v jazyku Python: knihovny Keras, Tensorflow*. Přeložil Rudolf PECINOVSKÝ. Praha: Grada Publishing. ISBN 9788024731001.
- KINGMA, P. Diederik a J. BA, 2017. Adam: A Method for Stochastic Optimization'. In: *Arxiv.org* [online]. 22. 12. 2014 [cit. 2024-06-02]. Dostupné z: <http://arxiv.org/abs/1412.6980>

- KLÍMA, Jan, 2015. *Zásuvný modul (plugin) QGISu import dat registru RUIAN* [online]. České vysoké učení technické, Katedra geomatiky, 2015 [cit. 2023-12-21]. Dostupné z: <https://portal.fsv.cvut.cz/svoc/2015/gak-klim.pdf>
- KROHN, Jon, Grant BEYLEVELD a Aglaé BASSENS, 2020. *Deep learning illustrated: a visual, interactive guide to artificial intelligence*. Boston: Addison-Wesley. ISBN 978-0-13-511669-2.
- KUHN, Max a Kjell JOHNSON, 2013. *Applied predictive modeling*. New York: Springer. ISBN 978-1461468486.
- LIU, Penghua et al., 2019. Building Footprint Extraction from High-Resolution Images via Spatial Residual Inception Convolutional Neural Network. *Remote Sensing* [online]. 11(7), 830 [cit. 2023-11-18]. ISSN 2072-4292. Dostupné z: <https://www.mdpi.com/2072-4292/11/7/830>
- MAGGIORI, Emmanuel et al., 2017. Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark. *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)* [online]. United States, Fort Worth [cit. 2023-11-16]. Dostupné z: <https://inria.hal.science/hal-01468452>
- MAO, Anqi et al., 2023. Cross-Entropy Loss Functions: Theoretical Analysis and Applications. In: KRAUSE, A. et al. (eds.) *Proceedings of the 40th International Conference on Machine Learning* [online]. PMLR (Proceedings of Machine Learning Research), s. 23803–23828 [cit. 2023-11-16]. Dostupné z: <https://proceedings.mlr.press/v202/mao23b.html>
- MCDERMONT, James, 2024. Hands-on Transfer Learning with Keras and the VGG16 Model. In: *Learndatasci.com* [online]. © 2024 [cit. 2023-11-16]. Dostupné z: <https://www.learndatasci.com/tutorials/hands-on-transfer-learning-keras/>
- MENG, Shuochuan, Mohammad Hesam SOLEIMANI-BABAKAMALI a Ertugrul TACIROGLU, 2023. Automatic Roof Type Classification Through Machine Learning for Regional Wind Risk Assessment. In: *Arxiv.org* [online]. 27. 5. 2023 [cit. 2023-11-16]. Dostupné z: <https://doi.org/10.48550/arXiv.2305.1731>
- MERKEL, Dirk, 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 2014(239). ISSN 1075-3583.
- OPEN GEOSPATIAL CONSORCIUM, 2024. CityGML. *Ogc.org* [online]. © 2024 [cit. 2024-02-25]. Dostupné z: <https://www.ogc.org/standard/citygml>
- OSINGA, Douwe, 2018. *Deep learning cookbook: practical recipes to get started quickly*. Cambridge: O'Reilly. ISBN 9781491995846.
- PAWAN\_DUBEY A ARVINDPDMN, 2019. Confusion Matrix. In: *Devopedia.org* [online]. 20. 8. 2019 [cit. 2024-06-02]. Dostupné z: [Dostupné z: https://devopedia.org/confusion-matrix](https://devopedia.org/confusion-matrix)
- PYTORCH FOUNDATION, 2023. PyTorch documentation. *Pytorch.org* [online]. © 2023 [cit. 2024-06-02]. Dostupné z: <https://pytorch.org/docs/stable/index.html#pytorch-documentation>

- QIAN, Zhen et al., 2023. Multi-task deep learning for large-scale building detail extraction from high-resolution satellite imagery. In: *Arxiv.org* [online]. 29. 10. 2023 [cit. 2023-11-16]. Dostupné z: <https://arxiv.org/abs/2310.18899>
- RAMACHANDRAN, P., Barret ZOPH a Quoc V. LE, 2017. Searching for Activation Functions. In: *Arxiv.org* [online]. 16. 10. 2017 [cit. 2024-06-02]. Dostupné z: <http://arxiv.org/abs/1710.05941>
- ROZMANOVÁ Naděžda et al., 2013. *Principy a pravidla územního plánování – Bydlení* [online]. Brno: Ústav územního rozvoje [cit. 2024-06-02]. <https://www.uur.cz/media/t0gdgqay/c3-2013.pdf>
- ROZMANOVÁ Naděžda et al., 2016. *Principy a pravidla územního plánování – Výroba, těžba a skladování* [online]. Brno: Ústav územního rozvoje [cit. 2024-06-02]. Dostupné z: <https://www.uur.cz/media/hbqhywa3/c1-20160809.pdf>
- RUSSAKOVSKY, O. et al., 2015. ImageNet Large Scale Visual Recognition Challenge. In: *Arxiv.org* [online]. 1. 9. 2014 [cit. 2024-06-02]. Dostupné z: <http://arxiv.org/abs/1409.0575>
- RUSSELL, Stuart a Peter NORVIG, 2021. *Artificial Intelligence A Modern Approach. Fourth edition*. London: Pearsons Education. ISBN 9781292401133.
- SIMONYAN, Karen a Andrew ZISSERMAN, 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *Arxiv.org* [online]. 4. 9. 2014 [cit. 2024-06-02]. Dostupné z: <https://arxiv.org/abs/1409.1556>
- SMITH, Oliver, 2023. Ubuntu Desktop 23.10: Mantic Minotaur deep dive. In: *Canonical.com* [online]. 12. 10. 2023 [cit. 2024-06-15]. Dostupné z: <https://canonical.com/blog/ubuntu-desktop-23-10-mantic-minotaur-deep-dive>
- ŠINDLEROVÁ, Veronika et al., 2020. *Principy a pravidla územního plánování – Občanské vybavení* [online]. Brno: Ústav územního rozvoje [cit. 2024-06-02]. Dostupné z: <https://www.uur.cz/media/3qan0ips/c4-2020-12-18.pdf>
- TAN, Mingxing a Quoc V. LE, 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: *Arxiv.org* [online]. 28. 5. 2019 [cit. 2024-06-02]. Dostupné z: <https://arxiv.org/abs/1905.11946>
- TENSORFLOW, 2023. Get started with TensorBoard. In: *Tensodflow.org* [online]. 25. 10. 2023 [cit. 2024-06-02]. Dostupné z: [https://www.tensorflow.org/tensorboard/get\\_started](https://www.tensorflow.org/tensorboard/get_started)
- Vyhláška č. 501/2006 Sb., o obecných požadavcích na využívání území.