



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## SYSTÉM DETEKCE ZMĚN UDÁLOSTÍ ZE DRUŽICOVÝCH DAT POMOCÍ UMĚLÉ INTELIGENCE

SYSTEM FOR CHANGE DETECTION FROM SATELLITE DATA USING ARTIFICIAL INTELLIGENCE

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

**Bc. Vilém Pecháček**

### VEDOUCÍ PRÁCE

SUPERVISOR

**Ing. Ján Mucha, Ph.D.**

**BRNO 2025**



# Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Vilém Pecháček

**ID:** 230626

**Ročník:** 2

**Akademický rok:** 2024/25

**NÁZEV TÉMATU:**

## Systém detekce změn událostí ze družicových dat pomocí umělé inteligence

**POKYNY PRO VYPRACOVÁNÍ:**

Satelitní pozorování jsou klíčová pro různé bezpečnostní aplikace, řízení katastrof nebo ekologické monitorování. Tradiční metody zpracování dat zahrnují přenos dat na pozemní stanice k analýze, což přináší zpoždění a neefektivitu. Cílem práce je navrhnout a implementovat systém detekce změn událostí ze satelitních snímků pomocí metod umělé inteligence s důrazem na efektivní využívání výpočetních zdrojů. Bude vykonána rešerše současných řešení pomocí lehkých neuronových sítí. Dále bude vhodně vybrán volně dostupný dataset satelitních snímků (např. Spot 5). Dataset bude předzpracován a statisticky popsán. Na základě rešerše bude zvolena vhodná metoda umělé inteligence pro detekci změn událostí (variační auto-ekodér, konvoluční neuronové sítě apod.). Systém bude evaluován pomocí standardních metrik za účelem porovnání výkonu s obdobnými systémy a optimalizován za účelem dosažení co nejnižších výpočetních zdrojů.

**DOPORUČENÁ LITERATURA:**

[1] Agency, ESA. Spot 5, Earth Online. Dostupné na: <https://earth.esa.int/eogateway/missions/spot-5> (Accessed: 16 June 2024).

[2] RŮŽIČKA, Vít, et al. RaVÆn: unsupervised change detection of extreme events using ML on-board satellites. Scientific reports, 2022, 12.1: 16939.

**Termín zadání:** 10.2.2025

**Termín odevzdání:** 27.5.2025

**Vedoucí práce:** Ing. Ján Mucha, Ph.D.

**Konzultant:** Bc. Radoslav Pitoňák

**prof. Ing. Jiří Mišurec, CSc.**  
předseda rady studijního programu

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Družicové pozorování sehrává zásadní roli v mnoha oblastech, jako je urbanizace nebo ekologické monitorování. Rychlé a přesné vyhodnocování satelitních snímků je klíčové pro rozhodování v reálném čase i pro dlouhodobé plánování. Tradiční metody zpracování dat obvykle způsobují zpoždění, vysoké náklady a nízkou efektivitu. Tato omezení představují významnou překážku v situacích, kde je rychlá reakce klíčová. Tato diplomová práce se zabývá návrhem a implementací systému pro automatickou detekci změn na družicových snímcích s využitím metod umělé inteligence a hlubokého učení. V práci je kladen důraz na zlepšení výpočetní efektivity, která je nezbytná pro zpracování velkých objemů dat v omezených výpočetních prostředích, například přímo na družicích. Výstupem práce je trojice modelů, které jsou srovnány pomocí standardních metrik. Výsledky ukázaly, že nejvyšší hodnoty poměru průniku a sjednocení (Intersection over Union, IoU) dosáhl model U-net (0,7794). Kompaktnější architektury založené na MobileNetV2 vykazovaly nižší hodnoty IoU (0,7090, resp. 0,7383), nicméně i nadále si zachovávaly konkurenceschopný výkon při výrazně nižších paměťových nárocích. Výsledky potvrzují, že i lehké modely mohou být vhodné pro úlohu detekce změn, pokud jsou správně optimalizovány.

## KLÍČOVÁ SLOVA

Družicové snímky, dálkový průzkum Země, detekce změn, umělá inteligence, hluboké učení, neuronová síť, analýza obrazových dat

## ABSTRACT

Earth observation plays a crucial role in various fields such as urban development or environmental monitoring. Fast and accurate interpretation of satellite imagery is essential for real-time decision-making as well as for long-term planning. Traditional data processing methods often lead to delays, high costs, and low efficiency. These limitations pose a significant obstacle in situations where rapid response is critical. This thesis focuses on the design and implementation of a system for automatic change detection in satellite imagery using artificial intelligence and deep learning techniques. The emphasis is placed on improving computational efficiency, which is essential for processing large volumes of data in resource-constrained environments, such as directly onboard satellites. The outcome of the work is a set of three models, evaluated using standard performance metrics. The results show that the highest value of Intersection over Union (IoU) was achieved by the U-net model (0.7794). More compact architectures based on MobileNetV2 exhibited lower IoU scores (0.7090 and 0.7383, respectively), but maintained competitive performance with significantly reduced memory requirements. These findings demonstrate that lightweight models can be suitable for change detection tasks when properly optimized.

## KEYWORDS

Satellite imagery, remote sensing, change detection, artificial intelligence, deep learning, neural network, image data analysis

PECHÁČEK, Vilém. *Systém detekce změn událostí ze družicových dat pomocí umělé inteligence*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedoucí práce: Ing. Ján Mucha, PhD.

# Prohlášení autora o původnosti díla

**Jméno a příjmení autora:** Bc. Vilém Pecháček  
**VUT ID autora:** 230626  
**Typ práce:** Diplomová práce  
**Akademický rok:** 2024/25  
**Téma závěrečné práce:** Systém detekce změn událostí ze družicových dat pomocí umělé inteligence

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\* Autor podepisuje pouze v tištěné verzi.

## PROHLÁŠENÍ O VYUŽITÍ GENERATIVNÍ UMĚLÉ INTELIGENCE

Čestně prohlašuji, že jsem při zpracování diplomové práce s názvem Systém detekce změn událostí ze družicových dat pomocí umělé inteligence využil/a nástroje generativní umělé inteligence (AI). Toto využití, jeho rozsah a účel byly konzultovány s vedoucím mé práce. Nástroje AI sloužily jako podpora v rámci řešerše, konkrétně pro vyhledávání informací a orientaci ve zdrojích, a dále při formálních a stylistických úpravách mnou koncipovaného textu.

Veškeré informace a podněty získané s pomocí AI byly mnou kriticky posouzeny, ověřeny z dalších zdrojů a samostatně zpracovány. Finální text je mým původním dílem a neobsahuje pasáže přímo převzaté z výstupů AI.

Prohlašuji, že jsem postupoval/a v souladu s doporučeními VUT pro využívání generativní AI, zejména co se týče její přípustné asistenční role a zákazu přímého kopírování výstupů. Nesu plnou odpovědnost za obsah, původnost a kvalitu odevzdané práce.

Brno .....

podpis autora\* .....

---

\*Autor podepisuje pouze v tištěné verzi.

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Jánmu Muchovi, Ph.D., a konzultantovi Bc. Radoslavu Pitoňákovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Rád bych poděkoval také svým rodičům za jejich podporu během mých studií.

# Obsah

Úvod	12
<b>1 Zpracování družicových snímků</b>	<b>13</b>
1.1 Družicové systémy pro pozorování Země	13
1.1.1 Nanodružice	14
1.2 Družicové snímky	14
1.2.1 Optické RS snímky	15
1.2.2 SAR snímky	16
1.3 Segmentace	17
1.4 Detekce objektu	18
1.5 Regresivní analýza	20
1.6 Odstranění oblačnosti	21
<b>2 Detekce změn v družicových snímcích pomocí neuronových sítí</b>	<b>22</b>
2.1 Motivace detekce změn	22
2.2 Princip detekce změn	23
2.3 Neuronové sítě	24
2.4 Základní modely neuronových sítí	27
2.4.1 Konvoluční neuronové sítě	27
2.4.2 Autoenkodéry	28
2.4.3 Transformery	29
2.4.4 Rekurentní neuronové sítě	30
2.4.5 Generativní adversariální sítě	30
2.5 Lehké neuronové sítě	32
2.5.1 Ořezávání	33
2.5.2 Kvantizace	33
2.5.3 Distilace znalostí	34
2.5.4 Hledání architektury neuronových sítí	34
2.6 Veřejně dostupné datasety pro detekci změn	35
2.7 Metody hlubokého učení pro detekci změn s učitelem	38
2.7.1 Metody založené na CNN	38
2.7.2 Metody založené na transformerech	39
2.8 Metody hlubokého učení pro detekci změn s částečným učitelem a bez učitele	40
2.8.1 Metody založené na adversariálním učení	42
2.8.2 Metody založené na vlastním tréninku	42
2.8.3 Metody založené na regularizaci konzistence	43

2.8.4	Metody bez učitele . . . . .	44
2.9	Evaluační metriky . . . . .	45
<b>3</b>	<b>Implementace</b>	<b>47</b>
3.1	Předzpracování dat . . . . .	47
3.2	Augmentace dat . . . . .	51
3.3	Statistický popis dat . . . . .	53
3.4	Návrh modelu . . . . .	56
3.4.1	Kodér . . . . .	58
3.4.2	Modul pozornosti konvolučního bloku . . . . .	59
3.4.3	Absolutní rozdíl . . . . .	61
3.4.4	Dekodér . . . . .	62
3.5	Trénování modelu . . . . .	62
3.5.1	První iterace . . . . .	63
3.5.2	Druhá iterace . . . . .	64
3.6	Optimalizace a zjednodušení architektury . . . . .	67
3.7	Optimalizace hyperparametrů . . . . .	72
3.8	Srovnání predikce modelů . . . . .	74
	<b>Závěr</b>	<b>77</b>
	<b>Literatura</b>	<b>78</b>
	<b>Seznam symbolů a zkratk</b>	<b>85</b>
	<b>Seznam příloh</b>	<b>87</b>
<b>A</b>	<b>Obsah elektronické přílohy</b>	<b>88</b>

# Seznam obrázků

2.1	Princip detekce změn . . . . .	23
2.2	Příklad neuronové sítě . . . . .	25
2.3	Struktura transformeru . . . . .	31
2.4	Struktura CNN sítě využívající architekturu Siamských sítí . . . . .	39
2.5	Struktura modelu využívající architekturu Transformerů . . . . .	41
2.6	Struktura modelu využívající architekturu GAN . . . . .	43
2.7	Hypotéza regularizace konzistence . . . . .	44
3.1	Princip předzpracování vstupních dat. . . . .	48
3.2	Porovnání obálek histogramů . . . . .	55
3.3	Podíl masek v datasetu . . . . .	56
3.4	Blokové schéma modelu s architekturou U-net . . . . .	57
3.5	Detekce změn modelu s architekturou U-net . . . . .	66
3.6	Blokové schéma modelu s MobileNetV2 kódérem . . . . .	69
3.7	Detekce změn modelu s MobileNetV2 kódérem . . . . .	71
3.8	Srovnání predikce modelů . . . . .	76

# Seznam tabulek

2.1	Aktivační funkce a jejich deriváty. Převzato z [30]. . . . .	26
2.2	Veřejně dostupné datasety pro detekci změn. Převzato z [37]. . . . .	36
3.1	Globální statistické hodnoty . . . . .	54
3.2	Přehled hyperparametrů použitých při trénování modelu . . . . .	63
3.3	Validační ztráta pro různé kombinace rychlosti učení a útlumu vah . .	73
3.4	Srovnání modelů . . . . .	73

# Úvod

Dálkový průzkum země je technickým odvětvím, které za poslední dekádu zaznamenalo významný pokrok. Ten je poháněn zejména rychlými inovacemi v oblasti mikroelektroniky a miniaturizace, která vedla k vývoji tzv. nanodružic. Tyto cenově dostupné družice umožnily širší přístup k vesmírným misím, což přineslo revoluci ve sběru dat o naší planetě a společnosti [10, 12].

Nižší náklady a miniaturizace zpřístupnily oběžnou dráhu soukromým vědeckým a vzdělávacím institucím. Nanodružice vytvářejí rozsáhlé konstelace, které poskytují data s vyšším prostorovým, časovým i spektrálním rozlišením. Díky tomu lze efektivně monitorovat klíčové procesy v reálném čase, jako jsou změny klimatu, zemědělské aktivity nebo urbanizace [10].

Jednou z klíčových oblastí dálkového průzkumu je detekce změn. Tato disciplína se zaměřuje na analýzu časových řad snímků s cílem identifikovat, klasifikovat a kvantifikovat změny, ke kterým na planetě dochází [9, 25].

Moderní metody detekce změn kombinují data z různých typů senzorů, jako jsou optické nebo radarové, což umožňuje přesnější analýzu i za nepříznivých meteorologických nebo světelných podmínek. Pokročilé algoritmy využívající strojové učení a hluboké neuronové sítě pak dokážou efektivně zpracovat velké objemy dat a odhalit jemné změny, které by jinak zůstaly nepovšimnuty [9].

Cílem diplomové práce je provedení rešerše a popsání technologií dálkového průzkumu s důrazem na detekci změn. Praktická část práce se zabývá návrhem a následnou implementací modelu hlubokého učení pro detekci změn.

Práce je členěna do tří kapitol. První kapitola se zabývá problematikou pořizování a zpracovávání družicových snímků. Popisuje základní metody analýzy těchto snímků. Druhá kapitola se již zabývá detekcí změn na družicových snímcích s využitím neuronových sítí a hlubokého učení. Jsou zde popsány důvody a princip detekce změn, dále jsou popsány základní modely neuronových sítí a problematika tzv. lehkých neuronových sítí. Je zde také provedena rešerše aktuálně dostupných datasetů pro učení modelů detekce změn. Následně jsou popsány přístupy učení s učitelem, s částečným učitelem a bez učitele. Nakonec jsou popsány evaluační metriky používané pro posouzení přesnosti modelu. Třetí kapitola je kapitolou praktické části diplomové práce, ve které je popsán samotný návrh neuronové sítě a jeho učení na zvoleném datasetu, který je předzpracován a statisticky popsán. Navržená architektura je pro dosažení lehkého modelu optimalizována. Každá iterace návrhu je evaluována pomocí standardních metrik. Nakonec je provedeno srovnání predikce jednotlivých návrhů na reálných družicových datech a výsledky jsou diskutovány. V závěru jsou shrnuty veškeré výsledky dosažené v diplomové práci.

# 1 Zpracování družicových snímků

## 1.1 Družicové systémy pro pozorování Země

Družicové systémy pro pozorování Země (earth observation, EO) jsou družice určené pro pozorování jevů na zemském povrchu. K tomuto účelu jsou vybaveny různými senzory, jejichž technologie se liší v závislosti na úkolu, pro jehož plnění byla družice navržena (např. pozorování přírodních jevů, sledování katastrof nebo monitorování lidské činnosti). EO družice pomocí těchto senzorů shromažďují různá data. U dat ve formě snímků můžeme EO senzory rozdělit na optické (viditelné a infračervené spektrum, LIDAR) a mikrovlnné (SAR). Ve vztahu k detekci změn jsou typy snímků blíže popsány v části 1.2 [1].

Dále lze EO družice dělit podle typu oběžné dráhy, na které operují. Např. pro EO družice sledující počasí je vhodné, aby byly na geostacionární oběžné dráze (přibližně 36 000 km nad Zemí), na níž jsou schopny nepřetržitě sledovat celou polokouli. V takové výšce je však obtížné dosáhnout uspokojivého prostorového rozlišení [2]. Z těchto důvodů je většina EO družic provozována v relativně nízkých výškách okolo 500 km. Nižší výšky už nejsou vhodné kvůli vzrůstajícímu aerodynamickému tření, což vyžaduje provádění manévru pro udržení oběžné dráhy [3].

Mezi klíčové EO družice lze zařadit Landsat, Sentinel, MODIS a WorldView. Počátky Landsatu sahají až do roku 1972 a jedná se o jeden z nejdéle trvajících EO programů. Nejnovější družice v této řadě jsou Landsat 8 a 9, které poskytují multispektrální data s rozlišením 15–30 m [4, 5].

Série družic Sentinel je součástí programu Evropské unie Copernicus, kdy každá družice má specifický účel. Sentinel-1 je vybaven SAR (viz 1.2.2), Sentinel-2 poskytuje multispektrální optická data (viz 1.2.1) a Sentinel-3 je specializovaný na monitorování oceánů a je schopen např. měřit jeho teplotu [4, 6].

MODIS (Moderate Resolution Imaging Spectroradiometer) je přístroj umístěný na družicích mise Terra a Aqua. Družice s tímto systémem se používají pro globální monitorování změn v ekosystémech, počasí nebo k vyhodnocování lesních požárů. Poskytují data v širokém spektru s rozlišením 250–1000 m [4, 7].

Družice WorldView společnosti Maxar Technologies dokáží poskytovat velmi detailní optická data s rozlišením až 30 cm. Tyto družice jsou nejčastěji používány k urbanistickému plánování, krizovému managementu nebo pro vojenské účely [4, 8].

S rozvojem družicových technologií a neustále se zvyšujícím počtu družic na oběžné dráze narůstá i objem vícezdrojových dat pro vzdálené snímání (Remote Sensing, RS), které produkují [9]. V roce 2008 se na oběžné dráze nacházelo 150 pozorovacích družic pořizujících 10 terabitů dat denně [12]. Rokem 2021 byl tento počet již na 971 družic a nadále rychlým tempem roste [14]. Jen mezi roky 2024

až 2033 se počítá s vypuštěním 5 401 nových družic pro pozorování země [15]. To klade značné požadavky na aplikace monitorující změny na zemském povrchu, a to zejména v oblasti jejich automatizace, autonomie a robustnosti [9].

### 1.1.1 Nanodružice

Nanodružice jsou rozměrově malé družice vážící obvykle 1–10 kg, jejichž hlavní výhodou jsou nižší pořizovací náklady oproti standardním družicím. Díky tomu jsou vyvíjeny jak univerzitami, tak soukromými a státními vesmírnými agenturami. Nanodružice jsou na oběžnou dráhu Země vypouštěny ve velkých skupinách tvořících tzv. konstelaci. To umožňuje kombinaci vysokého časového a prostorového rozlišení a frekventované odesílání dat na pozemní stanice. Počet konstelací se navíc každým rokem zvyšuje, kdy k letošnímu roku obíhá Země přes 4100 nanodružic [10, 11].

Poslední dekáda znamenala zásadní změnu v trendech vývoje družicových systémů. S nástupem standardu CubeSat (formát 10x10x10 cm) a jeho flexibilitou, spolehlivostí a cenovou dostupností se otevřely nové možnosti vývoje vesmírných misí, a to zejména pro soukromé subjekty. S technologickým pokrokem v mikroelektronice a mikrosystémech se snižuje spotřeba energie, velikost a hmotnost nanodružic a tím klesá i jejich cena. Vypuštění jedné nanodružice stojí okolo 1 milionu dolarů, což je ve výrazném kontrastu s cenami tradičních družic jako Landsat nebo Sentinel, jejichž vývoj vyšel na 855 a 300 milionů dolarů [10].

Mezi nejvyužívanější konstelace nanodružic pro aplikace dálkového průzkumu země lze zařadit projekt PlanetScope provozovaný americkou společností Planet Labs. Jedná se o konstelaci více než 430 cubesatů, jež je schopná každodenně nasnímat celý povrch Země. Konstelace se pohybuje na orbitu ve výšce 475–525 km a denně je schopna nasnímat území o velikosti 200 milionů km<sup>2</sup>. PlanetScope je hojně využíván v soukromých projektech pozorování zemského povrchu a byl již úspěšně využit např. při oceánografii nebo studiu vegetace [10, 13].

## 1.2 Družicové snímky

V oblasti detekce změn na zemském povrchu se nejčastěji používají dva typy dat: optické RS a SAR snímky; které budou blíže popsány v podkapitolách 1.2.1 a 1.2.2. Tyto typy snímků jsou shromažďovány pasivními i aktivními senzory, které pokrývají různá elektromagnetická spektra. Jejich použití lze v rámci dálkového průzkumu země doplnit o další zdroje dat, jako jsou digitální výškové modely (Digital Elevation Model, DEM), data geografického informačního systému (Geographic Information System, GIS), data z mračen bodů (point cloud) nebo streetview snímky [9].

### 1.2.1 Optické RS snímky

Optické RS snímky lze rozdělit na RGB, hyperspektrální, multispektrální a panchromatické dle počtu obsažených pásem. Hyperspektrální snímky jsou obrazové objemové krychle skládající se ze stovek spektrálních pásem. Obsahují mnoho úzkých pásem napříč elektromagnetickým spektrem s obvyklou šířkou menší než 10 nm. Multispektrální snímky mají obvykle do 15 pásem (definice počtu kanálů multispektra není do dneška v odborné literatuře přesně stanovena). Spektrální rozlišení se pohybuje v 0,1 násobku vlnových délek [9].

Panchromatický obraz má pouze jedno pásmo (tj. černo-bílé pásmo) s šířkou pásma obvykle ve stovkách nm. Šířka pásma umožňuje udržet vysoký odstup signálu od šumu, což umožňuje vysoké a velmi vysoké prostorové rozlišení panchromatických dat. Z tohoto důvodu se obvykle panchromatické a multispektrální obrazy spojují, čímž je docíleno bohatších spektrálních a prostorových informací pro detekci změn [9, 15].

Hyperspektrální snímky mají stovky až tisíce úzkých pásem, díky kterým poskytují bohaté spektrální a prostorové informace. Mají velký význam při rozlišování jemných změn na zemském povrchu. Detailní informace o změnách jednotlivých spekter vytváří robustní zdroj pro detekci změn. Nevýhodou je větší redundance dat a nízké prostorové rozlišení, což způsobuje nejasnost textur okolo pixelů a smíšené pixely zabírají velkou část dat. Systémy detekce změn, které využívají hyperspektrální snímky, musí řešit problém smíšených pixelů, vysokých výpočetních nároků a limitovanou velikost trénovacího datasetu. K adresaci těchto problémů lze použít efektivní algoritmy umělé inteligence (Artificial Intelligence, AI) [9].

Multispektrální snímky lze pořizovat s vysokým a velmi vysokým prostorovým rozlišením (10 až 100 cm/pixel), s bohatými barvami a texturami. Obrázky s vysokým prostorovým rozlišením mohou podávat informace o struktuře pozemních objektů, díky čemuž jsou široce používány pro detekci změn. Nejběžněji se používají družicové snímky družic Landsat a Sentinel vzhledem k jejich nízkým pořizovacím nákladům. Mezi další družice pořizující snímky s vysokým rozlišením patří např. Quickbird, SPOT, Gaofen nebo Worldview [9].

Optické RS obrazy jsou široce využívány pro detekci změn, jelikož poskytují bohaté spektrální a prostorové informace. Jejich nevýhodou však je, že jejich optické senzory spoléhají na sluneční osvětlení a použitá vlnová délka se blíží viditelnému světlu. Z těchto důvodů jsou často ovlivňovány slunečním zářením a mraky. V následující podkapitole budou probrány SAR snímky, které využívají vlnovou délku 1–100 cm a mají vlastní zdroj osvětlení. Dokážou tedy snímat ve dne i v noci a téměř za všech povětrnostních podmínek [9].

## 1.2.2 SAR snímky

Technika SAR (Synthetic Aperture Radar) vysílá mikrovlnné pulzy směrem k zemskému povrchu. Tato rozptýlená vysokofrekvenční mikrovlnná energie je následně vypočtena a je z ní sestaven snímek. SAR funguje na senzorech o vlnových délkách 1-100 cm. Na rozdíl od kamerových senzorů využívajících vlnové délky blíže k viditelnému spektru, dokáže SAR proniknout čistou tmou, oblačností a nepříznivým počasím, což patří mezi jeho hlavní výhody [17].

Princip fungování SAR je takový, že radar vypočítává čas, za který rádiová vlna opustí anténu, odrazí se od zemského povrchu a urazí vzdálenost zpět k počátku. Když radarové vlny dopadnou na zemský povrch, podíl energie rozptýlené do senzoru závisí na mnoha faktorech jako vlhkost, drsnost a sklon povrchu nebo frekvence, polarizace a úhel dopadu. K definování obrazu pomocí SAR je potřeba velkého množství dat a značný výpočetní výkon. SAR také vyžaduje velmi dlouhou anténu, která se vytváří uměle pomocí pohybu antény na přijímači tak, aby pokryla co největší plochu. Vytváření snímku z přijatých dat je komplexní proces [17].

SAR je ceněným nástrojem pro dálkový průzkum země, který je využíván v civilní i vojenské sféře. Vojenské aplikace SAR zahrnují shromažďování zpravodajských informací, průzkum nebo dohled nad oblastí. V civilním odvětví slouží jako nástroj pro monitorování zemědělství, topografické plánování, geologie, oceánografie apod. [17]

Metody klasifikace scén na družicových snímcích (Remote Sensing Image Classification, RSISC) se zaměřují na identifikaci a kategorizaci obsahových prvků. To je nezbytné v odvětvích jako je např. environmentální monitoring, městské plánování nebo zemědělství. Vývoj RSISC prošel několika etapami: od klasifikace na úrovni pixelů k analýze na úrovni objektů a později až k metodám na úrovni celé scény. Tento pokrok byl podpořen zejména rozvojem hlubokého učení, které se osvědčilo v automatické extrakci relevantních znaků přímo z obrazových dat, což eliminovalo potřebu ruční přípravy vstupů [18].

Zásadní technologií v klasifikaci scén představují konvoluční neuronové sítě (Convolutional Neural Network, CNN, viz část 2.4.1), a to díky schopnosti efektivně extrahovat prostorové a kontextuální informace z obrazu. CNN jsou schopné rozpoznávat i složitější vzory, což z nich činí klíčový nástroj pro RSISC. V praxi existují tři hlavní přístupy k využití CNN:

- **Předtrénované modely jako extraktory znaků** spočívá v aplikaci již naučených váhových parametrů pro extrakci obecných vlastností, což redukuje potřebu velkých datasetů. Tento přístup je efektivní zejména při použití předem natrénovaných modelů [18].
- **Jemné doladění (fine-tuning)** předtrénovaných modelů je přístup, při kte-

rém jsou váhy modelu částečně upraveny pro konkrétní úlohy v rámci dálkového průzkumu, což zvyšuje přesnost klasifikace. To je sice náročnější na data a výpočetní výkon, na druhou stranu to však zajišťuje lepší výsledky v případě specifických scénářů [18].

- **Trénink od nuly** je ideální volbou, pokud jsou k dispozici rozsáhlé a různorodé tréninkové datasey. Výhodou tohoto přístupu je, že model je přizpůsoben specifickým požadavkům a může dosáhnout vyšší přesnosti, nevýhodou je však jeho výpočetní a datová náročnost [18].

Kromě CNN existují i alternativní přístupy k řešení RSISC, a to zejména autoenkodéry (viz 2.4.2) a generativní adversariální sítě (Generative Adversarial Network, GAN, viz 2.4.5). Autoenkodéry se využívají zejména pro redukci dimenzionality, která se používá pro zjednodušení práce s velkými datovými sadami a také poskytuje skryté reprezentace dat bez nutnosti anotací. GAN jsou zase užitečné pro přidání realistických syntetických dat, která se použijí pro trénink modelu [18].

Mezi novější přístupy v RSISC patří např. modely založené na pozornosti (attention-based models) a transformery (viz 2.4.3), které se také osvědčují v přesné klasifikaci scén. Modely založené na pozornosti soustředí „pozornost“ modelu na důležité oblasti obrazu, čímž dosahují detailnější analýzy specifických částí scény. Transformery jsou sice v dálkovém průzkumu méně běžné, avšak přinášejí s sebou schopnost učit se globální vzory, což zvyšuje jejich efektivitu v případech, kdy jsou vztahy mezi vzdálenými částmi obrazu zásadní. Transformery tak mohou mít klíčový význam pro aplikace vyžadující detailní analýzu komplexních struktur [18].

Mezi hlavní problémy metod RSISC patří vysoká závislost na anotovaných datech, riziko přetrénování modelu nebo omezená generalizace modelů na reálná data z různých podmínek. To omezuje přesnost a přenositelnost modelů RSISC na různé geografické regiony a environmentální podmínky. Budoucí výzkum v této oblasti se proto bude věnovat nejpravděpodobněji na metody fúze znaků, které mohou obohatit modely o nové reprezentace, a na techniky zvyšující interpretovatelnost, aby bylo snazší modely pochopit a aplikovat je v praxi [18].

## 1.3 Segmentace

Segmentace družicových snímků je proces klasifikace jednotlivých pixelů obrazu do různých kategorií tak, aby byly rozpoznány a vizuálně odděleny konkrétní objekty a oblasti zájmu (např. vodní plochy, vegetace, budovy). Tento proces slouží pro detailní analýzu velkých obrazových dat a je široce používán v odvětvích jako je urbanizace, monitorování vegetace, krizové řízení apod [19].

Segmentace družicových snímků je formou tzv. semantické segmentace, kdy se každý pixel obrazu přiřazuje určité třídě na základě předem definovaných kategorií.

Cílem je zvýšení schopnosti modelů správně identifikovat a klasifikovat konkrétní vzory ve snímcích, a to i v případech, kdy mají některé oblasti rozdílnou barvu a tvar [19].

Dřívější metody klasifikace družicových snímků, mezi které patřilo např. prahování nebo klastrovací algoritmy typu k-means, trpěly omezenou přesností. A to především při práci s rozsáhlými a složitými datovými sadami. Po nástupu neuronových sítí se výkon segmentačních úloh výrazně zlepšil díky automatické extrakci složitých příznaků z obrazových dat [19].

Moderní metody segmentace družicových snímků staví převážně na architekturách CNN, které jsou navrženy speciálně pro práci s obrazovými daty. Jedná se o hierarchickou strukturu vrstev, která umožňuje extrahovat stále komplexnější příznaky. Mezi populární architektury patří plně konvoluční sítě (Fully Convolutional Networks, FCN), U-Net a pyramidového zpracování scény (Pyramid Scene Parsing Network, PSPNet) [19].

FCN zajišťují plnou segmentaci obrazu v tréninku typu end-to-end a jsou základem většiny moderních segmentačních metod. U-Net je rozšířením FCN, které zahrnuje dekonvoluční vrstvy pro zachování detailů a přesnou rekonstrukci původní velikosti obrazu. U-Net se osvědčil zejména při segmentaci malých a složitých struktur. PSPNet pak využívá pyramidovou strukturu pro zachycení vícerozměrných kontextových informací, což umožňuje lepší rozpoznávání globálních vzorů v obrazových datech [19].

Mezi hlavní výzvy při řešení segmentačních úloh patří zejména potřeba velkého množství označených dat, které jsou nutné pro trénink. Družicové snímky zachycují různé oblasti na Zemi a obsahují rozmanité typy povrchů a klimatických podmínek. Segmentační modely proto musí být robustní a schopné dobře se přizpůsobit těmto rozdílům. Také je nutné mít na paměti vysoké nároky na výpočetní výkon, jelikož družicové snímky využívané pro segmentaci mají zpravidla vysoké rozlišení a jsou zpracovávány ve velkých objemech [19].

## 1.4 Detekce objektu

Detekce objektů na družicových snímcích má zásadní roli v mnoha oblastech dálkového průzkumu, od monitoringu životního prostředí, přes správu přírodních zdrojů až po krizové řízení. Detekce zahrnuje proces identifikace a klasifikace objektů, které se na snímcích vyskytují. Cílem je nejen lokalizovat tyto objekty, ale i popsat jejich typ a potenciálně i stav. Výzvy spojené s tímto procesem vycházejí ze specifík družicových dat, ve kterých se vyskytuje široká škála objektů různé velikosti, rozlišení a kontextu. Dalším faktorem je vysoká hustota informací a složitost pozadí, která může způsobit, že různé objekty mají podobné vizuální charakteristiky [20].

Jednou z hlavních výzev je variabilita velikosti a tvaru objektů. Na satelitních snímcích lze najít jak malé objekty (např. vozidla), tak i rozsáhlé oblasti (např. lesní porost nebo vodní plochy), které mají odlišné prostorové i spektrální vlastnosti. Modely hlubokého učení musí být schopné tyto objekty rozlišit a správně klasifikovat, a to bez ohledu na jejich velikost nebo rozlišení [20].

Další výzvou je komplexní pozadí družicových snímků. Klasifikace objektů může být obtížná, jelikož tyto objekty často sdílejí podobné barvy nebo textury s jinými prvky obrazu, což detekci ztěžuje. Kromě toho objekty mohou být částečně překryté vegetací nebo stíny, což je pro přesnou segmentaci a identifikaci problém [20].

V neposlední řadě je tu omezená dostupnost anotovaných dat. Pro trénink modelů hlubokého učení jsou rozsáhlá a kvalitně anotovaná data klíčová. Sbíráni a anotace těchto dat je časově náročná a finančně nákladná, což omezuje možnosti rozsáhlého nasazení pokročilých modelů [20].

V souvislosti s těmito výzvami se mezi modely hlubokého učení prosadily zejména CNN. Ty totiž mají výborné výsledky při extrahování komplexních vizuálních příznaků z obrazových dat. Dvě nejpoužívanější architektury pro detekci objektů jsou Faster R-CNN a You Only Look Once (YOLO) [20].

Faster R-CNN používá dvoufázový přístup, kdy nejprve generuje regiony zájmu, ve kterých by se objekty mohly vyskytovat, a poté každý z těchto regionů klasifikuje. Tato metoda poskytuje vysokou přesnost detekce, ale kvůli složitosti celého procesu může být pomalejší, což omezuje její použití v aplikacích vyžadujících rychlé reakce [20].

YOLO naopak provádí detekci objektů v jednom kroku. To dovoluje rychlejší zpracování, avšak má za následek menší přesnost a zvýšené riziko vynechání menších objektů. Tato architektura je však ideální pro aplikace vyžadující rychlou detekci v reálném čase [20].

Pro zvýšení přesnosti se v rámci detekce objektů používají různé techniky:

- **Učení přenosem znalostí (transfer learning)** používá modely, které byly předtrénovány na rozsáhlých obrazových datových sadách a následně doladěny na menší sadě. Tento postup šetří čas potřebný na trénink a zároveň zlepšuje výslednou přesnost detekce [20].
- **Augmentace dat** pomáhá rozšířit datovou sadu o nové data pomocí úprav na stávajících datech. A to pomocí rotace, změny měřítka, převrácení nebo úpravě osvětlení. Tento přístup pomáhá modelu učit se různé variace objektů a tím zvyšuje jeho robustnost vůči odlišným podmínkám v satelitních datech [20].
- **Víceměřítkové učení (multi-scale learning)** je zahrnuto v některých pokročilejších architekturách. Tento přístup zahrnuje zpracování obrazu na různých měřítkách, což pomáhá modelu lépe detekovat malé i velké objekty, a to

i v případech, kdy jsou objekty výrazně zmenšeny nebo naopak zvětšeny [20].

## 1.5 Regresivní analýza

Regresivní analýza je jedna z důležitých metod v oblasti statistického strojového učení aplikovaného na data z dálkového průzkumu Země. Cílem regrese je kvantifikování vztahu mezi nezávislými proměnnými (např. spektrálními pásmy družicového snímku) a závislou proměnnou, která představuje odhadovanou hodnotu (např. hustotu vegetace, teplota povrchu). Použité metody zahrnují jednoduchou lineární regresi, logistickou regresi a pokročilejší metody, jako jsou podporované regresivní vektory (Support Vector Regression, SVR) a hluboké učení [21].

Lineární regrese je jednou z nejjednodušších a nejčastěji používaných metod v analýze družicových dat. Metoda modeluje lineární vztah mezi vysvětlujícími proměnnými (družicovými pásmy) a odpovědní proměnnou (např. teplota povrchu). Nevýhodou lineární regrese je omezená schopnost zachytit nelineární vztahy v datech. V praxi se proto často používají nelineární regresivní modely, jako polynomiální regrese nebo modely založené na neuronových sítích [21].

Mezi pokročilé regresivní metody patří již zmíněné SVR a neuronové sítě. SVR je populární volbou pro analýzu družicových dat, jelikož dokáže pracovat s vysokorozměrnými datovými sadami a zvládá efektivně modelovat nelineární vztahy pomocí jaderných funkcí. Výhodou neuronových sítí (zejména CNN), je jejich schopnost automaticky extrahovat komplexní vzory z obrazových dat. To je při modelování nelineárních vztahů mezi spektrálními pásmy a environmentálními parametry klíčové [21].

Regresivní modely se často používají např. při odhadech zemědělské produkce, kde jsou proměnné jako normalizovaný rozdílový vegetační index (Normalized Difference Vegetation Index, NDVI) použity jako vstupy pro odhad výnosu pšenice, kukuřice, apod. Mezi další aplikace lze zahrnout např. monitoring kvality vody, kde se spektrální data používají k odhadu obsahu chlorofylu nebo suspendovaných pevných částic. V oblasti sledování změn pokryvu půdy se regrese používá k predikci rozlohy lesních ploch nebo detekci změn v rozloze vodních ploch [21].

Mezi hlavní výhody regresivních metod patří jejich interpretovatelnost a schopnost poskytnout kvantitativní odhady s mírou nejistoty. To umožňuje vyhodnotit, jak přesné jsou predikce modelu a které faktory mají největší vliv na odpovědní proměnnou. Na druhé straně jsou však tyto modely často citlivé na kvalitu a rozsah tréninkových dat, což omezuje jejich schopnost zobecňovat se na nové oblasti nebo na snímky s odlišnými charakteristikami. Proto je nutné provádět křížovou validaci a další doprovodné metody hodnocení, aby riziko přeučení modelu bylo co nejnižší [21].

## 1.6 Odstranění oblačnosti

Odstraňování oblačnosti je důležitou součástí předzpracování dat u družicových snímků. Oblačnost totiž může značně ovlivnit kvalitu a použitelnost získaných dat. Přítomnost mraků často zakrývá pozorovaný povrch a způsobuje problémy při analýze změn, klasifikaci nebo odhadech environmentálních parametrů. Z těchto důvodů se vyvinuly různé techniky a přístupy k identifikaci a odstranění oblačnosti, které umožňují získat čistější a kvalitnější snímky pro další analýzu [22].

Mezi tradiční metody patří prahové metody, časové diference a statistické přístupy. Prahové metody využívají viditelnou reflexi a infračervené teploty mraků k odhalení oblastí s oblačností. Jejich výkon však může být snížen na snímcích s nízkým kontrastem mezi mraky a povrchem. Časová diference využívá vícečasové snímky k identifikaci změn v hodnotách pixelů, které odpovídají přítomnosti mraků. Tento přístup však nereflktuje změny způsobené jinými faktory, jako jsou záplavy nebo změny ve vegetaci. Statistické metody pak kombinují spektrální a prostorové charakteristiky extrahované z obrazu pomocí algoritmů jako je stroj podpůrných vektorů (Support Vector Machine, SVM) nebo rozhodovací stromy. Tento přístup má často problémy s generalizací a jsou náchylné k chybám způsobeným různými typy pozadí [22].

V poslední době se pro efektivní detekci oblačnosti stále více využívají CNN a další metody hlubokého učení. Tyto metody poskytují state-of-the-art výsledky při klasifikaci a segmentaci z obrázků, a to díky schopnosti automaticky extrahovat komplexní vzory z dat [22].

Metody detekce oblačnosti také často využívají multispektrální nebo hyperspektrální snímky. Například techniky založené na multispektrálních snímcích kombinují různé pásma viditelného a infračerveného spektra, což umožňuje lépe rozlišit mezi oblačností a jinými typy povrchů, jako je sníh nebo voda, které mají podobné vizuální charakteristiky [22].

CNN mohou být nasazeny na hardwarově omezených systémech, jako jsou programovatelná hradlová pole (Field Programmable Gate Array, FPGA) nebo ARM procesory. To umožňuje detekci oblačnosti v reálném čase na palubě družice. Tento přístup zvyšuje rychlost zpracování, ale také snižuje potřebu přenosu velkých objemů dat na pozemní stanice a šetří tím přenosovou kapacitu [22].

Použití metod hlubokého učení přináší řadu výhod, jako vyšší přesnost nebo schopnost modelovat složité a nelineární vztahy mezi spektrálními pásmy a charakteristikami oblačnosti. Tyto modely však vyžadují velké množství tréninkových dat a jsou náročné na výpočetní výkon, což může být omezující pro nasazení na družici. Další výzvou je správná detekce v podmínkách kde se oblačnost a povrch vizuálně překrývají (např. oblast pokrytá sněhem) [22].

## 2 Detekce změn v družicových snímcích pomocí neuronových sítí

### 2.1 Motivace detekce změn

S rozvojem technologií dálkového průzkumu země se výrazně zvýšily i možnosti sběru dat, která se stala klíčovým zdrojem pro monitorování detekce změn. Detekce změn se zaměřuje na identifikaci změn v konkrétní geografické oblasti porovnáním snímků pořízených v různých časových obdobích. Tato metoda získala velkou pozornost díky široké škále praktických využití, například při detekci požárů, sledování environmentálních změn, monitorování katastrof, analýze urbanistického rozvoje a správě krajiny [23].

Detekce změn je také hojně využívána ve vojenském a bezpečnostním odvětví. Vojenské aplikace zahrnují detekci přemístování vojenských jednotek, monitorování výstavby vojenských zařízení či hodnocení škod způsobených útoky. Moderní technologie, jako je SAR, poskytují přesné informace o terénních změnách i za špatných povětrnostních podmínek, což je kritické pro včasnou reakci (viz podkapitola 1.1.2) [25].

Detekce změn však s sebou přináší i technologické výzvy, kterým je potřeba čelit. Dvojice snímků používané pro detekci jsou často pořízeny za odlišných podmínek, jako jsou různé úhly, osvětlení nebo roční období. Tyto faktory mají za následek odlišnou kvalitu obrazu snímků projevující se v rozdílu prostorového, spektrálního a časového rozlišení nebo přítomností různých typů šumů a artefaktů [24].

Rychlý vývoj technologií hlubokého učení (zejména CNN) zaznamenaný v posledních letech vedl k rozvoji metod pro detekci změn. Umělé neuronové sítě se staly výkonově bezkonkurenční a zcela odsunuly původní tradiční metody pro detekci změn (např. dříve hojně používané bodové klasifikační metody) [24].

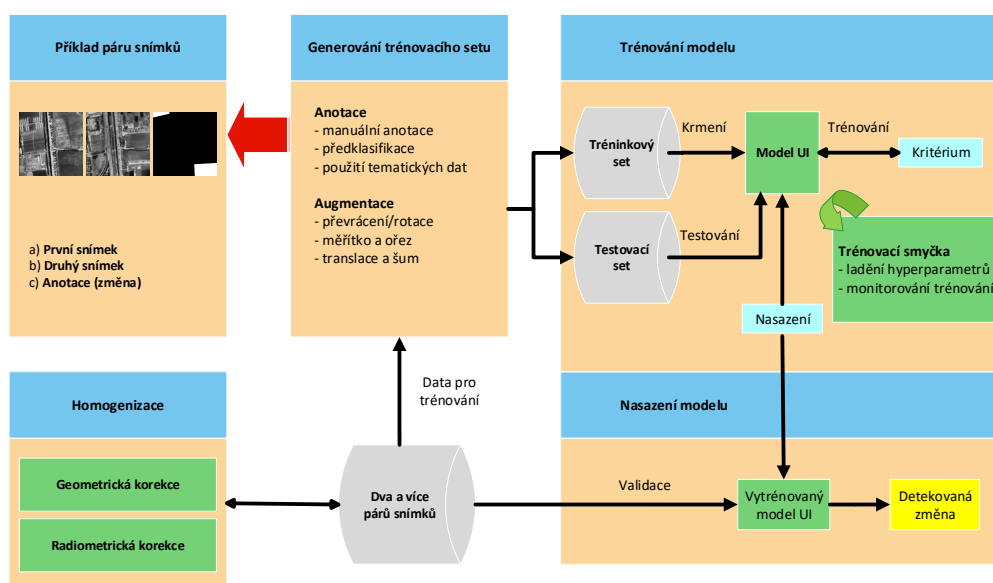
V současné době stojí technologie dálkového průzkumu země, mezi níž patří i detekce změn, před problémem rychle se zvyšujícího množství družic na oběžné dráze Země. Momentálně většina EO družic zasílá nezpracovaná data na Zemi, kde dochází ke zpracování. Omezená kapacita přenosu z vysílací stanice na zařízení a rychlost přenosu má za následek zpožděnou dostupnost dat a neefektivní využití pozemních stanic. To má nepříznivý dopad na časově citlivé aplikace, u kterých jsou data vyžadována s nízkou latencí. Zmíněné narůstání počtu družic a zvyšování rozlišení snímání bude tento problém v budoucnu nadále prohlubovat [26].

Současný pokrok v hardwaru a strojovém učení umožňuje hledat způsob, jak problém neefektivního využití sestupného kanálu vyřešit. V předchozích letech byly úspěšně nasazeny klasifikátory jako součást družice, využívané k segmentaci oblač-

nosti, povodní nebo detekci bouřek. Nevýhodou klasifikátorů s učitelem však je, že detekují pouze událost, na kterou byly natrénovány [26].

## 2.2 Princip detekce změn

Obecný přístup k řešení detekce změn na družicových snímcích za použití neuronových sítí se skládá z několika kroků. Před samotným trénováním modelu je nutné upravit a předzpracovat samotný dataset, který bude použit. Obecná architektura detekce změn je zobrazena na obr. 2.1, kdy jednotlivé kroky budou popsány níže. Používané modely neuronových sítí jsou popsány v kapitole 2.4 [9, 23].



Obr. 2.1: Princip detekce změn s využitím neuronových sítí. Převzato z [9].

Kvůli rozdílům v osvětlení scény, atmosférických podmínkách a ročnímu období v době pořízení snímku nebo v rozdílné poloze družice mezi snímky je obvykle potřeba nejprve data homogenizovat. K tomu se využívají dvě běžně používané metody, a to geometrická a radiometrická korekce [9].

Geometrická korekce má za cíl sladit dva nebo více párů snímků pomocí registrace nebo koregistrace. Srovnání mezi odpovídajícími pozicemi může být dosaženo pouze za předpokladu, že se dané pozice překrývají [9].

Radiometrická korekce má za úkol eliminovat rozdíly v osvětlení scény nebo odrazivosti způsobené procesem digitalizace a také minimalizaci zkreslení způsobeného

absorpcí a rozptylem v atmosféře. To pomáhá snížit falešné detekce v následném trénování modelu [9].

K vývoji modelu je zapotřebí mít velký a vysoce kvalitní trénovací set, který pomůže algoritmům zvoleného modelu pochopit, že určité vzorce náleží konkrétním výsledkům. Data, která vznikla za více období, jsou označena nebo anotována pomocí určitých technik (např. ruční anotace, předklasifikace, použití tématických dat), aby bylo pro model snadné naučit se charakteristiky změněných objektů [9].

V případě nedostatku trénovacích dat je vhodné provést široce používanou augmentaci. Jedná se o umělé zvýšení počtu dat pomocí horizontálního nebo vertikálního převrácení, rotace, změny měřítka, oříznutí, posunutí apod. Tento krok výrazně zvýší rozmanitost údajů dostupných v trénovacím setu [9].

Po vygenerování trénovacího setu ho lze obvykle rozdělit podle počtu vzorků a nebo geografické oblasti na trénovací set (pro trénování modelu) a testovací set (pro hodnocení přesnosti modelu). Proces učení a testování se provádějí střídavě a iterativně. Během trénovacího procesu je model optimalizován podle kritéria přesnosti, kterým může být chybová funkce [9].

Posledním krokem je samotné nasazení modelu v praktických aplikacích. To umožní ověření schopnosti zobecnění a robustnosti modelu, což je důležitý krok pro celkové zhodnocení zvoleného postupu detekce změn na bázi umělé inteligence [9].

## 2.3 Neuronové sítě

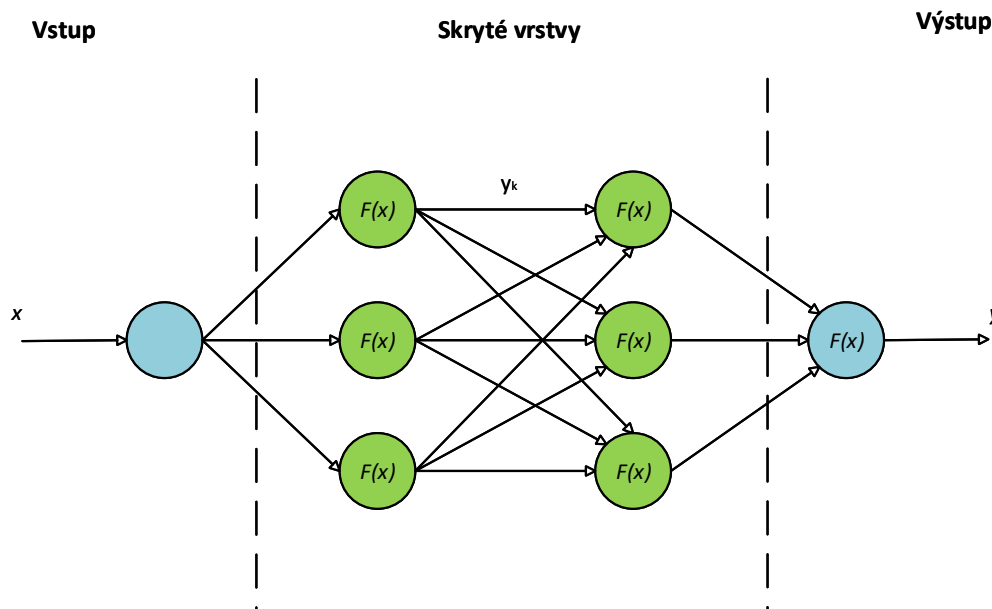
Neuronová síť je modelem strojového učení, který je odvozen od principu fungování lidského mozku. Stejně jako biologický mozek se i neuronové sítě skládají z velkého množství neuronů, které tvoří jednotlivé vrstvy modelu. A to vstupní vrstvu, jednu nebo více skrytých vrstev a výstup. Každý neuron je pak spojen se vstupy neuronů nadřazené vrstvy. Příklad struktury neuronové sítě viz obrázek 2.2 [27].

Každý neuron se skládá ze vstupů s rozdílnými váhami, prahu (bias) a výstupu. Pokud je výstup neuronu nad zadanou prahovou hodnotou, aktivuje se a odesílá data do další vrstvy sítě. Neuron lze popsat pomocí rovnice

$$u_k = b_k + \sum w_{kj}x_{kj}, \quad (2.1)$$

kde  $u_k$  je výstupní signál neuronu,  $b_k$  je zkreslení (bias) neuronu a  $w_{kj}$  je váhou vstupu  $x_{kj}$ . Nakonec je daný výstup předán aktivační funkci  $\phi$ , která definuje konečnou podobu výstupu neuronu

$$y_k = \phi(u_k) [27, 28]. \quad (2.2)$$



Obr. 2.2: Struktura neuronové sítě.

Aktivační funkce je důležitou součástí neuronových sítí, zajišťuje totiž nelinearitru. Bez použití aktivační funkce by byl signál na výstupu neuronu lineární funkcí, a tedy polynomem prvního řádu. Taková neuronová síť by pak byla modelem s lineární regresí, což je nežádoucí, jelikož takový model by nebyl schopen aproximovat složité vztahy mezi daty, pro které byly neuronové sítě navrženy [29].

Dále by měla být aktivační funkce diferencovatelná. To je nezbytné pro algoritmus zpětné propagace (backpropagation) používaný při trénování neuronové sítě. Obecný zápis aktivační funkce viz rovnice 2.2. V neuronových sítích se používají různé aktivační funkce. Několik nejčastěji používaných typů je definovaných v tabulce níže [29, 30].

Další stěžejní částí vývoje každého modelu je trénování, při kterém si jednotlivé neurony optimalizují hodnoty svých vah tak, aby jejich predikce nebo klasifikace daného problému dávala co nejpřesnější výsledky. Jeden z možných přístupů je pak algoritmus zpětné propagace [27, 28].

Tab. 2.1: Aktivační funkce a jejich deriváty. Převzato z [30].

Název	Vztah	Derivát
Sigmoid	$\frac{1}{1+e^{-x}}$	$\frac{e^{-x}}{(1+e^{-x})^2}$
Hyperbolický tangens	$\frac{2}{1+e^{-2x}} - 1$	$\frac{1}{\cosh^2(x)}$
ReLU <sup>1</sup>	$\max(0, x)$	$\begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$
Modifikovaná ReLU	$\max(0, x) + \alpha \min(0, x)$	$\begin{cases} 1, & \text{if } x \geq 0 \\ \alpha, & \text{otherwise} \end{cases}$
PReLU <sup>2</sup>	$\max(0, x) + \alpha \times \min(0, x)$	$\begin{cases} 1, & \text{if } x \geq 0 \\ \alpha, & \text{otherwise} \end{cases}$
ReLU6	$\min(\max(0, x), 6)$	$\begin{cases} 0, & \text{if } x < 0 \text{ or } x > 6 \\ 1, & \text{if } 6 \geq x \geq 0 \end{cases}$
ELU <sup>3</sup>	$\max(0, x) + \min(0, \alpha(e^x - 1))$	$\begin{cases} \alpha e^x, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0 \end{cases}$
SELU <sup>4</sup>	$\gamma(\max(0, x) + \min(0, \alpha(e^x - 1)))$	$\begin{cases} \gamma \alpha e^x, & \text{if } x < 0 \\ \gamma, & \text{if } x \geq 0 \end{cases}$
Swish	$\frac{x}{1+e^{-x}}$	$\frac{e^x(x+e^x+1)}{(e^x+1)^2}$
Mish	$x \tanh(\log(1 + e^x))$	$\frac{e^x(4e^x+4x+6e^{2x}+4e^{3x}+4)}{(2e^x+e^{2x}+2)^2}$

<sup>1</sup> rektifikovaná lineární jednotka (Rectified Linear Unit, ReLU)

<sup>2</sup> parametrická ReLU

<sup>3</sup> exponenciální lineární jednotka (Exponential Linear Unit, ELU)

<sup>4</sup> škálovaná ELU (Scaled ELU)

Zpětná propagace je iterativní algoritmus využívaný pro učení dopředných neuronových sítí (Feed Forward Network, FFN) s využitím optimalizační metody gradientního sestupu. Dopředná neuronová síť je matematickou funkcí složenou z několika vrstev, kde jednotlivé neurony v síti reprezentují lineární kombinace vstupů následované nelineární aktivační funkcí. Gradientní sestup je pak iterativní metodou pro hledání absolutního minima funkce. Cílem zpětné propagace je minimalizovat odchylku sítě od námi požadovaných hodnot [27, 32].

Derivace backpropagation je aplikací gradientního sestupu pro optimalizaci a je závislá na definici loss function (chybová funkce). Mezi běžně používané chybové funkce patří součet čtvercových chyb (Summed Squared Error, SSE)

$$E^{SSE} = \sum_{i \in \text{Vstp}} \sum_{s \in \text{Trn}} (T_i(s) - r_i(s))^2, \quad (2.3)$$

a chyba vzájemní entropie (Cross Entropy Error, CE)

$$E^{CE} = \sum_{i \in \text{Vstp}} \sum_{s \in \text{Trn}} [T_i(s) \ln(r_i(s)) - (1 - T_i(s)) \ln(1 - r_i(s))], \quad (2.4)$$

kde  $i$  náleží výstupu neuronové sítě,  $s$  náleží vzorku trénovací sady,  $T_i(s)$  je skutečnou hodnotou výstupu  $i$  trénovací sady  $s$  a  $r_i(s)$  je predikovaná hodnota výstupu  $i$  vzorku trénovací sady  $s$ . Každý parametr hmotnosti  $w_{ij}$  (váha spojení z  $j$  do  $i$ ) se aktualizuje o hodnotu záporného gradientu velikosti chyby

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \quad (2.5)$$

kde  $\eta$  je velikost kroku, který moduluje vnitřní kompromis mezi hladkou konvergencí vah a rychlostí konvergence a  $E$  reprezentuje chybovou funkci. V případě nízké hodnoty  $\eta$  systém konverguje plynule, ale pomalu. Pro větší hodnoty  $\eta$  je konvergence rychlejší, avšak na úkor její hladkosti. Proto se parametr  $\eta$  nazývá rychlost učení (learning rate) [32].

## 2.4 Základní modely neuronových sítí

### 2.4.1 Konvoluční neuronové sítě

Konvoluční neuronová síť (CNN) je speciálním typem neuronových sítí pro zpracování dat, která mají mřížkovou topologii. Obrazová data můžeme interpretovat jako 2D mřížku pixelů. Jak již jméno napovídá, CNN využívají matematickou operaci konvoluce. Konvoluce je lineární operací a CNN je taková neuronová síť, která namísto násobení matic používá konvoluci alespoň v jedné své vrstvě [33].

Jelikož výpočetní data mají nejčastěji diskrétní charakter, využívá se diskrétní konvoluce, která je definována vztahem

$$s[t] = (x * w)(t) = \sum_{a=-\infty}^{\infty} x[a]w[t - a], \quad (2.6)$$

kde  $x$  a  $w$  je dvojice signálů v čase  $a$ , s kterými byla v čase  $t$  provedena diskrétní konvoluce. V rámci terminologie strojového učení se často nazývá  $x$  vstupem (input) a  $w$  jádrem (kernel). Ve strojovém učení běžně pracujeme se vstupem, který je multidimenzionálním polem a jádro je multidimenzionálním polem naučitelných parametrů. Tyto multidimenzionální pole nazýváme tensory. Protože každý prvek vstupu a jádra musí být uložen odděleně, předpokládáme, že tyto funkce jsou nulové všude kromě konečné množiny bodů, pro které ukládáme hodnoty. To v praxi umožňuje provádět nekonečný součet jako součet přes konečný počet prvků pole [33].

V rámci strojového učení také běžně používáme konvoluci na více než jedné ose naráz. To platí např. pro 2D obrázek  $I$ , který je vstupem. Ve výpočtu použijeme také 2D jádro. Dvojměrná diskretní konvoluce je definována jako

$$s[i, j] = (I * K)[i, j] = \sum_m \sum_n I[m, n]K[i - m, j - n] [33]. \quad (2.7)$$

Vrstvy tradičních neuronových sítí používají maticové násobení pro popis interakce mezi vstupem a výstupem. To znamená, že každý ze vstupů interaguje s každým výstupem. CNN však typicky mají tzv. řídkou interakci (sparse interaction), která je docílena menší velikostí jádra oproti vstupu. Např. při zpracovávání obrazu může mít vstupní obrázek miliony pixelů a i přesto je možné detekovat malé artefakty obrazu (např. hrany) pomocí jader o velikosti pouze stovek pixelů. To znamená, že stačí ukládat méně parametrů, což snižuje paměťové nároky a zvyšuje efektivitu, a k dosažení výstupu je zapotřebí méně výpočtů. Pokud máme  $m$  vstupů a  $n$  výstupů, násobení maticí vyžaduje  $m \times n$  parametrů a použitý algoritmus bude mít výpočetní složitost  $O(m \times n)$  [33].

## 2.4.2 Autoenkodéry

Autoenkodér se skládá ze dvou logických částí, a to z enkodéru a dekodéru. Enkodér se učí vlastnosti vstupních dat a přenáší je do jiného prostoru, aby byla reprezentována v jiné smysluplné reprezentaci. Dekodér vezme konvertovaný signál enkodéru a konvertuje jej zpět do původního prostoru. Autoenkodér obsahuje tři neuronové vrstvy: vstupní, skrytou a výstupní. Počet neuronů ve vstupní a výstupní vrstvě je stejný, počet neuronů ve skryté vrstvě je neomezený. Vstupní a skrytá vrstva tvoří enkodér, skrytá a výstupní vrstva dekodér. Původní data  $X$  jsou na vstupu modelu, po procesu kódování jsou to data  $h$ . Tento proces je definován pomocí vztahu

$$h = f(x) = f(W_1 X + b_1), \quad (2.8)$$

kde  $W_1$  je matice vah mezi vstupní a skrytou vrstvou a  $b_1$  je vektor zkreslení.  $f(x)$  je aktivační funkcí nelineární transformace [34].

Proces dekódování se skládá z mapování konvertovaných dat  $h$  zpět na původní prostor a tím dosáhnout rekonstrukce původních dat  $X^d$ . Proces dekódování je definován vztahem

$$X^d = g(x) = g(W_2 h + b_2), \quad (2.9)$$

kde  $W_2$  je matice vah mezi skrytou a výstupní vrstvou a  $b_2$  je vektor zkreslení.  $g(x)$  je nelineární nebo afinní transformační funkce [34].

Proces kódování je de facto re-extrakce dat do specifického kódu pomocí deterministických mapovacích vztahů. Proces dekódování je pak konverze daného kódu zpět do podoby vstupních dat. Autoenkodér je trénovaný na hledání parametrů (matice vah  $W$  a vektor zkreslení  $b$ ) označených jako  $\theta$ , které minimalizují rekonstrukční chybu mezi  $X^d$  a  $X$ . Celková rekonstrukční chyba autoenkodéru je pak definována vztahem

$$J_{AE}(\theta) = J(X, X^d) = \frac{1}{2} \sum_{i=1}^n \|x_i^d - x_i\|^2, \quad (2.10)$$

kde  $x_i$  je vstup s indexem  $i$  a  $x_i^d$  je jeho rekonstruovaný výstup. Pro výpočet rekonstrukční chyby pro jeden vzorek se použije vzorec

$$L(X, X^d) = \|X^d - X\|^2, \quad (2.11)$$

kde  $X^d = \{x_1^d, x_2^d, \dots, x_n^d\}$  jsou výstupní data a  $X = \{x_1, x_2, \dots, x_n\}$  jsou data vstupní [34].

Výše popsany princip platí pro konvenční autoenkodér. V praxi existuje hned několik typů autoenkodérů, kdy každý je specifický a může být vhodný k odlišným úlohám. Jmenovitě se jedná např. o konvoluční autoenkodér, autoenkodér pro odstranění šumu (denoising autoencoder), variační autoenkodér nebo řídký autoenkodér (sparse autoencoder) [34]. V oboru detekce změn se však nejčastěji používá právě konvenční autoenkodér [24].

### 2.4.3 Transformery

Transformer je poměrně novou architekturou hlubokého učení, která byla představena společností Google v roce 2017.[35] Transformer (viz obr. 2.3) je model skládající se z kodéru a dekodéru, z nichž každý je zásobníkem  $L$  identických bloků. Každý blok kodéru se skládá z modulu vícehlavé pozornosti (multi-head attention) a poziční dopředné sítě [36].

Modul vícehlavé pozornosti zajišťuje mechanismus, který modelu umožňuje zaměřit se na různé aspekty dat v různých reprezentativních podprostorech. Ve vícehlavé pozornosti je vstup mapován do různých lineárních podprostorů (hlav), kdy váhy jednotlivé pozornosti se počítají zvlášť. Výstupy těchto podprostorů jsou pak zřetězeny a lineárně transformovány pro konečný výstup [37].

Protože transformer postrádá opakující se strukturu nebo konvoluci, vyžaduje metodu pro zpracování informací o relativní poloze v sekvenci. Proto je do vstupního vkládání (input embedding) přidáno poziční kódování (positional encoding), které tyto informace poskytne. Poziční kódování je obvykle matice o stejném rozměru jako vstup, která obsahuje hodnoty vypočítané na základě polohy a rozměru [36, 37].

Poziční dopředné sítě se obvykle skládají z jednotlivých vrstev a nelineární aktivační funkce. Nezávisle zpracovávají prvky v sekvenci na každé pozici. Kolem každého modulu je použito reziduální spojení, po kterém následuje modul normalizace (layer normalization). Vstup každé podvrstvy prochází reziduálním spojením, kdy se vstup přidá k výstupu podvrstvy a zachová si původní informace. Následně každý výstup každé podvrstvy prochází modulem normalizace, aby byla zajištěna stabilita a konvergence modelu během jeho učení [36, 37]. V dekodéru se navíc vkládá mezi modul vícehlavé pozornosti a FFN modul křížové pozornosti (cross-attention) [36].

## 2.4.4 Rekurentní neuronové sítě

Rekurentní neuronové sítě (Recurrent Neural Network, RNN) se používají zejména k detekci vzorů v sekvenci dat (např. ručně psané písmo). Jsou však vhodné i pro obrazová data, musí být však rozložena na sérii políček a zpracována jako sekvence. Na vyšší úrovni se RNN používají k rozpoznávání řeči nebo jako generátory textu. To, co odlišuje RNN od dopředných neuronových sítí, je způsob, jakým procházejí data sítí. Zatímco FFN předávají informace síti bez cyklů, RNN cykly obsahují a přenáší informace rekurentně zpět do sebe. RNN tedy zohledňují předchozí vstup  $X_{t-1}$ , nejen aktuální vstup  $X_t$ . Skryté vrstvy jsou agregovány do jednoho skrytého bloku  $H$ , který lze rozšířit [38].

Tento proces předávání informací z předchozí iterace do skryté vrstvy lze popsat pomocí skryté proměnné  $\mathbf{H}_t \in \mathbb{R}^{n \times h}$  a vstupu  $\mathbf{X}_t \in \mathbb{R}^{n \times d}$  v čase  $t$ , kde  $n$  je počet vzorků,  $d$  je počet vstupů jednotlivého vzorku a  $h$  je počet skrytých jednotek. Dále je použita matice vah  $\mathbf{W}_{xh} \in \mathbb{R}^{d \times h}$  matice mezi skrytými vztahy  $\mathbf{W}_{hh} \in \mathbb{R}^{h \times h}$  a zkreslení  $\mathbf{b}_h \in \mathbb{R}^{1 \times h}$ . Nakonec se všechny tyto parametry předají aktivační funkci  $\phi$ , která je většinou sigmoidní nebo tanh funkce. Tyto matematické notace dávají dohromady vztahy pro skrytou proměnnou 2.12 a výstupní proměnnou 2.13, které jsou definovány následovně

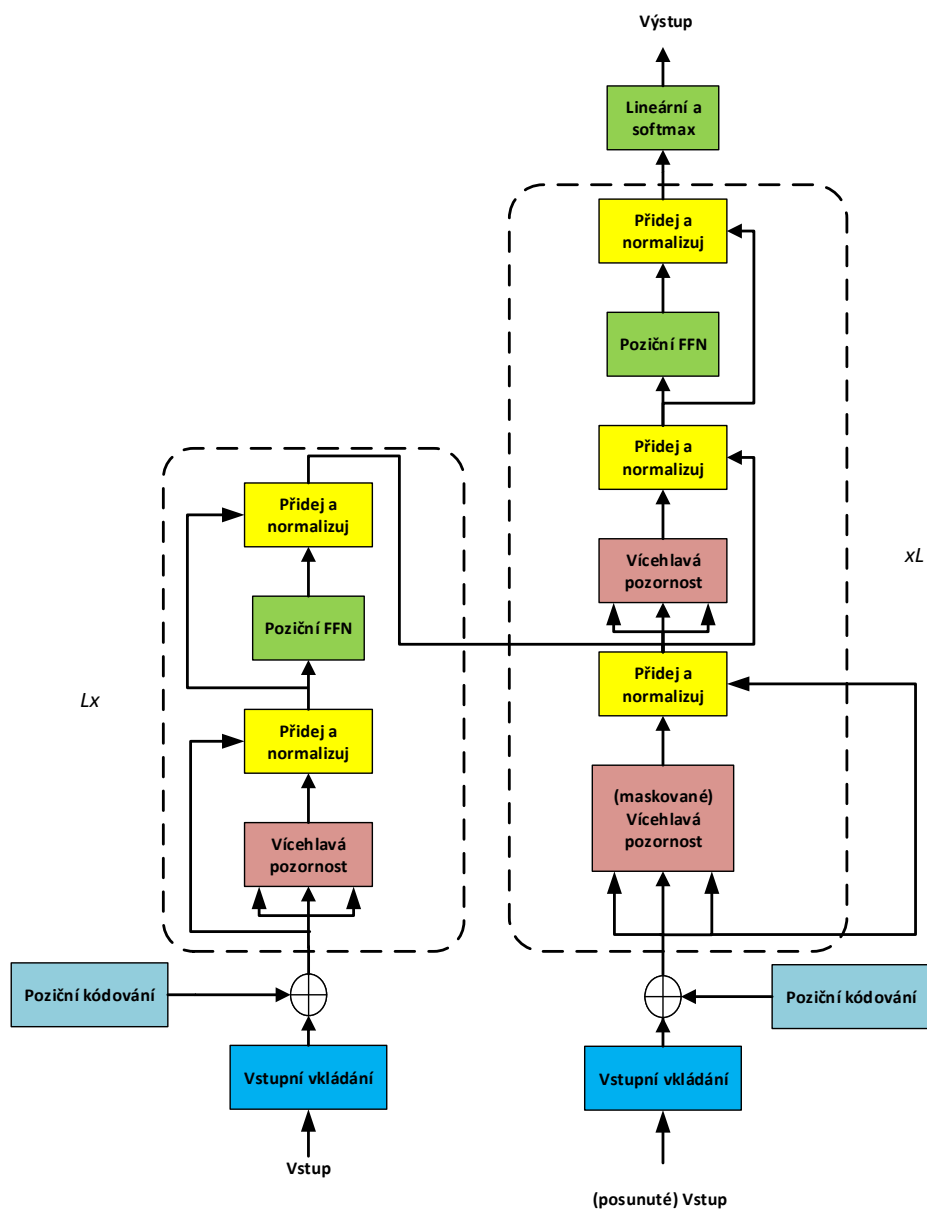
$$\mathbf{H}_t = \phi_h (\mathbf{X}_t \mathbf{W}_{xh} + \mathbf{H}_{t-1} \mathbf{W}_{hh} + \mathbf{b}_h), \quad (2.12)$$

$$\mathbf{O}_t = \phi_o (\mathbf{H}_t \mathbf{W}_{ho} + \mathbf{b}_o). \quad (2.13)$$

Jelikož  $\mathbf{H}_t$  rekurzivně obsahuje  $\mathbf{H}_{t-1}$ , a tento proces se opakuje pro každý krok, RNN v sobě zahrnuje stopy všech skrytých stavů, které předcházely  $\mathbf{H}_t$  a  $\mathbf{H}_{t-1}$  [38].

## 2.4.5 Generativní adversariální sítě

Generativní adversariální sítě (GAN) jsou specifickým typem generativního modelu, který využívá přístup inspirovaný teorií her a je schopen generovat realistické da-



Obr. 2.3: Struktura transformeru. Převezato z [36].

tové vzorky. GAN se skládá ze dvou hlavních neuronových sítí, a to generátoru a diskriminátoru. Ty společně soupeří v rámci procesu, který se nazývá adversariální trénink. Výsledkem tohoto procesu je model schopný generovat data, která jsou nerozlišitelná od těch skutečných. Toho se hojně využívá v oblastech jako je generování obrazu, tvorba syntetických dat apod. [39]

Jak již bylo zmíněno, GAN tvoří dvě neuronové sítě s odlišnými úlohami. Generátor má na základě náhodného vstupu (tzv. latentního vektoru) za úkol vytvořit výstup, který bude připomínat reálná data. Tento vstup slouží jako zdroj variability a umožňuje generátoru produkovat různé vzorky dat. Úkolem generátoru je tedy transformovat latentní vektor na výstup, který bude statisticky podobný datům v tréninkové sadě [39].

Diskriminátor má úlohu přesně opačnou, a to klasifikovat každý vzorek dat jako reálný (tj. pocházející z tréninkové sady) nebo falešný (vygenerovaný generátorem). Diskriminátor se učí rozpoznávat nuance mezi skutečnými a generovanými vzorky pomocí zpětného šíření chyb. To optimalizuje model tak, aby co nejlépe rozeznával rozdíl mezi oběma druhy dat [39].

Tréninkový proces GAN probíhá jako iterativní hra s nulovým součtem mezi generátorem a diskriminátorem. Generátor se snaží produkovat data, která jsou natolik realistická, že je diskriminátor nedovede rozlišit a ztratí schopnost rozpoznávat falešné vzorky. Tento postup je vysoce efektivní a umožňuje GAN generovat realistická data bez nutnosti explicitního modelování pravděpodobnostní distribuce původních dat, což tvoří jeho hlavní výhodu oproti klasickým metodám generativního modelování [39].

## 2.5 Lehké neuronové sítě

Lehké neuronové sítě (Light Neural Network, LNN) jsou technologií, která umožňuje nasazení hlubokého učení na zařízeních s omezeným výpočetním výkonem a kapacitou paměti. Mezi tyto zařízení patří např. mobilní zařízení, internet věcí (Internet of Things, IoT) nebo zařízení na okraji sítí. Fungování LNN na těchto zařízeních s omezenými zdroji je dosaženo pomocí redukce komplexnosti modelů hlubokého učení. Toho lze dosáhnout pomocí různých technik, mezi něž patří ořezávání (pruning), kvantizace, znalostní distilace a hledání architektury neuronových sítí (Neural Architecture Search, NAS). Tyto techniky jsou navrženy tak, aby umožnily úsporu paměti a výpočetních sil při minimálním dopadu na výkonnost nasazených modelů [41].

V rámci zpracování obrazu a dálkového průzkumu je použití lehkých neuronových sítí důležitým tématem. Umožňuje totiž efektivnější zpracování obrazových dat přímo na družici, která má k dispozici omezenou šířku pásma a také omezené

výpočetní vlastnosti. Příkladem je hluboká síť pro komprimované snímání (Deep Compressed Sensing Network - DCSN), která využívá techniky stlačování dat pro efektivní přenos hyperspektrálních dat. Model je navržen tak, aby zajistil nízkou vzorkovací rychlost a tím zmenšil objem přenášených dat a zároveň zvýšil jejich přenosovou robustnost [40].

### 2.5.1 Ořezávání

Ořezávání je jedna z nejstarších metod komprese. Zaměřuje se na redukci redundantních parametrů neuronových sítí. Cílem je odstranit nadbytečné váhy a neurony, které mají minimální vliv na výstup modelu. Základem této metody je identifikace těchto parametrů na základě jejich relativní důležitosti pro výslednou predikci. Například v konvolučních vrstvách lze ořezat některé filtry a tím snížit počet operací násobení a sčítání. To zvyšuje efektivitu výpočtů a snižuje spotřebu energie. Výsledkem je pak model s nižšími nároky na paměť a zrychleným inferenčním časem, což je důležité při nasazení např. na mobilních zařízeních [41].

Matematicky lze operaci ořezávání popsat pomocí vzorce

$$y = Wx, \quad (2.14)$$

kde  $W$  je váhová matice a  $x$  vstupní vektor. Při váhovém ořezávání se nastaví některé prvky matice na nulu, čímž se odstraní spojení mezi konkrétními neurony ve vrstvách. Při ořezávání filtrů se zase eliminují celé kanály či filtry z konvolučních vrstev. Zatímco váhové (tzv. neorganizované) ořezávání vyžaduje záznam indexů a je méně efektivní na hardwarové úrovni, organizované ořezávání filtrů je jednodušší pro nasazení na hardwaru. A to díky efektivnějšímu mapování struktury modelu do paměti a menšímu zatížení výpočetních jednotek. Vhodně provedeným ořezáváním tak lze dosáhnout až několikanásobné redukce paměťových nároků a výpočetní náročnosti modelu bez výrazného dopadu na jeho přesnost [41].

### 2.5.2 Kvantizace

Kvantizace je technika, která redukuje počet bitů potřebných pro reprezentaci váhových parametrů a aktivací v neuronových sítích. To přináší zrychlení výpočtů, protože nižší přesnost znamená menší datovou velikost a méně operací. Obvyklým postupem je přechod z 32bitových desetinných čísel na menší formáty, jako je např. 8bitová nebo 16bitová kvantizace, která si uchovává přesnost predikcí s minimálním dopadem na výsledky modelu. V aplikacích, kde je klíčová efektivita modelu, je využívána i binární kvantizace. Ta redukuje váhy na hodnoty 0 a 1 a umožňuje

realizaci logických operací XNOR, což zvyšuje rychlost výpočtů a snižuje paměťovou náročnost [41].

Speciálním typem je pak tzv. kvantizace s nízkou přesností, která využívá hodnoty menší než 8 bitů. Tento postup může zahrnovat tzv. trénování modelu s ohledem na kvantizaci (quantization-aware training), kdy je model upraven tak, aby kompenzoval ztráty, které způsobila kvantizace při samotném učení. Nebo se využívá tzv. smíšená přesnost, která umožňuje udržet vysokou přesnost predikcí, ale výrazně snižuje spotřebu energie a paměťové nároky [41].

### 2.5.3 Distilace znalostí

Distilace znalostí (knowledge distillation) je technika, kde komplexní a výkonný model, nazývaný učitel, předává své znalosti jednoduššímu modelu, označovanému jako student. Tento proces umožňuje provedení efektivní komprese modelu a zachovává klíčové poznatky složitějšího modelu. Cílem metody je natrénovat studentův model tak, aby co nejvěrněji napodobil výstupy učitele. V klasifikačních úlohách se používá modifikovaná funkce softmax, kde výstupy modelu učitele jsou zjemněny pomocí tzv. teplotního parametru  $t$ , který upravuje měkkost rozdělení pravděpodobností. Ta je popsána vzorcem

$$p_i(z_i, t) = \frac{\exp(z_i/t)}{\sum_j \exp(z_j/t)}, \quad (2.15)$$

kde  $p_i$  popisuje již zmíněnou pravděpodobnost, že vstupní vzorek patří do  $i$ -té třídy a  $z_i$  je tzv. logit, tedy výstup neuronové sítě před aplikací softmaxu pro  $i$ -tou třídu. Vyšší teplota způsobuje měkkčí rozložení pravděpodobností, což studentovi usnadňuje učení se z výstupů učitele [41].

Distilace znalostí zahrnuje nejen koncové výstupy modelu učitele, ale také skryté vrstvy, kdy se studentův model učí z mezivýstupů učitele. Výsledkem je pak optimalizovaný studentův model, který má menší nároky na výpočetní výkon a paměť a který dosahuje podobné přesnosti jako původní učitelův model. Distilace znalostí je rovněž kompatibilní s dalšími metodami, jako online distilace, při které učitel a student trénují současně, což zvyšuje efektivitu tréninku a snižuje nároky na zdroje [41].

### 2.5.4 Hledání architektury neuronových sítí

Hledání architektury neuronových sítí (NAS) automatizuje proces návrhu struktury neuronové sítě s ohledem na optimalizaci výpočetních a paměťových nároků. Cílem NAS je nalézt optimální konfiguraci modelu pro konkrétní výpočetní prostředí. Tento proces zahrnuje vyhledávací prostor, který může zahrnovat různé kombinace velikostí a typů vrstev a také strategii vyhledávání, která může být založena např. na

posilovaném učení nebo evolučních algoritmech. Výsledkem NAS je následně struktura modelu, která je výkonnější a zároveň úspornější než ručně navrhované modely [41].

## 2.6 Veřejně dostupné datasety pro detekci změn

Tabulka 2.2 obsahuje několik veřejně dostupných datasetů, které jsou v oblasti detekce změn široce používány jako experimentální data k trénování a ověření modelů umělé inteligence nebo k posouzení jejich výkonnosti. Uvedené datové sady se vyznačují hustým označováním (labeling), což je činí obzvláště vhodnými pro metody detekce změn s učitelem [37].

Tab. 2.2: Veřejně dostupné datasety pro detekci změn. Převzato z [37].

Název Datasetu	Typ Obrázku	Rozlišení	Počet Párů Obrázků	Rok Pořízení	Zdroj Obrázku
HRCUS-CD	RGB	0,5 m	11 388 párů 256 × 256 pixelů	2010 až 2022	-
GVLN	RGB	0,59 m	17 párů různých velikostí	2010 až 2021	Google Earth
EGY-BCD	RGB	0,25 m	6091 párů 256 × 256 pixelů	2015 až 2020	Google Earth
SI-BU	RGB	0,5–0,8 m	4932 párů 512 × 512 pixelů	2019 až 2021	-
BANDON	RGB	0,6 m	2283 párů 2048 × 2048 pixelů	-	Google Earth, Microsoft Virtual Earth, ArcGIS
Dynamic EarthNet	RGB	3 m	730 párů 1024 × 1024 pixelů	2018 až 2019	Planet Labs
CLCD	RGB	0,5–2 m	600 párů 512 × 512 pixelů	2017 až 2019	GF-2
S2Looking	RGB	0,5–0,8 m	5000 párů 1024 × 1024 pixelů	Různé, 1–3 roky	-
SYSU-CD	RGB	0,5 m	20 000 párů 256 × 256 pixelů	2007 až 2014	Google Earth
DSIFN	RGB	-	3940 párů 512 × 512 pixelů	-	Google Earth
Sense Earth2020	RGB	0,5–0,3 m	4662 párů 256 × 256 pixelů	-	-
Google Dataset	RGB	0,55 m	3109 párů 256 × 256 pixelů	2006 až 2019	Google Earth
LEVIR-CD	RGB	0,5 m	637 párů 1024 × 1024 pixelů	5–14 let	Google Earth
HRSCQD	RGB	0,5 m	291 párů 10 000 × 10 000 pixelů	2005 až 2012	IGN

Název Datasetu	Typ Obrázku	Rozlišení	Počet Párů Obrázků	Rok Pořízení	Zdroj Obrázku
WHU-CD	RGB	0,075 m	Jeden pár 15 354 × 32 507 pixelů	2012 až 2016	Aerial
CDD	RGB	3–100 cm	16 000 párů 256 × 256 pixelů	-	Google Earth
SZTAKI	RGB	1,5 m	13 párů 952 × 640 pixelů	-	-
Hypersp. CDD	Hypersp.	-	Tři páry různých velikostí	2004 až 2014	AVIRIS
River dataset	Hypersp.	30 m	Jeden pár 463 × 241 pixelů	2013 (květen - prosinec)	EO-1 Hyperion
MIS-WH	Multisp.	1 m	Jeden pár 7200 × 6000 pixelů	2002 až 2009	IKONOS
OSCD	Multisp.	10–60 cm	24 párů	2015 až 2018	Sentinel-2
SMARS	RGB, DSM	0,3 m/0,5 m	Dva páry 5600 × 5600 pixelů	-	Syntetické
LEVIR-CC	RGB, Přírodní Jazyk	0,5 m	10 077 párů 512 × 512 pixelů, 50 385 přirozených jazykových prohlášení	5–14 let	Google Earth
MSBC	RGB, Multisp., SAR	2 m	3769 párů 256 × 256 pixelů	2018 až 2019	GF-2, Sentinel-1, Sentinel-2A
MSOSCD	RGB, Multisp., SAR	-	5107 párů 256 × 256 pixelů	2015 až 2018	Sentinel-1, Sentinel-2

## 2.7 Metody hlubokého učení pro detekci změn s učitelem

Metody učení s učitelem využívají multitemporální RS snímky, které jsou anotované a obsahují husté změny, k trénování modelů neuronových sítí. Tyto natrénované modely se následně aplikují na dvojice neanotovaných snímků. V současné době patří metody CD s učitelem mezi nejrozšířenější. S rozvojem hlubokého učení se jako obzvláště úspěšné a perspektivní ukazují dva typy modelů: metody založené na CNN a transformerech. Tyto metody budou podrobněji popsány v následujících podkapitolách [37].

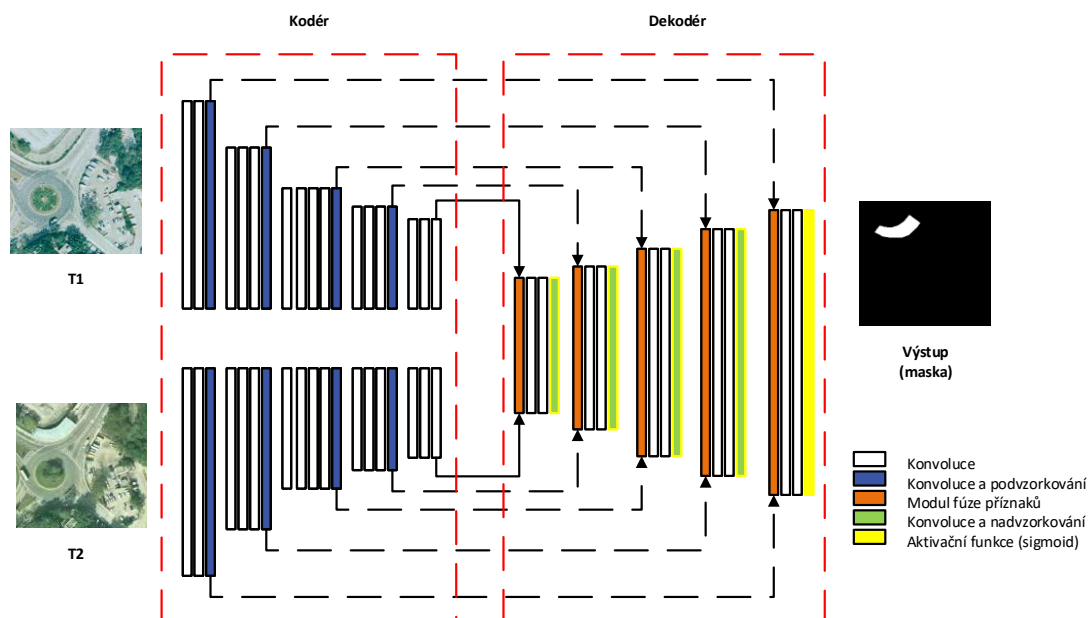
### 2.7.1 Metody založené na CNN

Metody učení s učitelem založené na CNN obvykle využívají architekturu kodér-dekodér. Taková struktura CNN umožňuje efektivní extrakci vlastností ze vstupních dat a následnou přesnou rekonstrukci. V kodéru dochází ke kódování vrstvy pomocí konvoluce a sdružování, kdy se nezpracovaná data postupně mapují do vysoce dimenzionálních reprezentací, které efektivně zachycují prostorové a sémantické informace. V dekodéru pak probíhá proces dekódování prostřednictvím operace dekonvoluce nebo převzorkování. Tímto způsobem dekodér rekonstruuje vysoce-rozměrné prvky na segmenty odpovídající vstupním datům [37].

Dřívější metody obvykle začínaly fúzí obrazu bitemporálního RS zobrazení, za použití techniky přímého vrstvení, diferencování nebo za pomoci analýzy hlavních komponentů (principal component analysis, PCA). Takhle sloučené snímky byly následně vloženy do modelu hlubokého učení s jedním vstupním kanálem [37].

S pokračujícím výzkumem v této oblasti vznikly široce využívané tzv. Siamské síťové struktury (Siamese network). V architektuře toho modelu je bitemporální obraz zpracován pomocí kodérů s identickou strukturou, což umožňuje fúzi prvků na úrovni map. Tato metoda umožňuje modelům lépe porozumět datům a lépe zachytit změny mezi bitemporálními obrazy. To zvyšuje celkovou přesnost a robustnost CD. Navrhování mechanismů pro přenos příznaků nebo metod fúze hlubokých příznaků mezi kodérem a dekodérem pomáhá uchovat více kontextových informací, což je zásadní pro zvýšení přesnosti detekce změn. Integrace různých mechanismů pozornosti pro přenos či fúzi příznaků se stala běžným přístupem, včetně pozornosti na úrovni kanálů, prostorové pozornosti a pozornosti na úrovni kanálů i prostoru. Tyto mechanismy umožňují přiřadit odlišné váhy každému vstupnímu prvku, což zdůrazňuje důležité vlastnosti dat a model se tak více zaměří na aspekty související s úlohou. Siamská struktura je výkonnou a široce využívanou metodou pro CD [37].

Aktuálně tedy většina mainstreamových metod založených na CNN spoléhá na strukturu Siamských síťových kodérů a dekodérů a kombinují je s mechanismy pozornosti v jádru přenosových a fúzních modulů. Struktura obdobné sítě je znázorněna na obr. 2.4 [37].



Obr. 2.4: Struktura CNN sítě využívající architekturu Siamských sítí. Převzato z [37].

## 2.7.2 Metody založené na transformerech

V roce 2020 byl týmem vědců společnosti Google představen model vizuálního transformeru (Vision Transformer - ViT), který byl průkopníkem v aplikaci architektury transformeru na úlohy počítačového vidění. Prostřednictvím experimentů na rozsáhlých souborech obrazových dat bylo prokázáno, že ViT dosahuje výkonu srovnatelného s CNN v úkolech klasifikace obrazu. Tento fakt znamenal oficiální vstup transformerů na pole vizuální domény a dnes se transformery běžně používají v projektech řešících úkoly na obrazových datech [37].

Co se CD úloh týče, dosáhly metody s učitelem založené na transformerech také významných výsledků. Tyto metody, obvykle kombinované s CNN, využívají sílu

CNN při extrakci lokálních příznaků a detailů obrazu, zatímco využívají schopnost transformeru zachycovat globální závislosti a kontextové informace. Tento duální přístup umožňuje komplexní pochopení složitějších změn v dálkovém průzkumu, což vede k velmi dobrému výkonu CD. Běžné struktury modelů, podobné těm, které se používají v úlohách sémantické segmentace v oblasti počítačového vidění, zahrnují použití transformeru jako extraktoru příznaků. Po extrakci mapuje dekodér, založený buď na CNN, nebo na transformerech, příznaky zpět na velikost vstupního obrazu, aby vytvořil výstup CD. Většina metod je postavena na architektuře, jejíž schéma je znázorněno na obr. 2.5 a lze je rozdělit na dva typy na základě typu použitého dekodéru [37].

Prvním typem je transformerový kodér + transformerový dekodér. Tento přístup plně využívá mechanismus sebezpozornosti (self-attention) v obou fázích kódování i dekódování a efektivně integruje globální informace během nadzorkování v procesu dekódování. Tato architektura s plnou pozorností (all-attention) navíc zachovává efektivitu při zpracování vzdálených závislostí a rozsáhlých kontextových informací, a to zejména právě při analýze složitých datových struktur dálkového průzkumu [37].

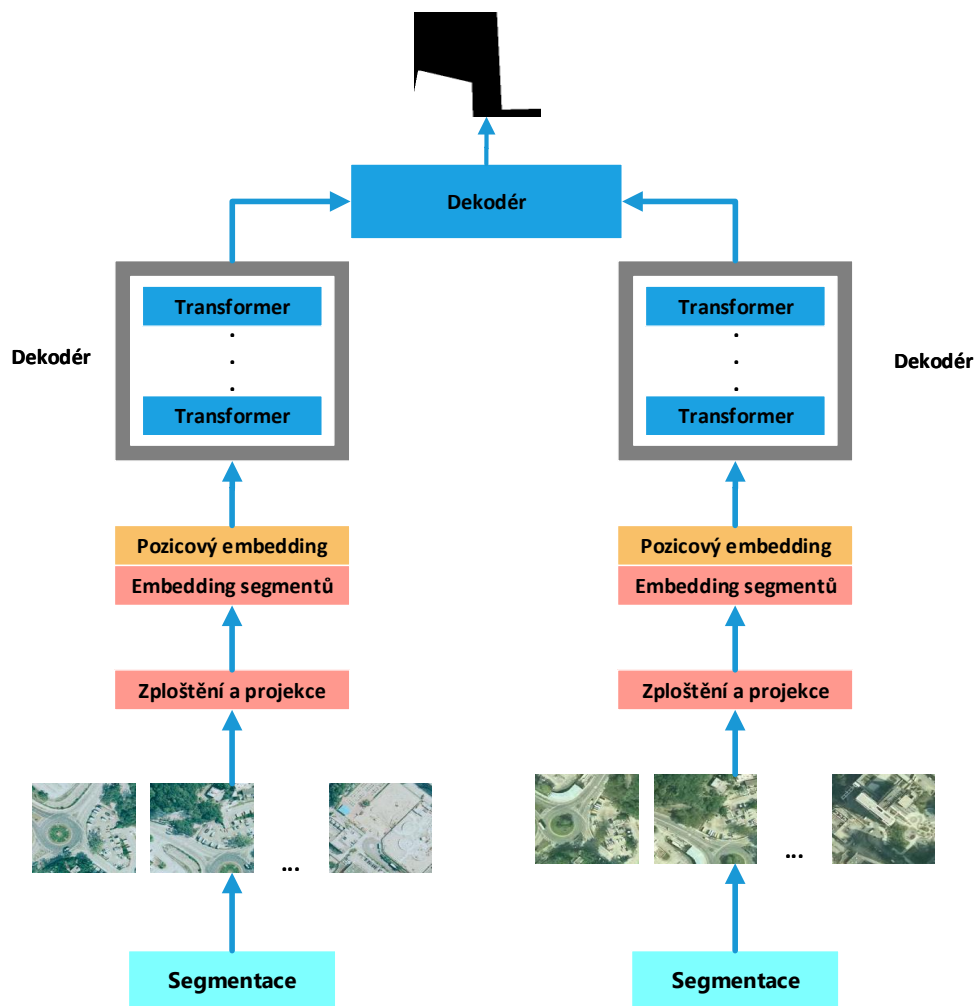
Druhým typem je transformerový kodér + CNN dekodér. V tomto případě funguje transformerový kodér jako extraktor příznaků, zachycující globální kontextové informace vstupních dat. Extrahované příznaky jsou následně předány CNN dekodéru pro jemnější segmentaci a následnou rekonstrukci obrazu [37].

Mimo výše zmíněné existují i další modely využívající transformery, které kombinují odlišné přístupy. Např. využívají lehký vícevrstvý perceptron (Multi-Layer Perceptron, MLP) jako dekodér nebo umísťují transformer mezi CNN kodér a dekodér [37].

## 2.8 Metody hlubokého učení pro detekci změn s částečným učitelem a bez učitele

Modely hlubokého učení s částečným učitelem leží na pomezí mezi metodami s učitelem a bez učitele. Tyto modely se trénují na anotovaných i neanotovaných vzorcích dat, kdy extrahují příznaky ze vzorků obdobně jako modely s učitelem, ale zároveň využívají postupy k rozšíření příznaků neanotovaných vzorků. Tento přístup umožňuje vytvářet efektivnější modely a zároveň snižovat závislost na označených datech, která mohou být v reálných scénářích obtížně sehnatelná, zejména ve velkém množství nutném pro trénování [37].

Modely využívající částečné učení se v oboru detekce změn a obecně dálkového průzkumu Země stávají stále více perspektivními. To je zapříčiněno zejména akumulací velkého množství neanotovaných dat, které souvisí s prudkým nárůstem počtu



Obr. 2.5: Struktura modelu využívající architekturu Transformerů. Převzato z [37].

RS družic na oběžné dráze. V současné době lze modely s částečným učitelem rozdělit do tří kategorií: metody založené na adversariálním učení, metody s vlastním tréninkem a metody s regularizací konzistence [37].

### 2.8.1 Metody založené na adversariálním učení

Tyto metody využívají princip GAN. Trénovací proces GANů představuje optimalizační problém, ve kterém se generátor snaží vytvářet stále realističtější vzorky, aby oklamal diskriminátor, který se snaží rozlišit mezi reálnými a generovanými vzorky. GAN jsou podrobněji popsány v podkapitole 2.4.5 [37].

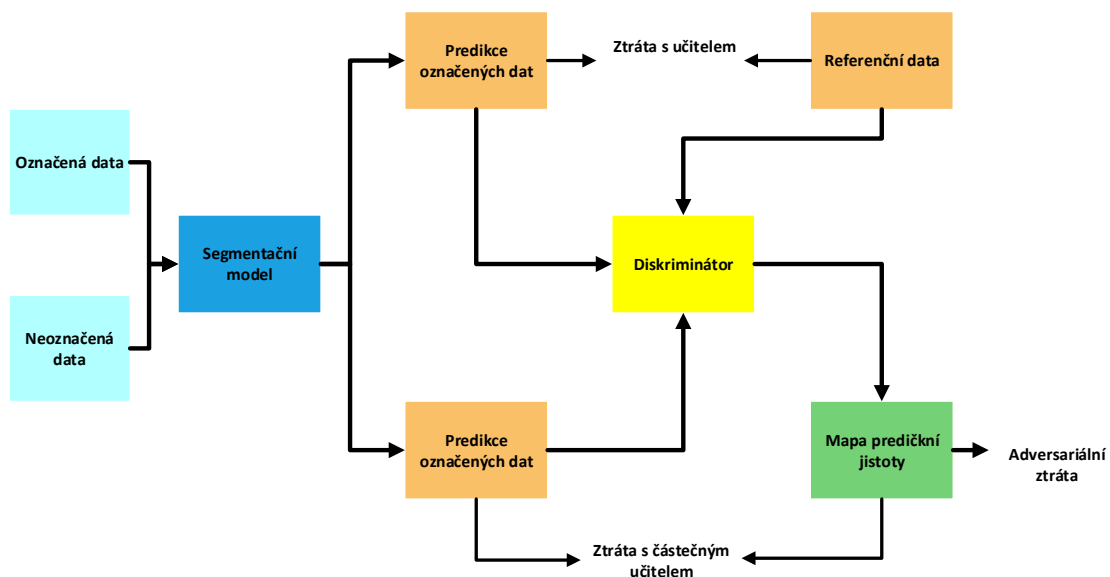
V úlohách CD založených na částečném učiteli hraje klíčovou roli diskriminátor, který se učí rozpoznávat rozdíly mezi reálnými mapami změn a mapami vytvořenými samotnou sítí. Diskriminátor je trénován spolu se segmentačním modelem CD, a to v adversariálním režimu, kdy se snaží přesně identifikovat skutečné a predikované značky. Tento proces umožňuje generovat mapy predikční jistoty i pro neoznačené vzorky. Model získává další užitečné informace tím, že do tréninku zahrnuje i neoznačené vzorky s vysokou predikční jistotou, což zlepšuje jeho schopnost detekce změn. Princip této metody je znázorněn na obr.2.6. Metody CD založené na adversariálním učení s částečným učitelem jsou v současnosti stále ve fázi vývoje [37].

### 2.8.2 Metody založené na vlastním tréninku

U modelů s vlastním tréninkem se parametry modelu natrénovaného na označených vzorcích nejprve použijí k predikci neoznačených vzorků, jejichž předpovědi se následně berou jako pseudo-anotace. Poté je tréninková sada o tyto neoznačené vzorky rozšířena spolu s jejich pseudo-anotací a model je znovu na této sadě trénován [37].

Samotrénování pak probíhá ve třech krocích. Prvním krokem je inicializace, kdy je model nejprve natrénován na dostupné sadě označených dat. Následuje generování pseudo-anotace, kdy model použije pro predikci neoznačené vzorky a vybere vzorky s vysokou predikční jistotou, kterým na základě výsledků přiřadí pseudo-anotace. Posledním krokem je znovu-trénování modelu, kdy jsou neoznačené vzorky s pseudo-anotací sloučeny do označených sad, čímž vznikne rozšířená tréninková sada, na níž je model znovu trénován [37].

Tento postup se iterativně opakuje, kdy se v každé iteraci generují nová tréninková data s pseudo-anotací získaná z neoznačených vzorků. V praxi však může samotrénování vytvářet šum, zejména pokud generovaná pseudo-anotace nejsou dostatečně spolehlivé. Takové pseudo-anotace mohou negativně ovlivnit výsledky tréninku modelu, a proto se často přidávají další strategie pro zvýšení stability a účinnosti samotrénování. Příkladem je aplikace efektivních metod řízení jistoty, které



Obr. 2.6: Struktura modelu využívající architekturu GAN. Převzato z [37].

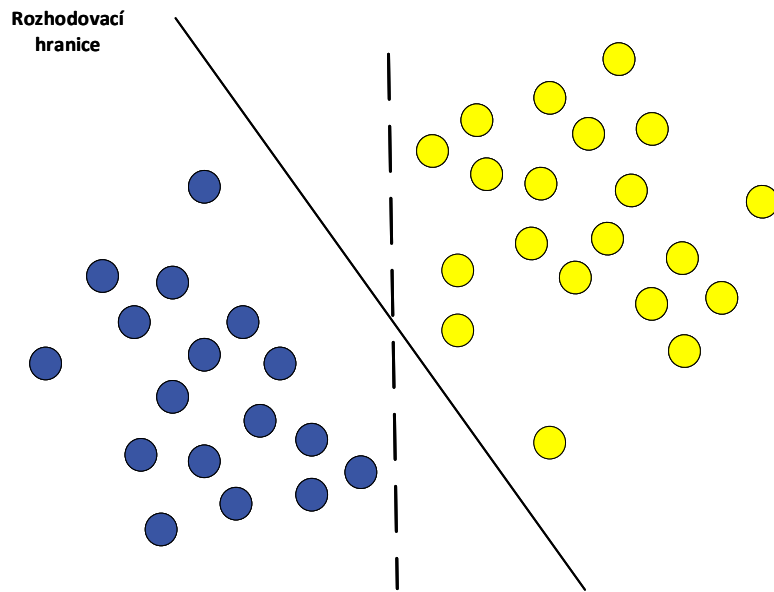
umožňují filtrovat neoznačené vzorky podle spolehlivosti [37].

### 2.8.3 Metody založené na regularizaci konzistence

Podstata metod regularizace konzistence spočívá v trénování modelu tak, aby pro jeden vzorek podrobený různým narušením nebo transformacím přiřazoval obdobné výstupy. Metoda vychází ze dvou základních hypotéz, a to hypotézy hladkosti a hypotézy shlukování [37].

Hypotéza hladkosti předpokládá, že vzorky, které se nacházejí blízko sebe, jsou pravděpodobně stejného označení (viz obr. 2.7). Vzorky stejné kategorie jsou obvykle v prostoru příznaků blíže k sobě než vzorky jiných kategorií. Z toho vyplývá, že model by měl poskytovat obdobné předpovědi pro vzorky, které spolu sousedí [37].

Hypotéza shlukování naznačuje, že rozhodovací hranice by měla ležet v oblastech s nízkou hustotou vzorků. Efektivní rozhodovací hranice by měla procházet co nejřidšími oblastmi vzorků. To modelu pomáhá být méně citlivý na šum a nepodstatné rysy, což zvyšuje stabilitu a celkovou přesnost modelu [37]. Klíčovým prvkem



Obr. 2.7: Hypotéza regularizace konzistence. Převzato z [37].

metod regularizace konzistence je způsob, jakým získat prostor narušení pro neoznačené vzorky a jak následně trénovat model, který bude vůči těmto narušením necitlivý. Tím se zajistí konzistentní předpovědi pro původní obraz i jeho různé narušené verze. Metody regularizace konzistence se podle prostoru narušení dělí do tří kategorií [37]:

- **Prostor narušení obrazu** zahrnuje operace jako rotace, škálování a barevné transformace, kterými se vytváří série narušených verzí původních obrazů [37].
- **Prostor narušení příznaků** pracuje s vnitřním prostorem příznaků obrazu přímo v rámci modelu, místo manipulace s obrazem samotným [37].
- **Prostor narušení modelu** upravuje samotný model, například vytvářením pseudo-annotace pro neoznačené vzorky pomocí různých modelů, které se vzájemně učí [37].

#### 2.8.4 Metody bez učitele

Metody hlubokého učení bez učitele využívají hluboké neuronové sítě k autonomnímu učení příznaků obrazových dat. To umožňuje detekci změn bez předchozích znalostí nebo zásahu z vnějšku. Tyto metody bez učitele, které nevyžadují anotovaná data,

využívají neupravené obrazové snímky k tréninku, což dovoluje zvýšit míru automatizace a přizpůsobivosti modelu [37].

Metody detekce změn bez učitele obvykle kombinují hluboké učení s tradičními technikami CD. Základní myšlenka spočívá v tom, že hluboké sítě extrahují reprezentace příznaků, které jsou následně zpracovány tradičními metodami CD pro získání výsledných map změn [37].

Běžným postupem u těchto metod je například využití lineárních iterativních technik pro vytvoření tzv. super-pixelů, po čemž následuje extrakce příznaků prostřednictvím hluboké sítě založené na kontraktivních autoenkodérech. Tyto extrahované příznaky jsou následně klasifikovány pomocí metod jako k-means pro binární rozdělení na oblasti změn a bez změn, což vede k finální mapě změn [37].

## 2.9 Evaluační metriky

Pro hodnocení metod detekce změn v oblasti dálkového průzkumu Země se používají různé evaluační metriky, které umožňují měřit výkonnost modelů na základě přesnosti a relevance jejich výstupů. Tyto metriky zahrnují počty správně klasifikovaných změněných a nezměněných pixelů, reprezentované jako skutečně pozitivní (True Positive, TP) a skutečně negativní (True Negative, TN), které vyjadřují počet správně identifikovaných změněných a nezměněných pixelů. Dále jsou zde také falešná pozitiva (False Positive, FP) a falešná negativa (False Negative, FN), které naopak označují nesprávně klasifikované změněné a nezměněné pixely. Výsledná hodnota se nazývá přesnost (precision) a odráží schopnost algoritmu přesně identifikovat změněné pixely, zatímco hodnota citlivosti (recall) ukazuje, jak algoritmus efektivně detekuje změněné pixely v rámci celého datasetu [24].

Další metrikou je tzv. průnik přes sjednocení (Intersection over Union, IoU), který je poměrem průniku a sjednocení predikované mapy a pravdivých dat. Tím poskytuje komplexní měřítko shody mezi těmito vrstvami. Výpočet IoU je definován vztahem

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}. \quad (2.16)$$

Celková přesnost (Overall Accuracy, OA) pak měří obecnou přesnost predikce pomocí následujícího vzorce

$$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \quad (2.17)$$

Důležitou složenou metrikou je F1 skóre, které je harmonickým průměrem mezi

přesností a citlivostí a je definováno následovně

$$F_1 = \frac{2TP}{2TP + FP + FN}. \quad (2.18)$$

Pro specifické aplikace, jako je např. klasifikace založená na matici záměn, se používá koeficient Kappa (Kappa Coeficient, KC), který představuje míru shody s klasifikační přesností. Kromě toho existují další metriky jako odmocnina střední čtvercové chyby (Root Mean Squared Error, RMSE) nebo změněná odmocnina střední čtvercové chyby (changed RMSE nebo cRMSE), které se používají zejména pro evaluaci trojrozměrných modelů detekce změn. RMSE je dána vztahem

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\widehat{\Delta H}_i - \Delta H_i)^2}, \quad (2.19)$$

kde  $\widehat{\Delta H}_i$  je predikovaná hodnota,  $\Delta H_i$  je skutečná hodnota změny a  $n$  je počet měření. cRMSE je pak dána vztahem

$$\text{cRMSE} = \sqrt{\frac{1}{n_c} \sum_{i=1}^{n_c} (\widehat{\Delta H}_i^C - \Delta H_i^C)^2}, \quad (2.20)$$

kde  $\widehat{\Delta H}_i^C$  je predikovaná hodnota změny výšky pro změněné pixely,  $\Delta H_i^C$  je skutečná hodnota změny pro změněné pixely a  $n_c$  je počet změněných pixelů, kde došlo ke změně výšky [24].

## 3 Implementace

### 3.1 Předzpracování dat

Prvním krokem je vyhledání a výběr vhodných datasetů, které budou použity. Na internetu je volně dostupných mnoho datasetů pro detekci změn, jak je uvedeno v tab. 2.2. Tyto datasety mají různé technické vlastnosti (RGB, hyperspektrální a multispektrální obraz), rozlišení a zaznamenávají různé typy změn (urbanizaci, přírodní prostředí, infrastrukturu apod.). V rámci této práce byly zvoleny datasety obsahující RGB snímky, s libovolným rozlišením a mapující změny primárně v urbanizaci a infrastruktuře. A to konkrétně datasety LEVIR, SYSU, EGY-BCD a DSIFN (parametry datasetů jsou obsaženy v tab. 2.2.).

Tyto datasety jsou, dříve než budou použity k trénování modelu, předzpracovány tak, aby data byla konzistentní. Toho je dosaženo pomocí několika kroků zpracování obrazu, během kterých dojde k (mimo jiné) sjednocení vlastností pixelů obrazu pomocí normalizace a následnému rozdělení do dlaždic (tiles), kdy výstupní snímky mají konstantní velikost 256x256 pixelů. Celý proces předzpracování je znázorněn na obr. 3.1. Jednotlivé kroky budou podrobněji popsány v následujících odstavcích.

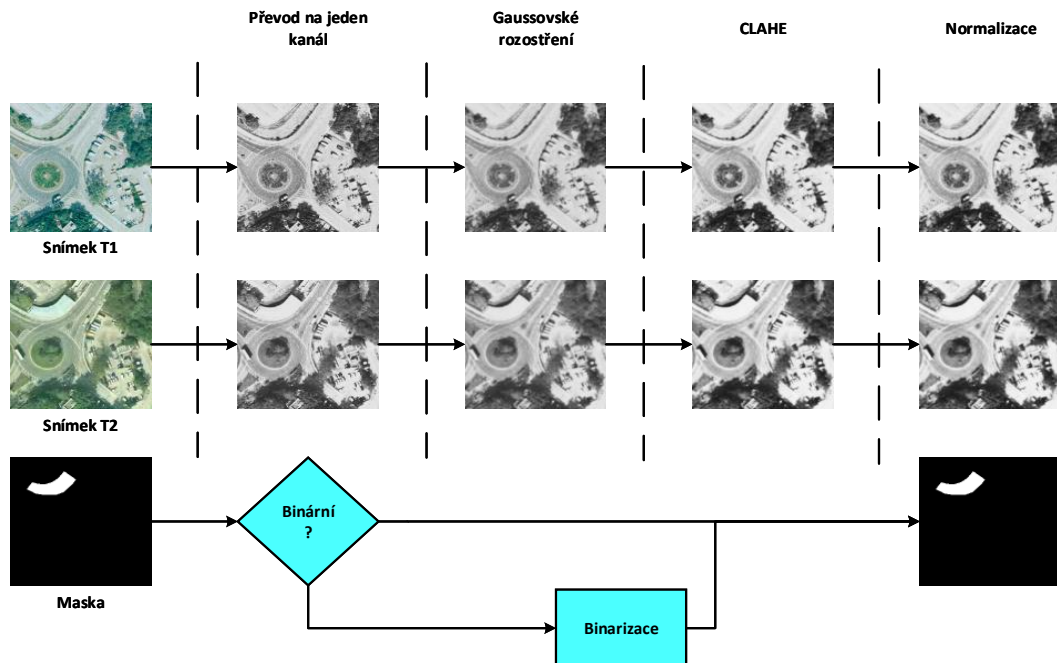
Nejprve jsou vytvořeny složky pro trojice snímků v čase  $t_1$  a  $t_2$  a jim příslušné masky změn. Do těchto tří složek jsou následně nahrána veškerá data zmíněných datasetů. Jak již bylo řečeno, data jsou nekonzistentní: snímky jsou různých velikostí (1024x1024, 512x512 a 256x256), mají různé jasové vlastnosti a šum. V prvním kroku předzpracování se pracuje s obrazovými vlastnostmi barevných snímků  $t_1$  a  $t_2$ , kdy dochází k odstranění šumu, ekvalizaci adaptivního histogramu s omezeným kontrastem a nakonec k normalizaci obrazu. U masky je provedena pouze binarizace jasových hodnot pixelů na hodnoty 0 a 255 (kdy pixel 255 značí detekovanou změnu). K práci s obrazovými daty byla použita knihovna OpenCV [42].

K odstranění šumu a redukci jemných detailů je použito Gaussovské rozostření (Gaussian blur), které se v oblasti zpracování obrazu běžně používá. Tento filtr je založen na pravděpodobnostní distribuci matematického modelu Gaussovy funkce, ve dvourozměrném prostoru definovanou vztahem

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (3.1)$$

kde  $x$  a  $y$  jsou souřadnice pixelu vůči středu a  $\sigma$  je směrodatná odchylka, která ovlivňuje šířku Gaussovy křivky, čímž určuje míru rozmazání. Funkce vytváří váženou masku, kde váhy klesají exponenciálně s rostoucí vzdáleností od středu, což vede k jemnému a přirozenému rozmazání obrazu [43].

Aplikace Gaussovského rozostření se provádí pomocí konvoluce, kde je původní



Obr. 3.1: Princip předzpracování vstupních dat

obraz filtrován pomocí masky odvozené z Gaussovy funkce

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k G(i, j) \cdot I(x + i, y + j), \quad (3.2)$$

kde  $I(x, y)$  je hodnota pixelu v původním obraze,  $G(i, j)$  je váha filtru a  $k$  je polovina velikosti masky. Proces zahrnuje zprůměrování hodnot pixelů s ohledem na jejich váhu, což má za následek zjemnění ostrých přechodů a odstranění šumu [44].

Adaptivní ekvalizace histogramu s omezeným kontrastem (Contrast Limited Adaptive Histogram Equalization, CLAHE) je metoda pro zlepšení kontrastu obrázku, která provádí histogramovou ekvalizaci na menších oblastech obrazu, namísto na obraze celém. Tento postup řeší některé limity tradiční histogramové ekvalizace (např. nadměrné zlepšení kontrastu v homogenních oblastech obrázku, které by mohlo zvýraznit šum). V CLAHE se obrázek rozdělí na tzv. dlaždice a následně se aplikuje na každou z těchto oblastí histogramová ekvalizace. Histogramy jsou následně oříznuty na předem definovanou mez pro kontrast (clip limit), aby se zabránilo již zmíněnému nadměrnému zesílení kontrastu. Po provedení ekvalizace se sousední dlaždice hladce

spojují pomocí bilineární interpolace, aby se předešlo viditelným hranicím mezi nimi [45].

Nakonec je u obrazových dvojic  $t_1$  a  $t_2$  vykonána min-max normalizace, která škáluje hodnoty pixelů v obrazu tak, aby spadaly do rozsahu  $[0, 255]$ , pomocí vzorce

$$x_{\text{norm}} = \frac{x - \min}{\max - \min} \times (\max_{\text{nove}} - \min_{\text{nove}}) + \min_{\text{nove}}, \quad (3.3)$$

kde  $x$  je hodnota pixelu,  $\min$  a  $\max$  jsou minimální a maximální hodnoty obrazu a  $\min_{\text{nove}}$ ,  $\max_{\text{nove}}$  definují nový rozsah [46].

U masky je nejprve testováno, zda-li je šedotónová. Pokud ne, je maska do odstínů šedé převedena. Následně je maska testována na binaritu. V rámci trénování modelu je nezbytné, aby masky obsahovaly pouze dvě hodnoty pixelů (0 a 255). Pro nebinární masky je následně provedena binarizace pomocí Otsuovy metody prahování. Jelikož většina masek v použitých datasetech již binární je, je v kódu uvedena tato metoda spíše pro úplnost.

Otsuova metoda je algoritmus používaný pro automatické určení prahu binarizace. Analyzuje histogram obrazu a hledá práh, který minimalizuje vnitrotřídní rozptyl, což odpovídá maximalizaci rozdílů mezi objektem a pozadím. Výsledná rovnice je

$$\sigma_B^2 = P_1(\mu_1 - \mu_T)^2 + P_2(\mu_2 - \mu_T)^2, \quad (3.4)$$

kde  $P_1$  a  $P_2$  jsou pravděpodobnosti tříd a  $\mu_1$ ,  $\mu_2$  a  $\mu_T$  odpovídají středním hodnotám [47, 48].

Po obrazovém zpracování je snímek předán k vytvoření výstřížků, kdy je rozdělen na samostatné snímky o velikosti 256x256 (pokud sám má větší rozměry). Během procesu je každý snímek označen a uložen do výstupních složek. Celkem bylo tímto postupem dosaženo datasetu o počtu **52 043 trojic**. Tímto je dataset připraven pro trénování modelu. Ještě předtím je možné provést augmentaci dat a dataset tím uměle zvětšit. Vhodnou augmentací je také možné dosáhnout lepší odolnosti modelu vůči variacím (změny kontrastu, šumu apod.).

### Výpis 3.1: Funkce pro předzpracování souboru

```
1 def preprocess_file(file_info):
2     try:
3         filename, input_folder, output_folder = file_info
4         image_path = os.path.join(input_folder, filename)
5         image = cv2.imread(image_path)
6
7         if image is None:
8             raise ValueError(f"Chyba při načítání obrázku:
9                 ↪ {filename}")
10
11         if output_folder != output_dir_mask: # Pokud snímek není
12             ↪ maska
13             # Převod na jeden kanál
14             image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
15             # Gaussian Blur
16             image = cv2.GaussianBlur(image, (5, 5), 0)
17             # CLAHE
18             clahe = cv2.createCLAHE(clipLimit=2.0,
19                 ↪ tileGridSize=(8, 8))
20             image = clahe.apply(image)
21             # Normalizace Min-Max
22             image = cv2.normalize(image, None, 0, 255,
23                 ↪ cv2.NORM_MINMAX).astype(np.uint8)
24
25         else: # Pokud snímek je maska
26             # Pokud obrázek není jednokanálový
27             if len(image.shape) == 3:
28                 image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
29
30             # Pokud obrázek neobsahuje pouze hodnoty 0 a 255, je
31             ↪ binarizován
32             if not np.all(np.isin(image, [0, 255])):
33                 threshold, image = cv2.threshold(image, 0, 255,
34                 ↪ cv2.THRESH_BINARY + cv2.THRESH_OTSU)
35             # Uložení threshold hodnoty
36             threshold_values.append({"file": filename,
37                 ↪ "threshold": threshold})
38
39         create_patches(image, os.path.splitext(filename)[0],
40             ↪ output_folder)
41
42     except Exception as e:
43         print(f"Chyba při zpracování souboru {filename}: {e}")
```

## 3.2 Augmentace dat

Po předzpracování dat je provedena jejich augmentace. V této části kódu je tedy s určitou pravděpodobností provedena na každé trojici snímků jedna nebo více změn a tato trojice je následně uložena jako nová a přidána k datasetu. Míra pravděpodobnosti, že k augmentaci dané trojice dojde, se určuje na základě toho, jak moc si přejeme dataset rozšířit. Jelikož velikost datasetu byla již před augmentací shledána jako dostačující (52 043 trojic), na samotném datasetu provedena nebyla a je zde uvedena pro úplnost problematiky.

Na dvojicích snímků  $t_1$  a  $t_2$  mohou být provedeny s pravděpodobností 0,5 následující operace:

- **Náhodná rotace** - Tato funkce provádí náhodnou rotaci obrázku kolem jeho středu o zadaný úhel. Nejprve je získán střed rotace, který je určen jako polovina šířky a výšky obrázku. Následně je vypočítána transformační matice a nakonec je aplikována rotace s určeným interpolačním režimem nejbližší soused (`cv2.INTER_NEAREST`)

Výpis 3.2: Funkce pro náhodnou rotaci

```
1 # Funkce pro náhodnou rotaci
2 def random_rotation(image, angle, interpolation=
3     cv2.INTER_NEAREST):
4     height, width = image.shape[:2]
5     center = (width // 2, height // 2)
6     rotation_matrix = cv2.getRotationMatrix2D(center,
7         angle, 1.0)
8     return cv2.warpAffine(image, rotation_matrix,
9         (width, height), flags=interpolation)
```

- **Převrácení obrazu** - Tato funkce provádí převrácení obrazu, kdy parametr `flip_code` určuje směr převrácení (0 pro horizontální, 1 pro vertikální a -1 pro obě osy).

Výpis 3.3: Funkce pro převrácení obrazu

```
1 # Funkce pro náhodné převrácení
2 def random_flip(image, flip_code):
3     return cv2.flip(image, flip_code)
```

- **Náhodná změna jasu a kontrastu** - Tato funkce upravuje jas a kontrast obrazu. Parametr `alpha` je škálovací faktor kontrastu, který určuje velikost rozdílu mezi světlými a tmavými oblastmi. Parametr `beta` je odchylkou pro jas, kdy přidává nebo odebírá konstantní hodnotu k jasu, čímž zvyšuje nebo snižuje celkovou světlost obrazu.

### Výpis 3.4: Funkce pro náhodnou změnu jasu a kontrastu

```
1 # Funkce pro náhodnou změnu jasu a kontrastu
2 def random_brightness_contrast(image, alpha, beta):
3     return cv2.convertScaleAbs(image, alpha=alpha,
4         beta=beta)
```

Změny v trojici snímků musí být konzistentní, tzn. že každý snímek musí podstoupit stejné obrazové a transformační změny se stejnými parametry. Masky jsou binární, tedy obrazové změny u nich nemají žádný význam, ovšem transformační modifikace musí zůstat totožné se snímky  $t_1$  a  $t_2$ . Konzistence je dodržena pomocí funkce `augment_images`, která provádí výše zmíněné augmentace na trojicích. U rotace si lze povšimnout, že snímky  $t_1$ ,  $t_2$  a masky používají jiné interpolační techniky. Dvojice snímků používá interpolaci `INTER_LINEAR`. Tato interpolace umožňuje plynulé přechody mezi hodnotami pixelů, což je důležité pro zachování detailů v obrazových datech. Naopak u masek je vhodnější interpolace `INTER_NEAREST`, protože masky obsahují pouze binární hodnoty pixelů a použití interpolace s nejbližším sousedem zajistí, že tyto hodnoty zůstanou diskrétní a nebudou interpolovány takovým způsobem, který by mohl změnit jejich povahu.

### Výpis 3.5: Funkce pro augmentaci

```
1 # Funkce pro konzistentní augmentaci
2 def augment_images(t1_image, t2_image, mask_image):
3     # Náhodná rotace
4     if random.random() < 0.5:
5         angle = random.uniform(-30, 30)
6         t1_image = random_rotation(t1_image, angle,
7             interpolation=cv2.INTER_LINEAR)
8         t2_image = random_rotation(t2_image, angle,
9             interpolation=cv2.INTER_LINEAR)
10        mask_image = random_rotation(mask_image, angle,
11            interpolation=cv2.INTER_NEAREST)
12
13    # Náhodné převrácení
14    if random.random() < 0.5:
15        flip_code = random.choice([-1, 0, 1])
16        t1_image = random_flip(t1_image, flip_code)
17        t2_image = random_flip(t2_image, flip_code)
18        mask_image = random_flip(mask_image, flip_code)
19
20    # Náhodná změna jasu a kontrastu (jen T1 a T2)
21    if random.random() < 0.5:
22        alpha = random.uniform(0.8, 1.2)
23        beta = random.uniform(-30, 30)
24        t1_image = random_brightness_contrast(t1_image, alpha,
25            beta)
26        t2_image = random_brightness_contrast(t2_image, alpha,
27            beta)
28
29    return t1_image, t2_image, mask_image
```

## 3.3 Statistický popis dat

Statistický popis datasetu je dalším krokem přípravy dat pro trénování modelu strojového učení, včetně úloh detekce změn. V tomto případě se analyzuje trojice obrázků: dva časově odlišné snímky  $t_1$ ,  $t_2$  a jejich odpovídající maska změn. Statistický popis datasetu umožňuje identifikovat jeho charakteristiky, jako je rozložení hodnot pixelů, rozsah intenzit a přítomnost odlehlých hodnot. Pro detekci změn má pochopení rozdílů mezi časovými snímky zásadní význam, jelikož modely neuronových sítí závisí na zřetelných a konzistentních vzorcích přítomných v datech. Pro masky se zaměřujeme spíše na podíl hodnot "změna" a "bez změny". Každý snímek v datasetu lze charakterizovat pomocí základních statistik:

- **Průměr a medián** popisují střední hodnotu pixelů.
- **Rozptyl a směrodatná odchylka** ukazují variabilitu v datech.

Pro snímky  $t_1$  a  $t_2$  jsou tyto metriky důležité k určení, zda snímky pocházejí ze srovnatelných zdrojů, nebo jestli mezi nimi existují významné rozdíly.

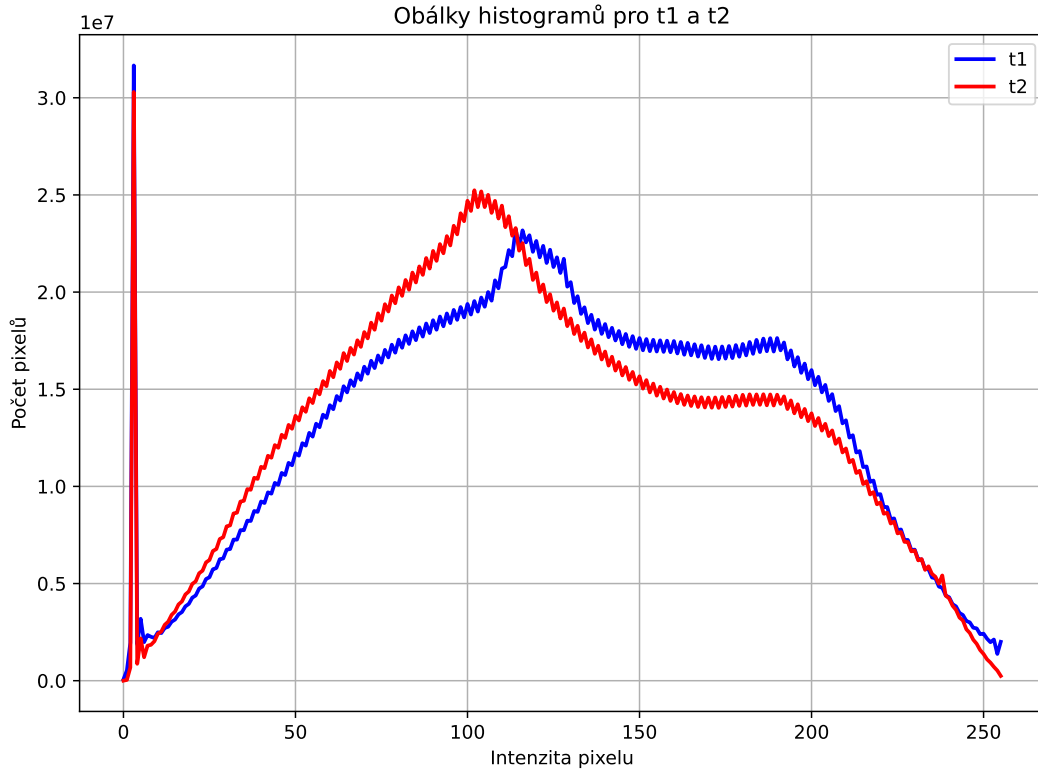
Tab. 3.1: Globální statistické hodnoty

Metoda	$t_1$	$t_2$
Průměr	124.3379	117.8577
Medián	124.2587	116.2370
Rozptyl	3010.4889	3034.4847
Směrodatná odchylka	53.2585	53.6814

V tabulce 3.3 jsou výsledné hodnoty základní statistické analýzy pro všechny snímky datasetu. U průměru a mediánu jsou hodnoty pro  $t_1$  mírně vyšší než u  $t_2$ . To naznačuje celkově světlejší tón snímků  $t_1$ . Medián blízký průměru obou snímků zase ukazuje rovnoměrné rozložení hodnot.

Rozptyl a směrodatná odchylka jsou téměř stejné. To znamená, že variabilita intenzity pixelů v obou časech je podobná.

Výsledky ukazují, že snímky jsou srovnatelné z hlediska statistických vlastností, ale existují mezi nimi jemné rozdíly, které by mohly být podstatné pro detekci změn.



Obr. 3.2: Porovnání obálek histogramů.

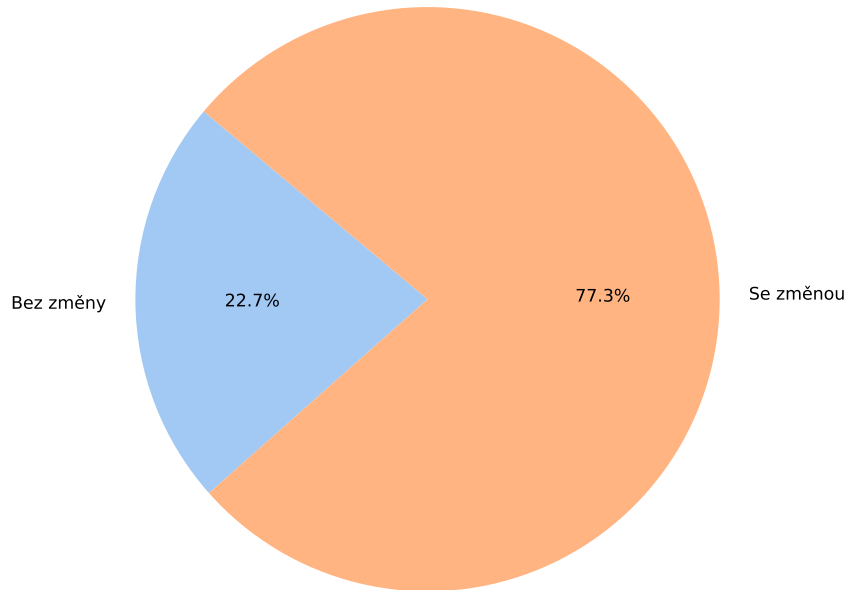
Dalším nástrojem statistické analýzy datasetu je histogram pixelů, který je základním nástrojem pro vizuální analýzu rozložení intenzity v obrázcích. Pro snímky  $t_1$  a  $t_2$  histogramy pomáhají identifikovat globální změny v jasnosti, kontrastu nebo saturaci. U masek nám pak analýza intenzity pixelů pomůže identifikovat rozložení tříd na masky se změnou a bez změn (viz obr. 3.3).

Na obr. 3.2 jsou porovnané histogramy obou časových snímků. Oba histogramy vykazují podobné trendy s totožnou koncentrací pixelových hodnot v určitém rozsahu intenzity. Rozsah pokrývá celé spektrum, od minimálních hodnot (0) k maximálním (255). To naznačuje, že oba snímky zachycují širokou škálu světlých a tmavých oblastí, což je žádoucí pro učení modelu schopného rozpoznávat změny v různých podmínkách osvětlení. Lze si všimnout i patrných rozdílů mezi histogramy. Snímek  $t_1$  má ve srovnání s  $t_2$  mírně posunutý vrchol intenzity. To by odpovídalo nižšímu průměru a mediánu dle tab. 3.3. Pokud by histogramy byly téměř identické, bylo by obtížnější odlišit skutečné změny od šumu.

V histogramu si lze také povšimnout výskytu zcela tmavých oblastí. To je dáno některými snímky v datasetu, které jsou uměle natočeny (již od zdroje) a jejich část tvoří informačně redundantní plocha, která je vyplněna černými pixely.

Na obr. 3.3 je znázorněn podíl masek obsažený v datasetu. Bez změn je 22,7%

Podíl masek se změnou a beze změny

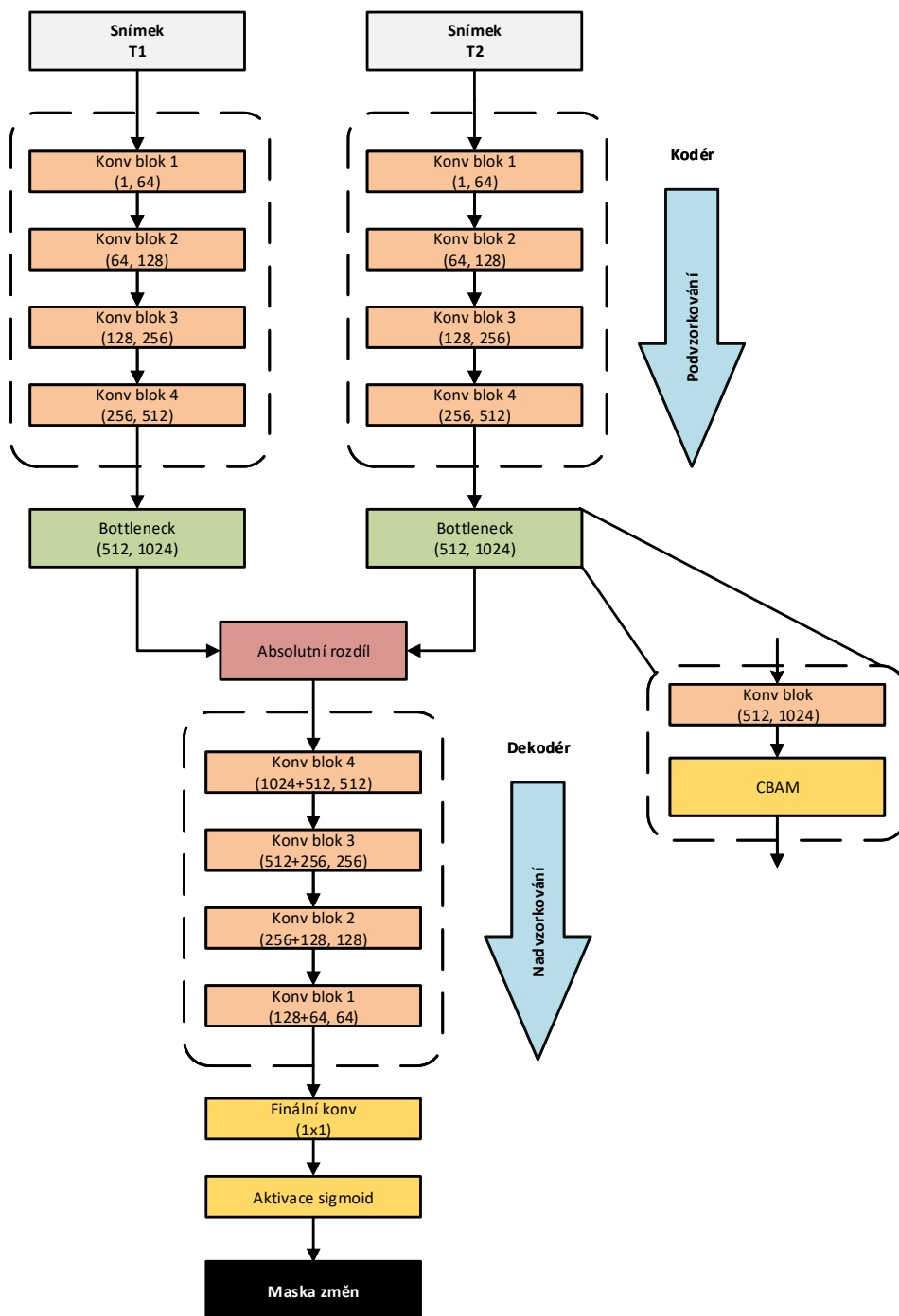


Obr. 3.3: Podíl masek v datasetu.

masek a se změnou 77,3%. Převaha masek se změnami je vhodnější pro trénování modelů, u kterých se počítá s nasazením v oblastech, kde se očekávají frekventované změny. V opačném případě by bylo vhodné zvětšit počet snímků bez změn pomocí augmentace, nebo zvolit jinou strategii pro vyrovnaní počtu.

### 3.4 Návrh modelu

Pro realizaci modelu byla vybrána architektura založená na konvolučních sítích (viz 2.4.1), konkrétně Siamese U-net. Tato architektura se skládá ze dvou identických větví sdílejících stejné váhy. Každá větev zpracovává jeden vstupní snímek a extrahuje z něj reprezentaci v  $n$ -rozměrném prostoru. Výsledné reprezentace jsou nakonec porovnány za účelem vyhodnocení podobnosti mezi vstupními daty. Tímto přístupem je dosaženo efektivního rozlišování mezi změněnými a nezměněnými oblastmi v úloze detekce změn [49]. Pro naprogramování modelu byla využita knihovna PyTorch [50]. Blokové schéma modelu je zobrazeno na obr. 3.6 na následující straně.



Obr. 3.4: Blokové schéma modelu Siamese U-net.

### 3.4.1 Kodér

Model obsahuje dvě větve kodéru s celkem čtyřmi konvolučními bloky. Každý blok se skládá ze dvou operací konvoluce o rozměru 3x3, normalizace dávky (batch normalization), aktivační funkce ReLU (viz 2.3), volitelného CBAM modulu (aplikován pouze v konvolučním bloku úzkého hrdla) a volitelnou dropout vrstvou (aplikovanou pouze v konvolučních blocích dekodéru).

Normalizace dávky slouží ke zlepšení rychlosti trénování a stability modelu díky normalizaci výstupů neuronů v každé dávce na průměr a rozptyl. Normalizované hodnoty se transformují pomocí dvou naučených parametrů: škálování (gamma) a posunutí (beta), což umožňuje modelu stále se učit potřebné transformace. Díky normalizaci dávky model vykazuje rychlejší konvergenci a lepší generalizaci [51].

Dropout je technika regularizace, která slouží k prevenci přetrénování (overfitting) modelu. Během trénování dropout náhodně "vypíná"(tj. nastavuje na nulu) určité procento neuronů v síti. Tento proces se provádí během každé iterace, což znamená, že různé podmnožiny neuronů jsou při každém kroku ignorovány. Cílem je, aby model nebyl příliš závislý na konkrétních neuronech a aby se tím zvýšila jeho schopnost generalizace [52].

Výpis 3.6: Definice konvolučního bloku

```
1 def conv_block(self, in_channels, out_channels, use_cbam=False,
2   ↪ use_dropout=False):
3     """Konvoluční blok s volitelným CBAM a Dropout2d"""
4     layers = [
5         nn.Conv2d(in_channels, out_channels, kernel_size=3,
6   ↪ padding=1),
7         nn.BatchNorm2d(out_channels),
8         nn.ReLU(inplace=True),
9         nn.Conv2d(out_channels, out_channels, kernel_size=3,
10   ↪ padding=1),
11         nn.BatchNorm2d(out_channels),
12         nn.ReLU(inplace=True),
13     ]
14
15     if use_cbam:
16         layers.append(CBAM(out_channels))
17
18     if use_dropout:
19         layers.append(nn.Dropout2d(p=0.3))
20
21     return nn.Sequential(*layers)
```

Každá vrstva kodéru dvojnásobně navyšuje počet výstupních kanálů, tj. z 1 (model přijímá jednokanálový vstup - šedotónové snímky) na 64 a následně 128, 256 a 512. Každá větev kodéru je zakončena tzv. úzkým hrdlem (bottleneck). Tato poslední vrstva kodéru obsahuje nejabstraktnější reprezentaci vstupního obrazu a její hlavní funkcí je vytvoření komprimované reprezentace vstupního obrazu, která zachovává důležité informace pro následující kroky v dekodéru.

Výpis 3.7: Vrstvy kodéru

```

1 self.enc_conv1 = self.conv_block(1, 64, use_cbam=False)
2 self.enc_conv2 = self.conv_block(64, 128, use_cbam=False)
3 self.enc_conv3 = self.conv_block(128, 256, use_cbam=False)
4 self.enc_conv4 = self.conv_block(256, 512, use_cbam=False)

```

### 3.4.2 Modul pozornosti konvolučního bloku

Modul pozornosti konvolučního bloku (Convolutional Block Attention Module, CBAM) je poměrně lehký a efektivní modul pozornosti vyvinutý pro CNN. Obecně jsou moduly pozornosti používány za účelem dosažení lepšího "soustředění" pozornosti modelu na oblasti s důležitými informacemi. Pro úlohy počítačového vidění se jedná zejména o potlačení učení na oblastech obsahujících šum a rušivé prvky.

CBAM obsahuje dvě klíčové složky, a to kanálovou a prostorovou pozornost. Kanálový modul se zaměřuje na "co" je důležitá informace v každém kanálu mapy příznaků a provádí agregaci prostorových informací dvěma způsoby [53]:

- **Průměrný pooling** extrahuje globální statistiky pomocí průměrné hodnoty [53].
- **Maximální pooling** extrahuje nejvýznamnější vlastnosti v mapě příznaků [53].

Matematicky lze kanálovou pozornost definovat jako

$$M_c(F) = \sigma(MLP(PrumerPool(F)) + MLP(MaxPool(F))), \quad (3.5)$$

kde  $\sigma$  je sigmoid funkce a  $F$  představuje vstupní mapu příznaků [53].

### Výpis 3.8: Modul kanálové pozornosti

```

1 class ChannelAttention(nn.Module):
2     def __init__(self, channels, reduction=16):
3         super(ChannelAttention, self).__init__()
4         self.global_avg_pool = nn.AdaptiveAvgPool2d(1)
5         self.global_max_pool = nn.AdaptiveMaxPool2d(1)
6
7         self.fc1 = nn.Linear(channels, channels // reduction,
8                               ↪ bias=False)
9         self.relu = nn.ReLU(inplace=True)
10        self.fc2 = nn.Linear(channels // reduction, channels,
11                              ↪ bias=False)
12
13        self.sigmoid = nn.Sigmoid()
14
15    def forward(self, x):
16        batch, channels, _, _ = x.shape
17        avg_out =
18            ↪ self.fc2(self.relu(self.fc1(self.global_avg_pool(x)
19            .view(batch, channels))))
20        max_out =
21            ↪ self.fc2(self.relu(self.fc1(self.global_max_pool(x)
22            .view(batch, channels))))
23        y = self.sigmoid(avg_out + max_out).view(batch, channels,
24            ↪ 1, 1)
25        return x * y

```

Prostorový modul se zaměřuje na to "kde" jsou důležité informace. Modul nejprve provádí, obdobně jako kanálový modul, průměrné a maximální pooling operace podél kanálové osy. Následně jsou tyto dva deskriptory zřetězeny a filtrovány konvoluční vrstvou rozměru 7x7, která generuje 2D prostorovou mapu pozornosti. Matematicky lze prostorovou pozornost definovat jako

$$M_s(F) = \sigma(f_{7 \times 7}([PrumerPool(F); MaxPool(F)])), \quad (3.6)$$

kde  $f_{7 \times 7}$  je konvoluce velikosti o rozměru 7x7 [53].

### Výpis 3.9: Modul prostorové pozornosti

```
1 class SpatialAttention(nn.Module):
2     def __init__(self):
3         super(SpatialAttention, self).__init__()
4         self.conv = nn.Conv2d(2, 1, kernel_size=7, padding=3,
5             ↪ bias=False)
6         self.sigmoid = nn.Sigmoid()
7
8     def forward(self, x):
9         avg_out = torch.mean(x, dim=1, keepdim=True)
10        max_out, _ = torch.max(x, dim=1, keepdim=True)
11        y = torch.cat([avg_out, max_out], dim=1)
12        y = self.sigmoid(self.conv(y))
13        return x * y
```

Typ CBAM implementovaný v použitém modelu vychází z původního návrhu, avšak místo samostatné MLP pro každý typ pooling operace sdílí stejnou sadu plně propojených vrstev pro obě operace. Tím je dosaženo nižší výpočetní náročnosti a zároveň je zachována schopnost modulu efektivně modelovat důležité znaky v datech.

### Výpis 3.10: CBAM

```
1 class CBAM(nn.Module):
2     def __init__(self, channels, reduction=16):
3         super(CBAM, self).__init__()
4         self.channel_attention = ChannelAttention(channels,
5             ↪ reduction)
6         self.spatial_attention = SpatialAttention()
7
8     def forward(self, x):
9         x = self.channel_attention(x)
10        x = self.spatial_attention(x)
11        return x
```

## 3.4.3 Absolutní rozdíl

Výstupy úzkých hrdel kodéru jsou přivedeny na vstup bloku absolutního rozdílu. Úkolem tohoto bloku je vypočítat rozdíl mezi extrahovanými rysy (features) z obou větví. Matematicky lze absolutní rozdíl definovat jako

$$D = |F_{T1} - F_{T2}|, \quad (3.7)$$

kde  $F_{T1}$  a  $F_{T2}$  jsou extrahované rysy ze snímků T1 a T2 po průchodu kodérem. Absolutní rozdíl ponechává pouze informaci o velikosti změny, směr změny (kladný

nebo záporný) nás nezajímá. Místo absolutního rozdílu lze použít i jiné, komplexnější metody, jako je např. konkatenace. Výpočet absolutního rozdílu je však rychlejší a výpočetně méně náročný. Výsledné rozdílové mapy jsou předány do dekodéru. V kódu je blok definován následovně:

Výpis 3.11: Definice absolutního rozdílu

```
1 diff = [torch.abs(ft1 - ft2) for ft1, ft2 in zip(features_t1,
    ↪ features_t2)]
```

### 3.4.4 Dekodér

Dekódovací část rekonstruuje masku změn pomocí postupné rekonstrukce prostorového rozlišení. Dekodér obsahuje čtyři rekonstrukční vrstvy. Každá vrstva interpoluje výstup na vyšší rozlišení a spojuje mapy příznaků z nižší vrstvy dekodéru. Jak bylo již zmíněno, v konvolučních blocích dekodéru je aplikována dropout regularizace. Posledním krokem je 1x1 konvoluce a sigmoid aktivace, která mapu transformuje na binární podobu [0,1].

Výpis 3.12: Vrstvy dekodéru

```
1 self.dec_conv4 = self.conv_block(1024 + 512, 512, use_cbam=False,
    ↪ use_dropout=True)
2 self.dec_conv3 = self.conv_block(512 + 256, 256, use_cbam=False,
    ↪ use_dropout=True)
3 self.dec_conv2 = self.conv_block(256 + 128, 128, use_cbam=False,
    ↪ use_dropout=True)
4 self.dec_conv1 = self.conv_block(128 + 64, 64, use_cbam=False,
    ↪ use_dropout=True)
5
6 self.final_conv = nn.Conv2d(64, 1, kernel_size=1)
```

## 3.5 Trénování modelu

Model byl trénován s pomocí následujících hardwarových zdrojů:

- **GPU** - NVIDIA GeForce RTX 4070 Ti SUPER 16GB
- **Procesor** - AMD Ryzen 7 1800X, 3600MHz, 8 jader, 16 logických procesorů
- **Paměť** - DIMM 4x16GB, 1333MHz
- **Disk** - KINGSTON SNV2S4000G

V tabulce 3.5 jsou uvedeny hyperparametry modelu, které byly během procesu učení nastavovány na níže uvedené hodnoty. Hyperparametry jsou nastavitelné parametry, které určují, jak se model učí, avšak nejsou odhadovány samotným učením,

ale člověkem - programátorem. Ovlivňují například rychlost konvergence, generalizaci nebo stabilitu trénování.

Tab. 3.2: Přehled hyperparametrů použitých při trénování modelu

Hyperparametr	Popis
learning_rate	Rychlost učení, určuje velikost aktualizací váh během trénování.
num_epochs	Maximální počet epoch, po které se model trénuje.
batch_size	Počet vzorků ve vstupní dávce během jedné iterace.
patience	Počet epoch bez zlepšení, než dojde k předčasnému zastavení.
min_delta	Minimální zlepšení metriky, které se považuje za významné.
criterion	Funkce ztráty pro binární klasifikaci s logity.
optimizer	Optimalizátor upravuje váhy modelu při učení.
weight_decay	Síla penalizace velkých vah v modelu.

### 3.5.1 První iterace

První iterace probíhala bez zapojení CBAM a dropout regularizace v konvolučním bloku. Pro trénování byl nasazen dataset, který byl předzpracován těmito kroky (viz 3):

- Převod na jednokanálovou reprezentaci (šedotónové snímky)
- Gaussovské rozostření
- CLAHE
- Min-max normalizace

Dataset byl dále rozdělen v poměru:

- **Trénovací data:** 39 032 dvojic
- **Validační data:** 7 806 dvojic
- **Testovací data:** 5 205 dvojic

Augmentace snímků nebyla použita. Model byl spuštěn s těmito hyperparametry:

- **Rychlost učení:** 0,001
- **Počet epoch :** 100
- **Velikost dávky:** 16
- **Trpělivost:** 10
- **Ztrátová funkce:** BCE
- **Optimalizátor:** Adam

Evaluační metriky modelu:

- **Přesnost:** 0,8439
- **citlivost:** 0,0534
- **F1-skóre:** 0,1004

- **IoU:** 0,0528
- **Pixelová přesnost:** 0,7687
- **Velikost:** 123,22 MB

Výsledek prvního pokusu byl neuspokojivý. Model přestal konvergovat přibližně po 30. epoše a následně byl zastaven mechanismem včasného ukončení. Původní hypotéza, která se později potvrdila, předpokládala, že některé kroky předzpracování dat odstranily důležité informace, což znemožnilo modelu efektivní extrakci příznaků. Problém modelu potvrzovaly i hodnoty výsledných metrik, které ukazují na výraznou nevyváženost mezi přesností a citlivostí. Vysoká přesnost při velmi nízké citlivosti naznačuje, že model označuje změny jen zřídka, ale pokud tak učiní, tak většinou správně. Nízká hodnota F1-skóre a IoU pak potvrzují, že model nedokáže efektivně zachytit většinu změn. Vysoká pixelová přesnost je zavádějící a odráží fakt, že většina pixelů v maskách změn zůstává bez změny.

### 3.5.2 Druhá iterace

Druhá iterace již probíhala se zapojením CBAM do úzkého hrdla kodéru a dropout regularizací v konvolučních blocích dekodéru s váhou 0,3. Po zkušenostech z první iterace byl dataset znovu předzpracován bez Min-Max normalizace, tedy s kroky:

- Převod na jednokanálovou reprezentaci (šedotónové snímky)
- Gaussovské rozostření
- CLAHE

Dataset byl také rozdělen v novém poměru, s větším důrazem na počet tréninkových dat:

- **Trénovací data:** 44 236 dvojic
- **Validační data:** 2 602 dvojic
- **Testovací data:** 5 205 dvojic

Augmentace snímků použita nebyla. Model byl spuštěn s těmito hyperparametry:

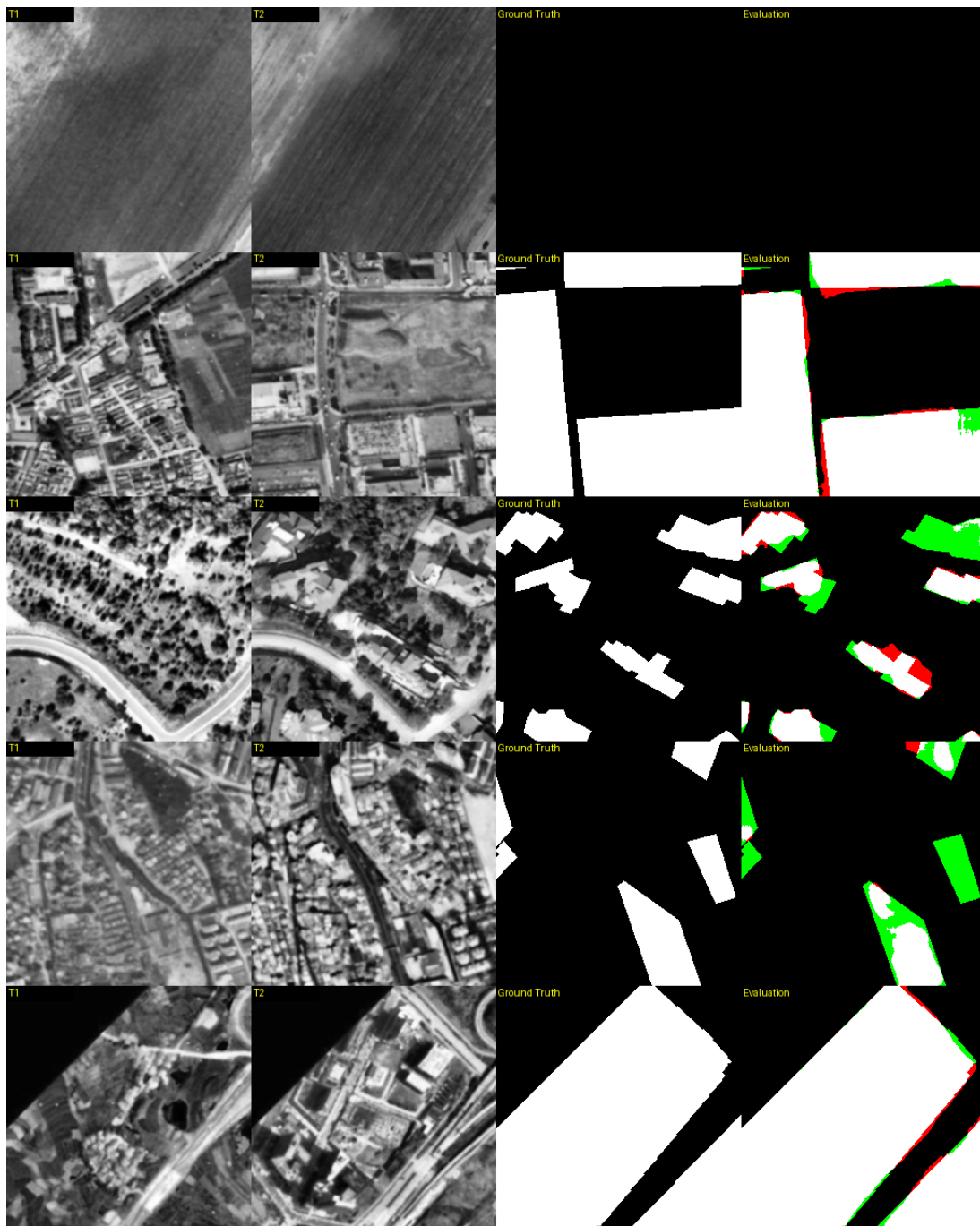
- **Rychlost učení:** 0,0001
- **Počet epoch :** 150
- **Velikost dávky:** 16
- **Trpělivost:** 10
- **Ztrátová funkce:** BCE
- **Optimalizátor:** Adam
- **Útlum vah:** 0,00005

Model byl zastaven mechanismem včasného ukončení při 87. epoše. Následně byl hodnocen pomocí evaluačních metrik:

- **Přesnost:** 0,9473

- **citlivost:** 0,9007
- **F1-skóre:** 0,9235
- **IoU:** 0,7794
- **Pixelová přesnost:** 0,9690
- **Velikost:** 123,22 MB

Model má poměrně vysokou přesnost. To znamená, že má nízkou falešnou pozitivitu a změny, které detekuje, jsou skutečné změny. Nižší citlivost než přesnost znamená, že model však občas skutečné změny přehlédne. Hodnota 0,9235 pro F1-skóre je velmi dobrá a naznačuje, že model dobře balancuje mezi přesností a pokrytím změn. Výsledek 0,7794 pro IoU je také poměrně dobrý, avšak tato hodnota závisí na přesnosti ruční segmentace změn v jednotlivých datasetech. Pixelová přesnost je velmi vysoká a naznačuje, že většina pixelů byla klasifikována správně. Tato metrika však nejméně ze všech vypovídá o kvalitě modelu, jelikož je ovlivňována vyvážeností datasetu (v našem případě 22,8% ku 77,2% viz 3.3). Velikost modelu (123,22MB) je poměrně velká. Pro případnou aplikaci modelu by to mohlo znamenat vyšší nároky na hardware a nevhodnost pro výpočetně slabší systémy. Na obr. 3.5 jsou uvedeny příklady detekce změn z testování modelu.



Obr. 3.5: Příklady detekce změn modelu s architekturou U-net na testovacích datech. Bílá - skutečně pozitivní (TP), černá - skutečně negativní (TN), červená - falešně pozitivní (FP), zelená - falešně negativní (FN).

## 3.6 Optimalizace a zjednodušení architektury

Vzhledem k poměrně velké velikosti modelu z druhé iterace je na místě dále se zabývat optimalizací a odlehčením modelu. Odlehčená nebo lehká neuronová síť je navržena tak, aby byla méně výpočetně náročná a co nejmenší velikosti, za které lze ještě dosáhnout pro nás akceptovatelné přesnosti modelu. Tyto modely mají redukováný počet konvolučních vrstev a menší počet parametrů, podrobněji byly lehké sítě popsány v kapitole 2.5.

Výpis 3.13: MobileNetV2 jako kodér

```
1 mobilenet = models.mobilenet_v2(  
2 weights=models.MobileNet_V2_Weights.IMAGENET1K_V1  
3 )  
4 self.encoder = mobilenet.features
```

Pro odlehčení naší neuronové sítě byl zvolen následující postup:

- Jako kodér použít efektivní neuronovou síť MobileNetV2
- Vyřadit modul pozornosti CBAM z úzkého hrdla
- Zmenšit počet filtrů v konvolučních vrstvách dekodéru

MobileNetV2 je efektivní neuronová síť navržena pro mobilní zařízení a obecně pro zařízení s omezeným výpočetním výkonem. Oproti předchozí verzi MobileNetV1 má MobileNetV2 několik zásadních inovací a vylepšení, které zlepšují poměr mezi přesností a výpočetní náročností modelu [54].

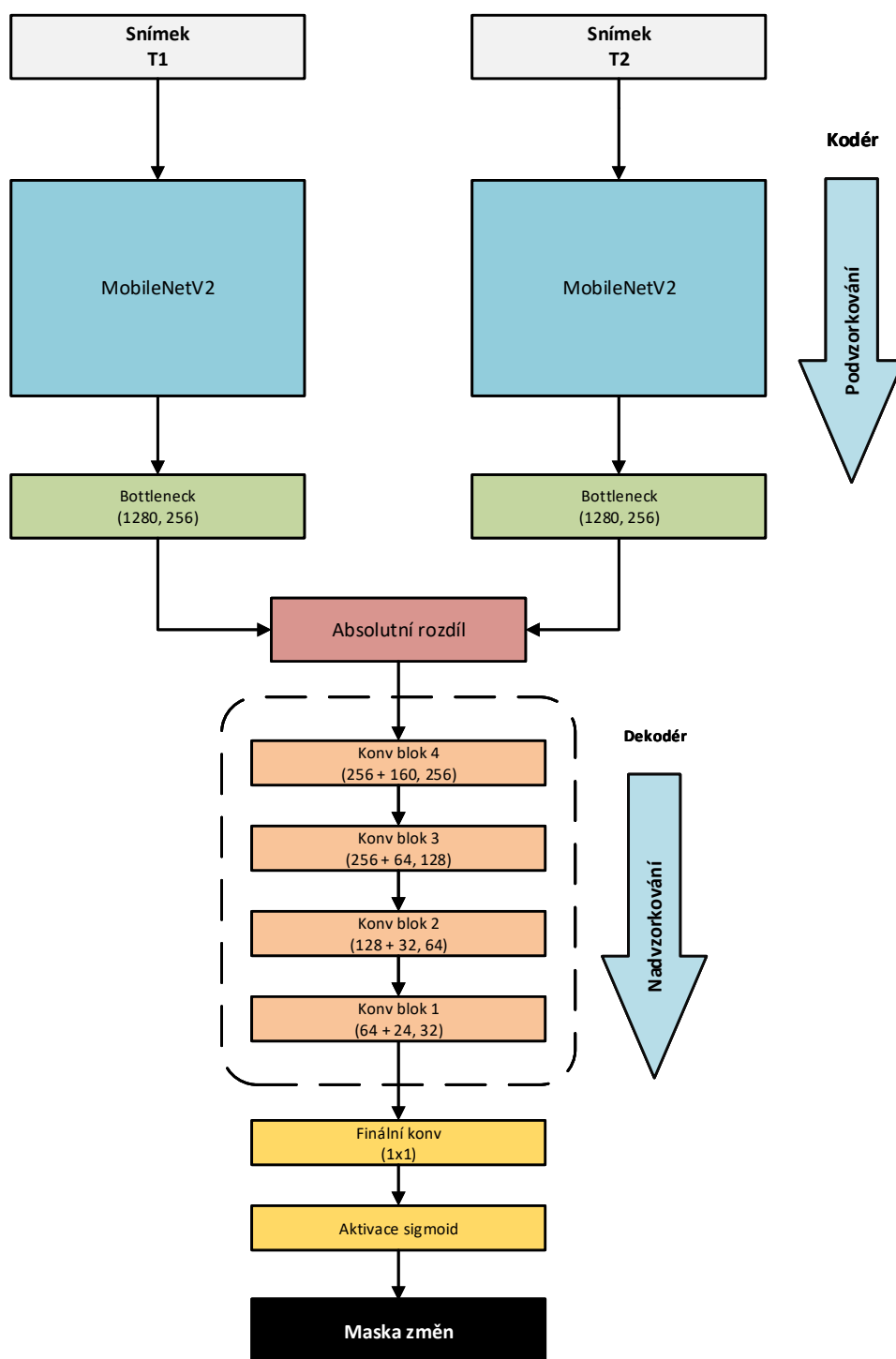
Jednou z hlavních změn v MobileNetV2 je zavedení invertovaných reziduálních bloků. Na rozdíl od tradičních reziduálních sítí, kde jsou vstupní a výstupní vrstvy široké a vnitřní vrstvy naopak úzké, MobileNetV2 používá opačný přístup. Vstupní a výstupní vrstvy jsou úzká hrdla, zatímco střední vrstva je širší a umožňuje efektivnější reprezentaci dat. Tento přístup zlepšuje schopnost sítě zachovávat informace a zároveň snižuje počet výpočtů potřebných pro inferenci [54].

Dalším důležitým prvkem MobileNetV2 je použití vrstev lineárního úzkého hrdla (linear bottleneck). Autoři modelu zjistili, že aplikace nelinearity, jako je ReLU aktivace, může v úzkých vrstvách vést ke ztrátě informace. Proto v těchto vrstvách odstranili nelineární transformace a použili lineární projekce, které umožňují lepší zachování prostorových informací a zároveň snižují riziko kolapsu reprezentací [54].

Model také využívá hloubkově separabilní konvoluce, což je technika, která rozděluje klasické konvoluce na dvě části: hloubkovou konvoluci (depthwise convolution), která provádí filtraci na jednotlivých kanálech zvlášť, a bodovou konvoluci (pointwise convolution), která kombinuje výsledky z různých kanálů. Tyto metody výrazně snižují počet operací potřebných ke zpracování obrazu, což umožňuje efektivnější výpočty i na méně výkonném hardwaru [54].

MobileNetV2 je používán v různých reálných aplikacích zpracování obrazu na mobilních telefonech nebo autonomních systémech. Díky své modularitě a nízkým nárokům na výpočetní výkon se stal jedním z preferovaných modelů pro implementaci neuronových sítí přímo na zařízeních bez nutnosti odesílání dat do cloudových serverů [54].

Blokové schéma odlehčeného modelu, který ve svém kodéru implementuje MobileNetV2, je zobrazeno na obrázku níže.



Obr. 3.6: Blokové schéma modelu implementující MobileNetV2.

Na rozdíl od U-net modelu má tento model místo čtyř konvolučních bloků u vstupu kodéru implementovanou architekturu MobileNetV2, jejíž výstup je předáván do úzkého hrdla s 1280 vstupními kanály a 256 výstupními. Jak bylo řečeno, modul pozornosti je z úzkého hrdla odstraněn. Struktura dekodéru je obdobná jako u předchozího modelu, počet kanálů konvolučních bloků je však zmenšen.

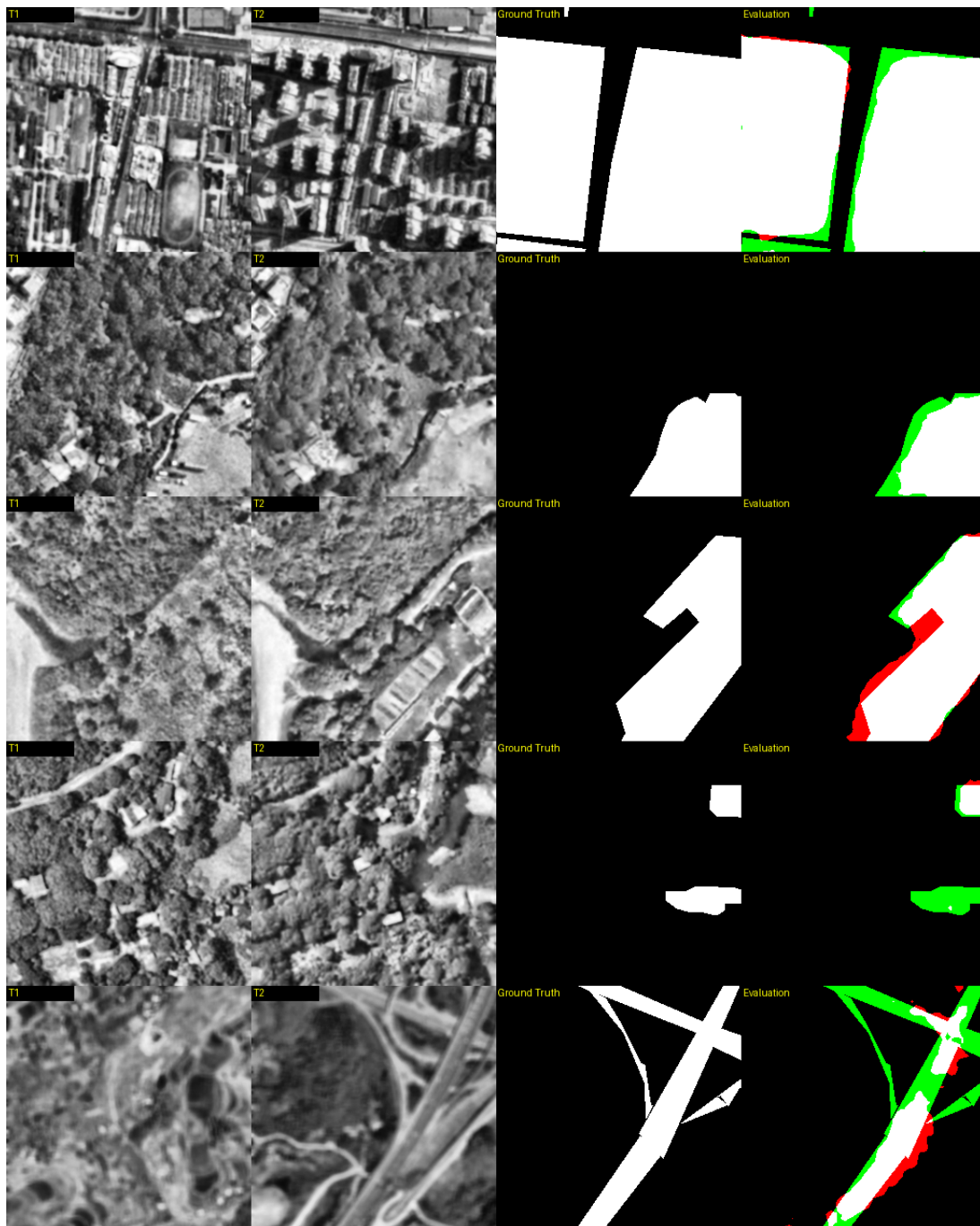
Pro trénování modelu byl použit totožný dataset jako u druhé iterace U-net. Augmentace snímků použita nebyla. Model byl trénován s těmito hyperparametry:

- **Rychlost učení:** 0,0001
- **Počet epoch :** 150
- **Velikost dávky:** 16
- **Trpělivost:** 10
- **Ztrátová funkce:** BCE
- **Optimalizátor:** Adam
- **Útlum vah:** 0,00005

Model byl zastaven mechanismem včasného ukončení po 62. epoše. Následně byl hodnocen pomocí evaluačních metrik:

- **Přesnost:** 0,9556
- **citlivost:** 0,8453
- **F1-skóre:** 0,8971
- **IoU:** 0,7090
- **Pixelová přesnost:** 0,9597
- **Velikost:** 10,63 MB

Pokud výsledky srovnáme s prvním modelem, všimneme si, že lehký model má o trochu lepší přesnost. Tedy o něco lépe eliminuje falešně pozitivní detekce. Ostatní metriky se však zhoršily. Model přehlíží více skutečných změn (vyšší podíl falešně negativních detekcí). Celková rovnováha mezi přesností a citlivostí reprezentovaná metrikou F1-skóre, se zhoršila. To ukazuje, že model upřednostňuje přesnost na úkor celkové správnosti. Výrazně klesla hodnota IoU, což znamená, že predikované oblasti se méně překrývají se skutečnými změnami, než tomu bylo u U-net modelu. Nižší je i pixelová přesnost. Přes většinové zhoršení metrik se však model radikálně zmenšil, a to na 10,63MB. Pokud tedy porovnáme výkonnost obou modelů s ohledem na jejich velikost, model s architekturou MobileNetV2 si vede velice dobře. Na obr. 3.7 jsou uvedeny příklady detekce změn z testování modelu.



Obr. 3.7: Příklady detekce změn modelu s MobileNetV2 kóděm na testovacích datech. Bílá - skutečně pozitivní (TP), černá - skutečně negativní (TN), červená - falešně pozitivní (FP), zelená - falešně negativní (FN).

## 3.7 Optimalizace hyperparametrů

Optimalizace hyperparametrů (Hyperparameter Optimization, HPO) je metodou pro konfiguraci hyperparametrů neuronové sítě, která se snaží minimalizovat chybu modelu na validační množině. HPO sdružuje několik metod, přičemž samotný proces optimalizace může být extrémně náročný na výpočetní prostředky a čas. HPO se obecně formuluje jako tzv. problém optimalizace černé skříňky, kde není znám analytický tvar cílové funkce [56].

Jednou z nejjednodušších a nejstarších metod optimalizace hyperparametrů je grid search, známý také jako full factorial design. Tato metoda spočívá ve vyhodnocení všech kombinací z konečných množin hodnot pro každý hyperparametr. I když je grid search jednoduchý na implementaci a snadno paralelizovatelný, trpí výrazným efektem zvaným prokletí dimenzionality, kdy počet kombinací roste exponenciálně s počtem hyperparametrů. [56].

Grid search má omezenou efektivitu zejména v případech, kdy má pouze malý počet hyperparametrů zásadní vliv na výkon modelu. V takovém případě je vhodnější použít náhodné prohledávání (random search), které pokrývá celý prostor rovnoměrněji a není vázáno na rigidní mřížku. Autoři metody uvádějí, že při pevném počtu evaluací  $B$  je počet hodnot, které grid search může efektivně prozkoumat na každém parametru roven pouze  $B^{1/N}$ , zatímco u random search může pokrýt až  $B$  různých kombinací bez této limitace [56].

I přes některé své nevýhody má grid search své místo při ladění modelu s malým počtem hyperparametrů nebo v případě, kdy je vyžadována replikovatelnost a jednoduchost implementace [56]. Náš optimalizovaný model s architekturou MobileNetV2 v kodéru je právě případem, kdy máme malý počet hyperparametrů a můžeme využít grid search pro doladění modelu a pokusit se zlepšit hodnoty výsledných metrik.

Pro grid search byly vytipovány tři rozdílné hodnoty pro rychlost učení ( $0,0001$ ;  $5 \times 10^{-5}$  a  $1 \times 10^{-5}$ ) a útlum vah ( $0,0001$ ;  $1 \times 10^{-5}$  a  $1 \times 10^{-6}$ ). To dělá celkem 9 kombinací, na kterých lze pozorovat konvergenci modelu na základě validační ztráty. Každá kombinace hyperparametrů byla trénována na 24 epochách, kdy na poslední epoše byla porovnávána validační ztráta viz tab. 3.7. Z grid search vyšly nejlepší hodnoty kombinace hyperparametrů  $5 \times 10^{-5}$  pro rychlost učení a  $0,0001$  pro útlum vah. Na základě tohoto zjištění byl model s MobileNetV2 kodérem znovu trénován s těmito hyperparametry:

- **Rychlost učení:** 0,00005
- **Počet epoch :** 100
- **Velikost dávky:** 16
- **Trpělivost:** 10
- **Ztrátová funkce:** BCE

- **Optimalizátor:** Adam
- **Útlum vah:** 0,00001

Po 62 epochách bylo trénování modelu zastaveno mechanismem včasného ukončení. Následná evaluace modelu poskytla tyto metriky:

- **Přesnost:** 0,9428
- **citlivost:** 0,8794
- **F1-skóre:** 0,9100
- **IoU:** 0,7383
- **Pixelová přesnost:** 0,9638
- **Velikost:** 10,63 MB

Z evaluace je patrné, že oproti původnímu modelu došlo k menšímu zhoršení přesnosti a zlepšení citlivosti, F1-skóre, IoU i pixelové přesnosti. Srovnání metrik všech modelů je přehledně zobrazeno v tab. 3.7.

Tab. 3.3: Validační ztráta pro různé kombinace rychlosti učení a útlumu vah

	Útlum vah		
Rychlost učení	$1 \times 10^{-5}$	$1 \times 10^{-6}$	0,0001
0,0001	0,1222	0,1214	0,1245
$5 \times 10^{-5}$	0,1233	0,1247	0,1200
$1 \times 10^{-5}$	0,1329	0,1281	0,1243

Validační ztráta

Tab. 3.4: Srovnání modelů

Architektura	Velikost	Doba učení	Přesnost	Citlivost	F1-skóre	IoU	Pix. př.
U-net	123,22 MB	87 epoch	0,9473	<b>0,9007</b>	<b>0,9235</b>	<b>0,7794</b>	<b>0,9690</b>
MobileNetV2	10,63 MB	62 epoch	<b>0,9556</b>	0,8453	0,8971	0,7090	0,9597
Opt. MobileNetV2	10,63 MB	62 epoch	0,9428	0,8794	0,9100	0,7383	0,9638

## 3.8 Srovnání predikce modelů

Po dokončení trénování modelů a jejich evaluace nadešel čas otestovat je na reálných datech a porovnat jejich výkonnost. Pro tento účel byl využit program Google Earth[55], který umožňuje zobrazovat družicové snímky v různých časových obdobích díky funkci časové osy. Tato vlastnost umožňuje sledovat změny v čase na konkrétních lokalitách. Bylo vybráno šest míst s různou mírou urbanistických změn.

Nejprve na obr. 3.8 porovnáme predikci těžkého modelu U-net s lehkým modelem s MobileNetV2 architekturou v kodéru.

První snímek (a) zobrazuje následky výbuchu síla 4. srpna 2020 v Bejrútu. Jedna z největších neatomových explozí v historii poškodila velkou část města. Dvojice snímků představuje nejbližší okolí síla před a po výbuchu. Oba modely zde shodně zaznamenaly největší změny v pravé dolní oblasti snímků, kde se nacházely plechové budovy přístavních skladů, které výbuch srovnal se zemí. U-net také správně označil oblast samotného síla. V tomto případě U-net vykazuje lepší citlivost.

Druhý snímek (b) zobrazuje stavbu nové budovy Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně (označeno červeně). Dvojice snímků zobrazuje lokalitu před zahájením a po dokončení stavby objektu. V tomto případě změny lépe zaznamenal model MobileNetV2.

Třetí snímek (c) zaznamenává stavbu nového úseku dálnice na Jižní Moravě u Starého Města u Uherského Hradiště. Nejpatrnější změnou na této dvojici snímků je sjezd dálnice (označeno červeně), který shodně zaznamenaly oba modely.

Čtvrtý pár snímků (d) zobrazuje rozvoj obchodního centra ve Starém Městě u Uherského Hradiště (označeno červeně). V tomto případě se jedná o dobře patrné změny v podobě nových budov, které byly postaveny na volném prostoru. Zde si oba modely vedly velmi dobře a masky změn zobrazují dobře rozeznatelné tvary budov. Zajímavostí je, že oba modely zaznamenaly přidání solárních panelů na budovu obchodního domu (označeno žlutě), kterou označili jako změnu.

Pátý pár (e) zobrazuje stavbu nových obytných domů na sídlišti Východ v Uherském Hradišti (označeno červeně). Zde oba modely správně zaznamenaly změny v podobě nových budov, U-net byl však v tomto případě citlivější a z jeho masky změn lze lépe rozeznat tvary jednotlivých budov.

Poslední dvojice snímků (f) představuje jemnou změnu pouze jedné budovy v husté zástavě v Brně Židenicích. Zde byl na ulici Gajdošova (označeno červeně) postaven nový obytný dům, který představuje jedinou změnu na tomto snímku. V tomto případě byl úspěšný model MobileNetV2, který změnu detekoval správně. Model U-net v daném místě detekoval změnu pouze v několika pixelech.

Na závěr ještě porovnáme predikce modelů s MobileNetV2 kodérem před a po optimalizaci hyperparametrů (na obrázku 3.8 označen jako (T) MobileNetV2).

Na prvním snímku (a) je patrné, že optimalizovaný model detekoval více změn než model původní.

Na druhém snímku (b) byl optimalizovaný model citlivější u detekce budovy v dolní části změn, kdy lze rozeznat i jednotlivá křídla budovy. Vedl si však hůře při detekci změn v horní části oblasti.

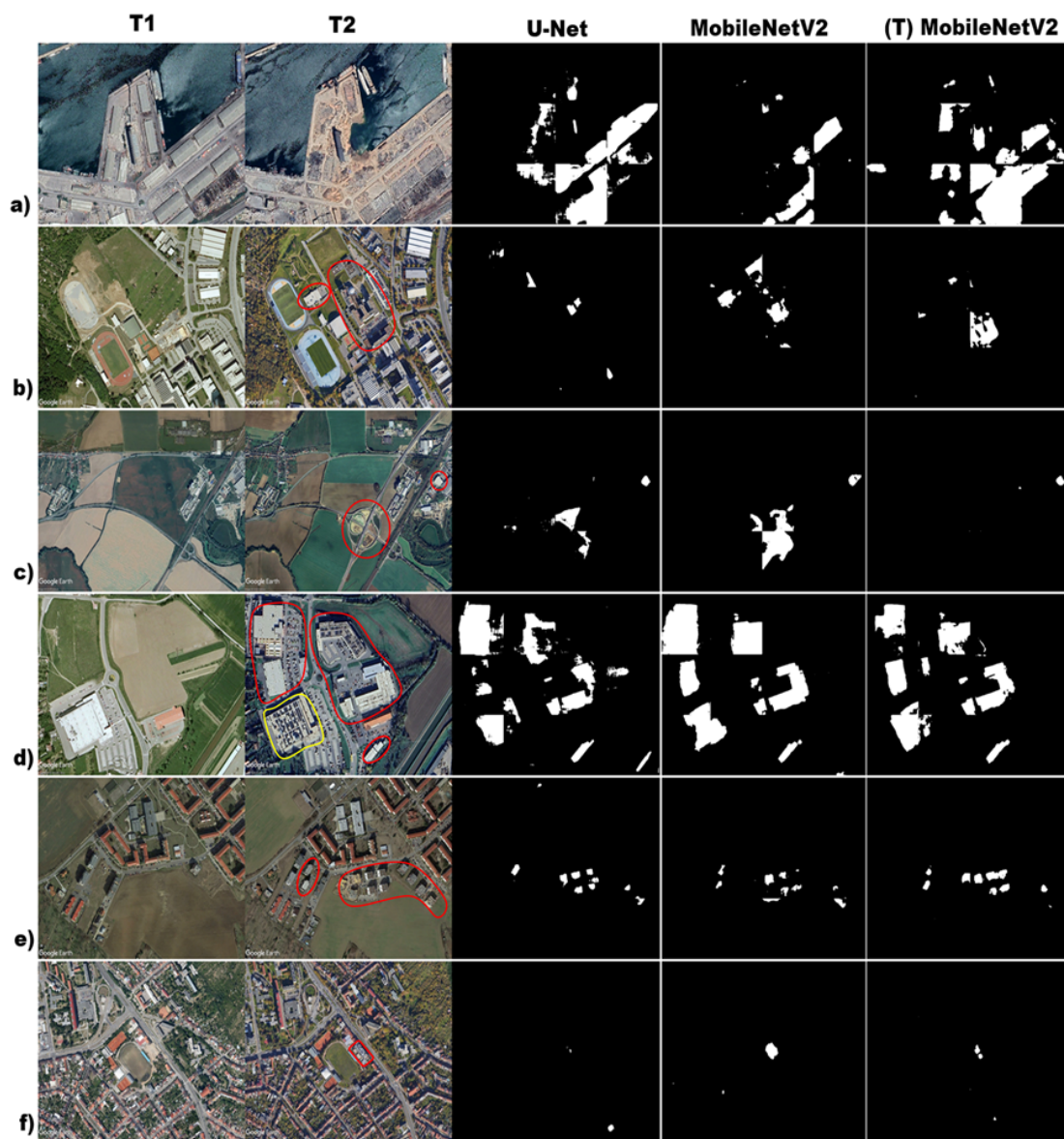
U třetího snímku (c) je rozdíl mezi detekcemi obou modelů markantní. Zde optimalizovaný model zcela selhal v detekci sjezdu dálnice.

Čtvrtý snímek (d) nového nákupního centra detekovaly oba modely víceméně shodně. Optimalizovaný model si však vedl o něco lépe, jelikož přibližně uprostřed oblasti změn dokázal detekovat menší budovu nalevo od kruhového objezdu, kterou původní model nedetekoval.

Na pátém snímku (e) dovedl optimalizovaný model více rozeznávat tvary nových obytných budov, avšak shodně s původním modelem nedokázal korektně detekovat čtyři nové budovy v pravé části oblasti změn.

U posledního snímku (e) je detekce obou modelů správná, kdy si optimalizovaný model zachoval schopnost detekovat i jemné změny v husté zástavbě.

Ze srovnání vyplývá, že optimalizace hyperparametrů vedla ve většině případů ke zlepšení detekční schopnosti modelu s MobileNetV2 kóděrem. Zejména v jemnější segmentaci a citlivější reakci na menší objekty. Optimalizovaný model byl schopný rozlišit některé detaily, které původní model přehlédl. Na druhou stranu se však ukázalo, že v některých scénářích dochází k větší chybovosti - konkrétně ve snímku (c) došlo ke zcela zásadnímu selhání detekce. Tyto výsledky ukazují, že ačkoli optimalizace může zlepšit průměrný výkon modelu, nezaručuje zlepšení napříč všemi typy scén.



Obr. 3.8: Srovnání predikce modelů: a) následky výbuchu sila v Bejrútu b) nová budova FEKT VUT v Brně c) sjezd nové dálnice u Uherského Hradiště d) obchodní centrum ve Starém Městě u Uherského Hradiště e) stavba nových bytových domů na sídlišti Východ v Uherském Hradišti f) stavba nového domu na ulici Gajdošova, Brno, Židenice

## Závěr

V rámci diplomové práce byla provedena rešerše na téma dálkového průzkumu Země a detekce změn na družicových snímcích. Byl popsán aktuální stav družicových systémů a popsány nejběžnější optické a mikrovlnné senzory využívané v dálkovém průzkumu. Dále byly popsány základní metody pro analýzu družicových snímků.

V další části byla popsána detekce změn pomocí neuronových sítí. Byla představena motivace a popsán obecný princip detekce změn. Následně byl vysvětlen princip neuronových sítí a popsány nejpoužívanější modely hlubokého učení. V návaznosti byla popsána problematika lehkých neuronových sítí. Dále byly představeny nejpoužívanější a aktuálně dostupné datasety pro úlohy detekce změn pomocí modelů hlubokého učení. Byly popsány používané postupy a architektury pro učení modelů s učitelem, s částečným učitelem a bez učitele, a byly diskutovány výhody a nevýhody těchto přístupů. Nakonec byly popsány evaluační metriky pro popis výkonnosti modelů hlubokého učení.

V praktické části diplomové práce bylo nejprve vybráno několik volně dostupných datasetů pro detekci změn. Ty byly následně předzpracovány pomocí jazyka Python s využitím knihovny OpenCV a sjednoceny do jednotného formátu. Dataset byl statisticky popsán a výsledky analýzy byly graficky zobrazeny a okomentovány.

Následně byl navržen první model neuronové sítě dle architektury U-net. Architektura modelu a jednotlivé funkční bloky byly podrobně popsány. První iterace modelu dopadla neúspěšně, načež byl problém týkající se předzpracování snímků odhalen a napraven. Druhá iterace byla již úspěšná a výkon modelu byl diskutován na základě evaluačních metrik.

Tento model byl následně optimalizován a upraven s cílem jeho odlehčení. K tomuto účelu byla zvolena lehká architektura MobileNetV2, která nahradila konvoluční bloky v kodéru původního U-netu. Výsledná velikost modelu se snížila z 123,22 MB na 10,63 MB, což představuje výraznou redukci, zatímco přesnost zůstala srovnatelná (0,9473 oproti 0,9556).

Optimalizovaná verze navíc při testování s různými kombinacemi hyperparametrů pomocí metody grid search dosáhla vyšší hodnoty F1-skóre (0,9100) a velmi dobré hodnoty přesnosti (0,9428). Oproti původnímu U-netu sice došlo k mírnému poklesu v citlivosti (0,8794 oproti 0,9007) a IoU (0,7383 oproti 0,7794), avšak tyto rozdíly jsou vyváženy výrazným snížením velikosti modelu a stále velmi dobrými výsledky.

Závěrečná srovnávací analýza predikcí jednotlivých modelů na časových snímcích reálných scén z Google Earth ukázala, že i přes mírné rozdíly v detekčním výkonu jsou výsledky optimalizované verze MobileNetV2 plně dostačující pro praktické využití, zejména v prostředích s omezenými výpočetními zdroji.

# Literatura

- [1] *Types of Earth observation satellites*. Online. In: JAPAN AEROSPACE EXPLORATION AGENCY. Earth-graphy. C2003. Dostupné z: <https://earth.jaxa.jp/en/eo-knowledge/eosatellite-type/index.html>. [cit. 2024-10-15].
- [2] *Earth observation satellites – Introduction*. Online. In: EUROPEAN SPACE AGENCY. ESA Eduspace. C2000-2014, 20 January 2011. Dostupné z: [https://www.esa.int/SPECIALS/Eduspace\\_EN/SEM7YN6SXIG\\_2.html](https://www.esa.int/SPECIALS/Eduspace_EN/SEM7YN6SXIG_2.html). [cit. 2024-10-15].
- [3] CRISP, N.H.; ROBERTS, P.C.E.; LIVADIOTTI, S.; OIKO, V.T.A.; EDMONDSON, S. et al. The benefits of very low earth orbit for earth observation missions. Online. *Progress in Aerospace Sciences*. 2020, roč. 117. ISSN 03760421. Dostupné z: <https://doi.org/10.1016/j.paerosci.2020.100619>. [cit. 2024-10-16].
- [4] ZHAO, Qiang; YU, Le; DU, Zhenrong; PENG, Dailiang; HAO, Pengyu et al. An Overview of the Applications of Earth Observation Satellite Data: Impacts and Future Trends. Online. *Remote Sensing*. 2022, roč. 14, č. 8. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs14081863>. [cit. 2024-10-17].
- [5] MASEK, Jeffrey G.; WULDER, Michael A.; MARKHAM, Brian; MCCORKEL, Joel; CRAWFORD, Christopher J. et al. Landsat 9: Empowering open science and applications through continuity. Online. *Remote Sensing of Environment*. 2020, roč. 248. ISSN 00344257. Dostupné z: <https://doi.org/10.1016/j.rse.2020.111968>. [cit. 2024-10-17].
- [6] BERGER, Michael; MORENO, Jose; JOHANNESSEN, Johnny A.; LEVELT, Pieter F. a HANSSEN, Ramon F. ESA's sentinel missions in support of Earth system science. Online. *Remote Sensing of Environment*. 2012, roč. 120, s. 84-90. ISSN 00344257. Dostupné z: <https://doi.org/10.1016/j.rse.2011.07.023>. [cit. 2024-10-17].
- [7] YUAN, Dongliang; ZHU, Jianrong; LI, Chunyan a HU, Dunxin. Cross-shelf circulation in the Yellow and East China Seas indicated by MODIS satellite observations. Online. *Journal of Marine Systems*. 2008, roč. 70, č. 1-2, s. 134-149. ISSN 09247963. Dostupné z: <https://doi.org/10.1016/j.jmarsys.2007.04.002>. [cit. 2024-10-17].
- [8] *Maxar Intelligence releases images from new WorldView Legion satellites*. Online. Intelligence Community News. 2024. Dostupné z:

<https://intelligencecommunitynews.com/maxar-intelligence-releases-images-from-new-worldview-legion-satellites/>. [cit. 2024-10-17].

- [9] SHI, Wenzhong; ZHANG, Min; ZHANG, Rui; CHEN, Shanxiong a ZHAN, Zhao. Change Detection Based on Artificial Intelligence: State-of-the-Art and Challenges. Online. Remote Sensing. 2020, roč. 12, č. 10. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs12101688>. [cit. 2024-09-28].
- [10] NAGEL, Gustavo Willy; NOVO, Evlyn Márcia Leão de Moraes a KAMPEL, Milton. Nanosatellites applied to optical Earth observation: a review. Online. *Ambiente e Agua - An Interdisciplinary Journal of Applied Science*. 2020, roč. 15, č. 3, s. 1-19. ISSN 1980-993X. Dostupné z: <https://doi.org/10.4136/ambiente-agua.2513>. [cit. 2024-10-16].
- [11] KULU, Erik. *World's largest database of nanosatellites*. Online. Nanosats database. C2014-2024. Dostupné z: <https://www.nanosats.eu>. [cit. 2024-10-16].
- [12] TATEM, Andrew; GOETZ, Scott a HAY, Simon. Fifty Years of Earth-observation Satellites. Online. American Scientist. 2008, roč. 96, č. 5. ISSN 0003-0996. Dostupné z: <https://doi.org/10.1511/2008.74.390>. [cit. 2024-09-28].
- [13] *PlanetScope*. Online. THE EUROPEAN SPACE AGENCY. Earth online. Dostupné z: <https://earth.esa.int/eogateway/missions/planetscope>. [cit. 2024-10-17].
- [14] How many Earth observation satellites are orbiting the planet in 2021? Online. In: Pixalytics. 18 August 2021. Dostupné z: <https://www.pixalytics.com/eosats-2021/>. [cit. 2024-09-28].
- [15] WIRATAMA, Wahyu a SIM, Donggyu. Fusion Network for Change Detection of High-Resolution Panchromatic Imagery. Online. Applied Sciences. 2019, roč. 9, č. 7. ISSN 2076-3417. Dostupné z: <https://doi.org/10.3390/app9071441>. [cit. 2024-10-06].
- [16] Novaspace: Earth Observation satellites set to triple over the next decade. Online. In: Satnews. 2024. Dostupné z: <https://news.satnews.com/2024/07/11/novaspace-earth-observation-satellites-set-to-triple-over-the-next-decade/#:~:text=This%20leading%20space%20consulting%20and%20market%20intelligence%20firm,the%20previous%20decade%20and%20marks%20a%20190%25%20increase..> [cit. 2024-09-28].

- [17] SINGH, Prabhishek; DIWAKAR, Manoj; SHANKAR, Achyut; SHREE, Raj a KUMAR, Manoj. A Review on SAR Image and its Despeckling. Online. *Archives of Computational Methods in Engineering*. 2021, roč. 28, č. 7, s. 4633-4653. ISSN 1134-3060. Dostupné z: <https://doi.org/10.1007/s11831-021-09548-z>. [cit. 2024-10-06].
- [18] KUMARI, Monika a KAUL, Ajay. Deep learning techniques for remote sensing image scene classification: A comprehensive review, current challenges, and future directions. Online. *Concurrency and Computation: Practice and Experience*. 2023, roč. 35, č. 22. ISSN 1532-0626. Dostupné z: <https://doi.org/10.1002/cpe.7733>. [cit. 2024-11-04].
- [19] YANG, Naisen a TANG, Hong. Semantic Segmentation of Satellite Images: A Deep Learning Approach Integrated with Geospatial Hash Codes. Online. *Remote Sensing*. 2021, roč. 13, č. 14. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs13142723>. [cit. 2024-11-05].
- [20] ADEGUN, Adekanmi Adeyinka; FONOU DOMBEU, Jean Vincent; VIRIRI, Serestina a ODINDI, John. State-of-the-Art Deep Learning Methods for Objects Detection in Remote Sensing Satellite Images. Online. *Sensors*. 2023, roč. 23, č. 13. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s23135849>. [cit. 2024-11-06].
- [21] HOLLOWAY, Jacinta a MENGERSEN, Kerrie. Statistical Machine Learning Methods and Remote Sensing for Sustainable Development Goals: A Review. Online. *Remote Sensing*. 2018, roč. 10, č. 9. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs10091365>. [cit. 2024-11-08].
- [22] PITONAK, Radoslav; MUCHA, Jan; DOBIS, Lukas; JAVORKA, Martin a MARUSIN, Marek. CloudSatNet-1: FPGA-Based Hardware-Accelerated Quantized CNN for Satellite On-Board Cloud Coverage Classification. Online. *Remote Sensing*. 2022, roč. 14, č. 13. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs14133180>. [cit. 2024-11-09].
- [23] SHAFIQUE, Ayesha; CAO, Guo; KHAN, Zia; ASAD, Muhammad a ASLAM, Muhammad. Deep Learning-Based Change Detection in Remote Sensing Images: A Review. Online. *Remote Sensing*. 2022, roč. 14, č. 4. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs14040871>. [cit. 2024-10-10].
- [24] CHENG, Guangliang; HUANG, Yunmeng; LI, Xiangtai; LYU, Shuchang; XU, Zhaoyang et al. Change Detection Methods for Remote Sensing in the Last Decade: A Comprehensive Review. Online. *Remote Sensing*. 2024, roč. 16, č. 13.

ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs16132355>. [cit. 2024-10-10].

- [25] SLESINSKI, Jakub; WIERZBICKI, Damian a KEDZIERSKI, Michal. Application of Multitemporal Change Detection in Radar Satellite Imagery Using REACTIV-Based Method for Geospatial Intelligence. Online. *Sensors*. 2023, roč. 23, č. 10. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s23104922>. [cit. 2024-10-10].
- [26] RŮŽIČKA, Vít; VAUGHAN, Anna; DE MARTINI, Daniele; FULTON, James; SALVATELLI, Valentina et al. RaVÆn: unsupervised change detection of extreme events using ML on-board satellites. Online. *Scientific Reports*. 2022, roč. 12, č. 1. ISSN 2045-2322. Dostupné z: <https://doi.org/10.1038/s41598-022-19437-5>. [cit. 2024-10-11].
- [27] STEWART, Matthew. *Introduction to Neural Networks: A detailed overview of neural networks with a wealth of examples and simple imagery*. Online. In: MEDIUM. Towards data science. 2019. Dostupné z: <https://towardsdatascience.com/simple-introduction-to-neural-networks-ac1d7c3d7a2c>. [cit. 2024-09-21].
- [28] IBM. *What is a neural network?* Online. Dostupné z: <https://www.ibm.com/topics/neural-networks>. [cit. 2024-09-21].
- [29] STEWART, Matthew. *Intermediate Topics in Neural Networks: A detailed overview of neural architecture, activation functions, loss functions, output units*. Online. In: MEDIUM. Towards data science. 2019. Dostupné z: <https://medium.com/towards-data-science/comprehensive-introduction-to-neural-network-architecture-c08c6d8e5d98>. [cit. 2024-09-21].
- [30] RASAMOELINA, Andrinandrasana David; ADJAILIA, Fouzia a SINCAK, Peter. A Review of Activation Function for Artificial Neural Network. Online. In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2020, s. 281-286. ISBN 978-1-7281-3149-8. Dostupné z: <https://doi.org/10.1109/SAMI48414.2020.9108717>. [cit. 2024-10-14].
- [31] RASCHKA, Sebastian. Online. In: Ahead of AI. 2016. Dostupné z: <https://sebastianraschka.com/>. [cit. 2024-09-23].
- [32] MUNRO, Paul. Backpropagation. Online. In: SAMMUT, Claude a WEBB, Geoffrey I. (ed.). *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2011, s. 73-73. ISBN 978-0-387-30768-8. Dostupné z: [https://doi.org/10.1007/978-0-387-30164-8\\_51](https://doi.org/10.1007/978-0-387-30164-8_51). [cit. 2024-09-22].

- [33] GOODFELLOW, Ian; BENGIO, Yoshua a COURVILLE, Aaron. *Deep learning*. Adaptive computation and machine learning series. Cambridge, MA: MIT press, [2016]. ISBN 0262035618.
- [34] LI, Pengzhi; PEI, Yan a LI, Jianqiang. A comprehensive survey on design and application of autoencoder in deep learning. Online. *Applied Soft Computing*. 2023, roč. 138. ISSN 15684946. Dostupné z: <https://doi.org/10.1016/j.asoc.2023.110176>. [cit. 2024-10-19].
- [35] VASWANI, Ashish, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N. GOMEZ, Łukasz KAISER a Illia POLOSUKHIN. *Attention is all you need*. In: Advances in Neural Information Processing Systems. 2017, vol. 30, s. 5998-6008. Dostupné z: <https://doi.org/10.48550/arXiv.1706.03762>. [cit. 2024-10-21]
- [36] LIN, Tianyang; WANG, Yuxin; LIU, Xiangyang a QIU, Xipeng. A survey of transformers. Online. *AI Open*. 2022, roč. 3, s. 111-132. ISSN 26666510. Dostupné z: <https://doi.org/10.1016/j.aiopen.2022.10.001>. [cit. 2024-10-21].
- [37] WANG, Lukang; ZHANG, Min; GAO, Xu a SHI, Wenzhong. Advances and Challenges in Deep Learning-Based Change Detection for Remote Sensing Images: A Review through Various Learning Paradigms. Online. *Remote Sensing*. 2024, roč. 16, č. 5. ISSN 2072-4292. Dostupné z: <https://doi.org/10.3390/rs16050804>. [cit. 2024-10-21].
- [38] M. SCHMIDT, Robin. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. Online. Computer Science. 23 Nov, 2019. Dostupné z: <https://doi.org/https://doi.org/10.48550/arXiv.1912.05911>. [cit. 2024-10-25].
- [39] GOODFELLOW, Ian; POUGET-ABADIE, Jean; MIRZA, Mehdi; XU, Bing; WARDE-FARLEY, David et al. Generative adversarial networks. Online. *Communications of the ACM*. 2020, roč. 63, č. 11, s. 139-144. ISSN 0001-0782. Dostupné z: <https://doi.org/10.1145/3422622>. [cit. 2024-10-31].
- [40] HSU, Chih-Chung; LIN, Chia-Hsiang; KAO, Chi-Hung a LIN, Yen-Cheng. DCSN: Deep Compressed Sensing Network for Efficient Hyperspectral Data Transmission of Miniaturized Satellite. Online. *IEEE Transactions on Geoscience and Remote Sensing*. 2021, roč. 59, č. 9, s. 7773-7789. ISSN 0196-2892. Dostupné z: <https://doi.org/10.1109/TGRS.2020.3034414>. [cit. 2024-11-02].
- [41] WANG, Ching-Hao; HUANG, Kang-Yang; YAO, Yi; CHEN, Jun-Cheng; SHUAI, Hong-Han et al. Lightweight Deep Learning: An Overview. Online.

- IEEE Consumer Electronics Magazine*. 2024, roč. 13, č. 4, s. 51-64. ISSN 2162-2248. Dostupné z: <https://doi.org/10.1109/MCE.2022.3181759>. [cit. 2024-11-01].
- [42] Online. OpenCV - Open Computer Vision Library. C2024. Dostupné z: <https://opencv.org/>. [cit. 2024-12-06].
- [43] GONZALEZ, Rafael C. a WOODS, Richard E. *Digital image processing*. Fourth edition. New York: Pearson, [2018]. ISBN 978-0-13-335672-4.
- [44] JAIN, Anil K. *Fundamentals of digital image processing*. Prentice-Hall International Editions. Englewood Cliffs: Prentice Hall, 1989. ISBN 0-13-332578-4.
- [45] CAMPOS, Gabriel Fillipe Centini; MASTELINI, Saulo Martiello; AGUIAR, Gabriel Jonas; MANTOVANI, Rafael Gomes; MELO, Leonimer Flávio de et al. Machine learning hyperparameter selection for Contrast Limited Adaptive Histogram Equalization. Online. *EURASIP Journal on Image and Video Processing*. 2019, roč. 2019, č. 1. ISSN 1687-5281. Dostupné z: <https://doi.org/10.1186/s13640-019-0445-4>. [cit. 2024-11-23].
- [46] HAN, Jiawei; KAMBER, Micheline a PEI, Jian. *Data mining: concepts and techniques*. 3rd ed. Amsterdam: Elsevier, c2011. ISBN 978-0-12-381479-1.
- [47] OTSU, Nobuyuki, et al. A threshold selection method from gray-level histograms. *Automatica*, 1975, 11.285-296: 23-27.
- [48] *Image Thresholding*. Online. OpenCV. 2024. Dostupné z: [https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html). [cit. 2024-11-25].
- [49] CHEN, Tao; LU, Zhiyuan; YANG, Yue; ZHANG, Yuxiang; DU, Bo et al. A Siamese Network Based U-Net for Change Detection in High Resolution Remote Sensing Images. Online. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2022, roč. 15, s. 2357-2369. ISSN 1939-1404. Dostupné z: <https://doi.org/10.1109/JSTARS.2022.3157648>. [cit. 2025-03-19].
- [50] PyTorch. PyTorch. [online]. Dostupné z: <https://pytorch.org/> [cit. 19. 3. 2025].
- [51] GeeksforGeeks. What is Batch Normalization in Deep Learning. [online]. 2021. Dostupné z: <https://www.geeksforgeeks.org/what-is-batch-normalization-in-deep-learning/> [cit. 19. 3. 2025].
- [52] GeeksforGeeks. Dropout in Neural Networks. [online]. 2021. Dostupné z: <https://www.geeksforgeeks.org/dropout-in-neural-networks/> [cit. 19. 3. 2025].

- [53] WOO, Sanghyun, PARK, Jongchan, LEE, Joon-Young a KWEON, In So. CBAM: Convolutional Block Attention Module [online]. Daejeon: Korea Advanced Institute of Science and Technology, 2018 [cit. 2025-03-19]. Dostupné z: <https://arxiv.org/abs/1807.06521>
- [54] SANDLER, Mark; HOWARD, Andrew; ZHU, Menglong; ZHMOGINOV, Andrey a CHEN, Liang-Chieh. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Online. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018, s. 4510-4520. ISBN 978-1-5386-6420-9. Dostupné z: <https://doi.org/10.1109/CVPR.2018.00474>. [cit. 2025-04-02].
- [55] GOOGLE LLC. Google Earth. Online. C2025. Dostupné z: <https://earth.google.com/web/>. [cit. 2025-04-26].
- [56] FEURER, Matthias a HUTTER, Frank. Hyperparameter Optimization. Online. In: HUTTER, Frank; KOTTHOFF, Lars a VANSCHOREN, Joaquin (ed.). Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Cham: Springer International Publishing, 2019, s. 3-33. ISBN 978-3-030-05317-8. Dostupné z: <https://doi.org/10.1007/978-3-030-05318-1>. [cit. 2025-05-07].

# Seznam symbolů a zkratek

- AI** Umělá inteligence - Artificial Intelligence
- CBAM** Konvoluční blok s mechanismem pozornosti - Convolutional Block Attention Module
- CD** Detekce změn - Change Detection
- CE** Chyba vzájemné entropie - Cross Entropy Error
- CLAHE** Adaptivní ekvalizace histogramu s omezeným kontrastem - Contrast Limited Adaptive Histogram Equalization
- CNN** Konvoluční neuronová síť - Convolutional Neural Network
- cRMSE** Změněná odmocnina střední čtvercové chyby - Changed Root Mean Squared Error
- DCSN** Hluboká síť pro komprimované snímání - Deep Compressed Sensing Network
- DEM** Digitální výškový model - Digital Elevation Model
- ELU** Exponenciální lineární jednotka - Exponential Linear Unit
- EO** Pozorování země - Earth Observation
- FCN** Plně konvoluční síť - Fully Convolutional Networks
- FFN** Dopředná neuronová síť - Feed Forward Network
- FN** Falešně negativní - False Negative
- FP** Falešně pozitivní - False Positive
- FPGA** Programovatelná hradlová pole - Field Programmable Gate Array
- GAN** Generativní adversiální síť - Generative Adversarial Network
- GIS** Geografický informační systém - Geographic Information System
- GPU** Grafický procesor - Graphics Processing Unit
- HPO** Optimalizace hyperparametrů - Hyperparameter Optimization
- IoT** Internet věcí - Internet of Things
- IoU** Průnik přes sjednocení - Intersection Over Union
- KC** Kappa koeficient - Kappa Coefficient
- LNN** Lehká neuronová síť - Light Neural Network
- MB** Megabajt - Megabyte
- MLP** Vícevrstvý perceptron - Multi-Layer Perceptron
- MODIS** Spektroradiometr s mírným rozlišením - Moderate Resolution Imaging Spectroradiometer
- NAS** Hledání architektury neuronové sítě - Neural Architecture Search
- NDVI** Normalizovaný rozdílový vegetační index - Normalized Difference Vegetation Index
- OA** Celková přesnost - Overall Accuracy
- PCA** Analýza hlavních komponentů - Principal Component Analysis

**PSPNet** Pyramidové zpracování scény - Pyramid Scene Parsing Network  
**ReLU** Rektifikovaná lineární jednotka - Rectified Linear Unit  
**RGB** Červená zelená modrá - Red Green Blue  
**RNN** Rekurentní neuronová síť - Recurrent Neural Network  
**RMSE** Odmocnina střední čtvercové chyby - Root Mean Squared Error  
**RS** Vzdálené snímání - Remote Sensing  
**RSISC** Metody klasifikace scén na družicových snímcích - Remote Sensing Image Classification  
**SAR** Syntetická apertura radaru - Synthetic Aperture Radar  
**SSE** Součet čtvercových chyb - Summed Squared Error  
**SVM** Stroj podpůrných vektorů - Support Vector Machine  
**SVR** Podporované regresivní modely - Support Vector Regression  
**TN** Skutečně negativní - True Negative  
**TP** Skutečně pozitivní - True Positive  
**ViT** Vizuální transformer - Vision Transformer  
**2D** Dvoj-rozměrný - Two-Dimensional

# Seznam příloh

A Obsah elektronické přílohy

88

# A Obsah elektronické přílohy

Dataset použitý v diplomové práci je dostupný na externím úložišti na tomto odkaze.

```
přílohy/  
├── předzpracování/  
│   ├── image_preprocess.ipynb  
│   └── statistics.ipynb  
├── sítě/  
│   ├── mobilenet/  
│   │   ├── model/  
│   │   │   └── siamese_unet.py  
│   │   ├── modules/  
│   │   │   ├── dataset.py  
│   │   │   ├── early_stop.py  
│   │   │   └── utils.py  
│   │   ├── eval.py  
│   │   ├── grid_search.py  
│   │   ├── requirements.txt  
│   │   ├── test.py  
│   │   └── train.py  
│   └── unet/  
│       ├── model/  
│       │   ├── cbam.py  
│       │   └── siamese_unet.py  
│       ├── modules/  
│       │   ├── dataset.py  
│       │   ├── early_stop.py  
│       │   └── utils.py  
│       ├── eval.py  
│       ├── requirements.txt  
│       ├── test.py  
│       └── train.py
```