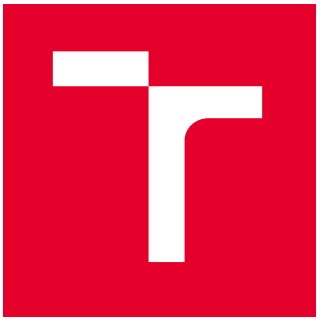


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

SÉMANTICKÁ SEGMENTACE LETECKÝCH SNÍMKŮ

SEMANTIC SEGMENTATION OF AERIAL IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — **Jiří Pazdera**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. Roman Adámek**

SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	**Jiří Pazdera**
Studijní program:	Mechatronika
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. Roman Adámek**
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Sémantická segmentace leteckých snímků

Stručná charakteristika problematiky úkolu:

Sémantická segmentace obrazových dat slouží k rozdělení jednotlivých částí obrazu do předem definovaných tříd na základě jejich významu. V dnešní době se s ní setkáváme hlavně ve spojení s konvolučními neuronovými sítěmi.

V případě sémantické segmentace leteckých snímků nám jde o rozčlenění snímku do kategorií odpovídající například cestám, polím, vodním tokům, zástavbě atd. V tomto případě budou letecké snímky využity pro hledání cesty pro pohyb mobilního robotu v oblastech, kde nejsou k dispozici jiné mapové podklady nebo již nejsou aktuální. Klíčem tedy bude najít ve snímku cesty, jejich typ a potom další průjezdné a neprůjezdné oblasti.

Cíle bakalářské práce:

1) Proveďte rešerši současného stavu poznání v oblasti sémantické segmentace leteckých snímků. Do rešerše zahrňte metody a dostupné datové sady anotovaných snímků použitelné pro testování.

2) Na základě rešerše zvolte alespoň dvě před trénované nejperspektivnější metody, které porovnáte vůči sobě na dostupných sadách leteckých snímků. Srovnejte metody na základě přesnosti klasifikace a výpočetního výkonu. Použijte letecké snímky z různých oblastí (městská zástavba, zemědělská krajina, lesy, ...) a různých ročních období.

3) Proveďte experiment, ve kterém natrénujete existující neuronové sítě na základě kategorií klasifikace zaměřené hlavně pro plánování cesty vozidla.

4) Demonstrujte plánování cesty ve výsledných segmentovaných snímcích s využitím algoritmu A*.

Seznam doporučené literatury:

[1] CHOLLET, François. 2019. Deep learning v jazyku Python: knihovny Keras, Tensorflow. Přeložil Rudolf PECINOVSKÝ. Praha: Grada Publishing. Knihovna programátora (Grada).

[2] De Marchi, Leonardo, and Laura Mitchell. 2019. Hands-On Neural Networks: Learn How to Build and Train Your First Neural Network Model Using Python. Birmingham, England: Packt Publishing.

[3] Michelucci, Umberto. 2019. Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection. 1st ed. Berlin, Germany: APress.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc. ředitel ústavu	doc. Ing. Jiří Hlinka, Ph.D. děkan fakulty

Abstrakt

Tato práce se zabývá sémantickou segmentací leteckých snímků a jejich následným využitím pro plánování trasy zachyceným terénem. První část představuje úvod do dané problematiky a teoretický popis současného stavu poznání. Část druhá popisuje testování dostupných segmentačních metod, vývoj vlastní datové sady a trénování existujícího modelu neuronové sítě. Na závěr je demonstrována možnost plánování trasy pomocí vhodného algoritmu.

Summary

This work deals with semantic segmentation of aerial images and their subsequent use for route planning. The first part represents an introduction to this issue and a theoretical description of the current state of knowledge. The second part describes testing of available segmentation methods, the development of custom dataset, and the training of an existing neural network model. Finally, the possibility of route planning using an appropriate algorithm is demonstrated.

Klíčová slova

Sémantická segmentace, datová sada, letecký snímek, neuronová síť, anotace, plánování trasy, algoritmus A*

Keywords

Semantic segmentation, dataset, aerial image, neural network, anotation, route planning, A* algorithm

Bibliografická Citace

PAZDERA, Jiří. Sémantická segmentace leteckých snímků [online]. Brno, 2024 [cit. 2024-05-24]. Dostupné z: https://www.vut.cz/studenti/zav-prace/detail/157781. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Roman Adámek.

Prohlašuji, že předloženou bakalářskou práci jsem zpracoval samostatně, pouze na základě vlastních poznatků, poznámek vedoucího a níže citovaných zdrojů.

Jiří Pazdera

Brno . . . .

Děkuji vedoucímu této bakalářské práce Ing. Romanu Adámkovi za poskytovanou zpětnou vazbu a vstřícný přístup v průběhu celého řešení této práce.

Jiří Pazdera

Obsah

1 Úvod

Sémantická segmentace je jednou z mnoha kategorií strojového vidění, které umožňuje počítačům extrahovat relevantní informace z obrazových dat. Hlavní charakteristikou tohoto přístupu je rozřazení všech pixelů vstupního snímku do předem stanoveného počtu klasifikačních tříd.

V současné době zažíváme velký rozmach umělé inteligence, který úzce souvisí se stále lepší dostupností výpočetního výkonu. Na sémantickou segmentaci má tento pokrok zcela zásadní dopad, jelikož umělá inteligence v podobě neuronových sítí umožňuje tvorbu algoritmů s nevídanou mírou adaptability. Automatické mapování zemského povrchu, na které je tato práce zaměřena, je pouze jednou z mnoha domén sémantické segmentace. K dynamickému rozvoji tohoto přístupu dochází dále například ve zpracování biomedicínských snímků nebo v oblasti autonomního řízení vozidel.

Hlavním cílem této práce je prozkoumat dostupné možnosti současných metod pro segmentaci leteckých snímků s důrazem na správnou identifikaci pozemních komunikací a následné plánování trasy pro mobilního robota. Úkolem první části je seznámit čtenáře s tradičními i moderními metodami sémantické segmentace včetně odpovídajících datových sad a některých evaluačních metrik. Nalezené perspektivní metody budou dále otestovány a porovnány z hlediska kvality segmentace a výpočetní náročnosti. Dalším dílčím cílem je potom tvorba vlastní datové sady a experiment s trénováním existujícího modelu neuronové sítě tak, aby bylo možné rozlišovat více druhů cest. Na základě uvedeného postupu by potom mělo dojít ke správné identifikaci průjezdných oblastí terénu, které poslouží k plánování trasy pro mobilního robota pomocí vhodně zvoleného algoritmu.

2 Rešerše

Tato kapitola představuje průzkum současného stavu poznání v oblasti sémantické segmentace obrazu a poskytuje základní přehled o používaných metodách, datových sadách, evaluačních metrikách a dále popisuje algoritmus pro plánování trasy segmentovaným snímkem.

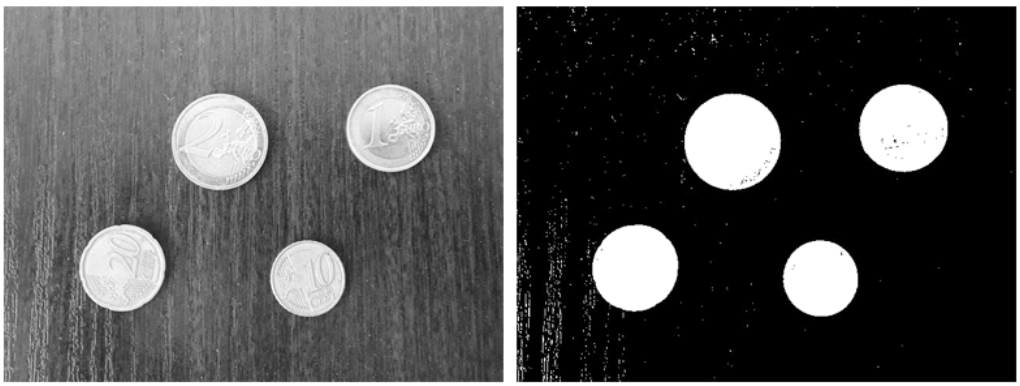
2.1 Tradiční metody

Pojmem tradiční metody souhrnně označujeme veškeré algoritmy sémantické segmentace, které nevyužívají pro zpracování vstupního snímku neuronové sítě. Tyto postupy jsou obvykle založeny na pevně daných kritériích a jejich použití je omezeno na řešení úzkého spektra úkolů. Právě nízká adaptabilita je významnou nevýhodou těchto přístupů oproti moderním metodám, které využívají neuronových sítí. Některé tradiční metody jsou ovšem velmi jednoduché a nenáročné na výpočetní výkon, proto se stále využívají v aplikacích, kde nízká přizpůsobivost není překážkou.

2.1.1 Prahování

Tato metoda patří mezi nejjednodušší algoritmy pro segmentaci obrazu. Vstupem bývá typicky černobílý obrázek a hodnota zvoleného prahu, který reprezentuje určitý stupeň šedi. Veškeré pixely jsou potom rozřazeny do dvou skupin podle toho, jestli jejich hodnota na černobílé škále přesahuje hodnotu prahu či nikoli [1]. Výstupem je potom binární maska, která nese informaci o rozdělení pixelů do dvou kategorií.

Kromě této základní aplikace existuje i mnoho modifikací, které umožňují zpracování jiných než černobílých snímků.



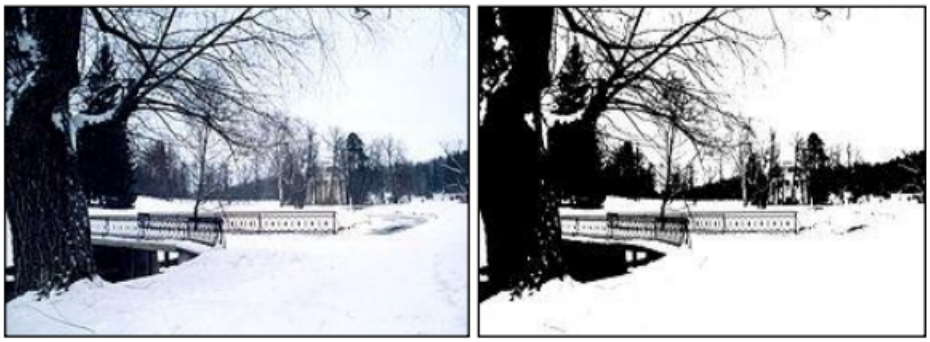
Obrázek 2.1: Příklad prahování, zdroj: [2]

2.1.2 Shlukování

Tato metoda slouží k rozdělení všech pixelů na vstupním snímku do zvoleného počtu shluků. Rozdělení je provedeno na základě zvolených vlastností.

Samotný algoritmus shlukování můžeme rozdělit do několika kroků. Na počátku je nutné pro každý pixel provést extrakci atributů, kterými můžou být např. barva, textura nebo umístění na vstupním snímku. Každý pixel je potom reprezentován vektorem těchto vlastností, který udává jeho pozici v mnoho rozměrném prostoru [3]. V tomto prostoru zvolíme souřadnice obecně K bodů, které reprezentují středy jednotlivých shluků. Souřadnice těchto bodů volíme buď náhodně, nebo na základě předchozích zkušeností. Pro každý bod prostoru je potom vypočtena jeho vzdálenost od vytvořených středů a je přiřazen ke středu s minimální vzdáleností. Proces opakujeme dokud zbývá alespoň jeden nepřiřazený pixel. Souřadnice středů jsou průběžně aktualizovány na základně průměrných vlastností přiřazených pixelů [1].

Na konci tohoto algoritmu jsou tedy veškeré pixely vstupního snímku rozděleny do K tříd podle zvolených vlastností.



Obrázek 2.2: Příklad shlukování, zdroj: [1]

2.1.3 Narůstání oblastí

Metoda narůstání oblasti (region growing) je založena na volbě výchozího pixelu a postupném přiřazování sousedních pixelů na základě podobnosti.

Prvním krokem této metody je určení výchozího pixelu, který můžeme zvolit buď náhodně, nebo manuálně dle vlastního výběru. Algoritmus následně zjistí vlastnosti osmi okolních pixelů a postupně je porovnává s výchozím. V případě, že vlastnosti splňují požadovanou míru podobnosti, je zkoumaný pixel ztotožněn s pixelem výchozím. Tento proces se opakuje pro každou nově přiřazenou jednotku, a dochází tak k narůstání homogenní oblasti. Růst oblasti je ukončen v okamžiku, kdy algoritmus nenachází žádný další pixel, jehož vlastnosti by byly dostatečně podobné s rozšiřující se oblastí. V takovém případě ukončujeme dosavadní proces a volíme nový výchozí pixel. Obvykle se jedná o ten, který jako první nevyhovoval požadovaným vlastnostem a nebyl přiřazen k dané oblasti [4].

Popsaný algoritmus potom pokračuje až do kompletní segmentace vstupního snímku do jednotlivých homogenních oblastí.

Obrázek 2.3: Příklad narůstání oblasti, zdroj: [5]

2.2 Neuronové sítě

Neuronové sítě, jakožto podmnožina umělé inteligence, patří v současné době mezi nejrychleji se vyvíjející oblasti ve světě moderních technologií. Kromě populárních jazykových modelů nachází neuronové sítě hojné využití v mnoha oblastech vědy a průmyslu, mezi které lze zařadit např. výzkum v medicíně a farmacii, optimalizaci vyhledávání, autonomní řízení vozidel či strojové vidění.

Klíčovou výhodou neuronových sítí v oblasti sémantické segmentace obrazu je jejich schopnost přizpůsobovat se různým prostředním, aniž by bylo nutné zasahovat do architektury samotné sítě. Během procesu trénování dostává model neuronové sítě k dispozici množinu dat, která obsahuje originální snímky i anotované masky, které mají význam požadovaného výstupu. Model je potom schopen pomocí známých mechanismů [6] sám přizpůsobit svoje vlastnosti tak, aby co nejlépe segmentoval vstupní snímky na požadovaný výstup.

Na rozdíl od tradičních metod, které často vyžadují manuální specifikaci konkrétních kritérií, jsou neuronové sítě schopny samy určovat, které atributy vstupních dat budou brát v potaz a jakou váhu jim budou přisuzovat. Pro sémantickou segmentaci obrazových dat jsou navíc využívány především konvoluční neuronové sítě, které během trénování pracují s vysoce abstraktními rysy. V praxi to znamená schopnost komplexního porozumění snímku a vnímání určitého kontextu [6]. Bližší charakteristiku konvolučních neuronových sítí je možné dohledat například v článku [7].

Značnou nevýhodou tohoto přístupu k sémantické segmentaci je citlivost učení na množství a kvalitu trénovacích dat. Žádný model nelze dostatečně dobře natrénovat, nemáme-li k dispozici dostatečné množství kvalitně anotovaných dat [6].

Další nevýhodou jsou potom vysoké nároky na výpočetní výkon během trénování a to zejména u složitých modelů a velkých datových sad. Trénování ovšem stačí správně provést pouze jednou a nadále už používat pouze natrénovaný model, který na výpočetní výkon není tolik náročný.

Výše uvedené nevýhody neuronových sítí je možné zmírnit pouhým laděním předtrénovaných modelů. Jedná se o techniku, která vyžaduje model natrénovaný na velké datové sadě, ta musí být svou povahou podobná jako řešený problém. Takto natrénované modely bývají často volně dostupné na internetových stránkách jako github.com nebo modelzoo.com. Získaný model je potom možné dotrénovat na specifické datové sadě, která by sama o sobě nebyla pro trénování dostatečná.

V následujících sekcích této podkapitoly jsou popsány některé významné architektury využívané v oblasti sémantické segmentace.

2.2.1 U-Net

Architektura této konvoluční neuronové sítě byla poprvé představena ve vědeckém článku [8], který byl publikován trojicí autorů v roce 2015. Název U-net byl odvozen z jejího tvaru, který připomíná písmeno U. Schématické znázornění této architektury je zobrazeno na obrázku 2.4.

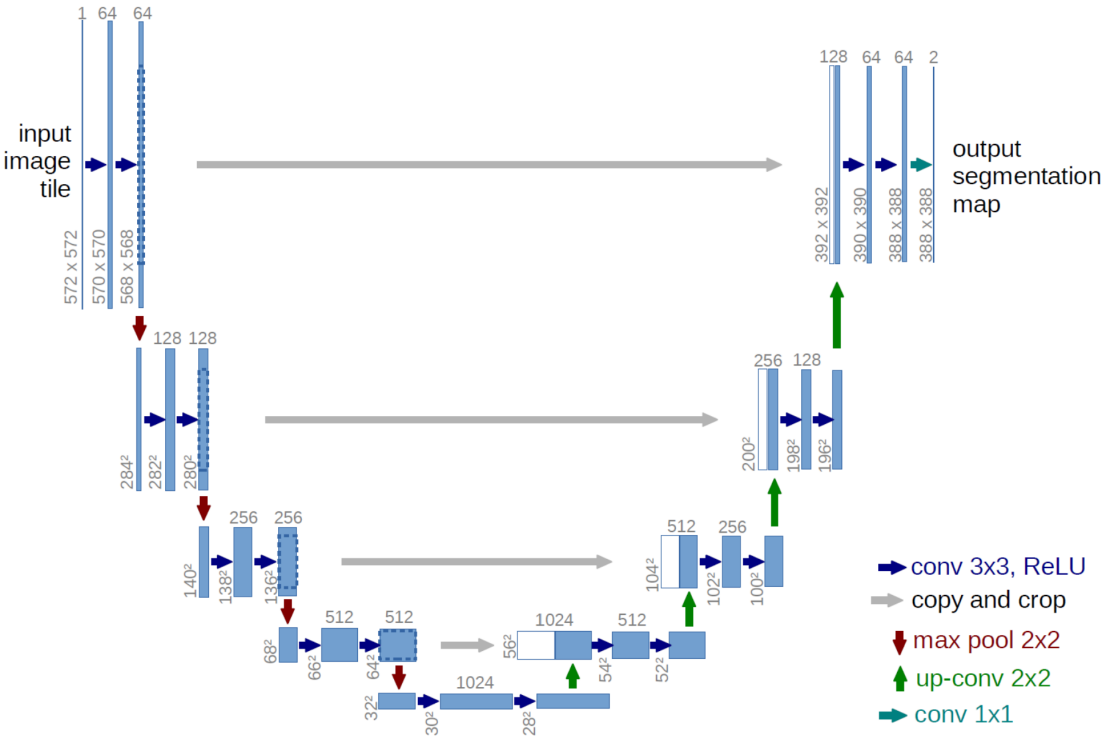
Původním účelem představeného modelu bylo segmentovat biomedicínské snímky, hojné využití ovšem záhy našel i v jiných oblastech sémantické segmentace. Ve své kategorii se jedná o velmi rozšířený a často využívaný model [9].

Architekturu této sítě lze rozdělit na dvě základní části, tedy na enkodér a dekodér. Jako enkodér označujeme část modelu, kterou na obrázku 2.4 můžeme vidět na levé straně. Do této části vstupuje snímek, jehož segmentaci chceme provést, a je postupně transformován řadou operací.

V každém stupni enkodéru jsou data nejprve podrobena konvoluci, která zvyšuje počet jeho kanálů, tím postupně dochází k nárůstu míry abstrakce učitelných prvků [10]. V případě uvedeného schématu dochází při každé konvoluci k drobnému zmenšení rozměrů vstupních dat, často se ovšem používá modifikace, která zachovává původní rozlišení a mění pouze počet kanálů, tedy třetí rozměr tenzoru dat.

Po konvoluci následuje aktivační funkce ReLU, která slouží k saturaci hodnot z předchozí operace a tím zavádí do modelu nelineární prvek. Značnou výhodou ReLU oproti některým jiným aktivačním funkcím je, že při procesu zpětné propagace nezpůsobuje problém mizejícího gradientu [11].

Poslední operací v každém stupni enkodéru je potom maxpooling. Při této operaci dochází ke snižování rozlišení zpracovávaného snímku. Hlavním efektem je snižování komplexity z hlediska výpočetního výkonu [11] při zachování podstatného kontextu.



Obrázek 2.4: Architektura U-Net, zdroj: [8]

Poté, co snímek projde zpracováním v enkodéru, vstupuje do dekodéru, kde je řadou konvolučních operací postupně zvyšováno jeho rozlišení až na původní hodnoty a zároveň je redukován počet jeho kanálů. Za každou konvoluční operací opět následuje aktivační funkce ReLU. Konečný počet kanálů odpovídá celkovému počtu tříd, které je daná síť schopna rozlišovat.

Mezi stranou enkodéru a dekodéru jsou taktéž tzv. skip connections, která jsou ve schématu zaznačena šedými šipkami. Tato spojení jsou typická pro architekturu U-net a slouží k přenosu dat mezi odpovídajícími stupni obou hlavních částí sítě. Tato operace vede ke zlepšení kvality detailu ve výsledném segmentovaném snímku.

Velkou výhodou této architektury je schopnost velmi přesné segmentace i v případech, kdy je k dispozici pouze omezené množství trénovacích dat. V oblasti špičkových modelů pro sémantickou segmentaci se zároveň jedná o poměrně malý model, který efektivně využívá dostupný výpočetní výkon.

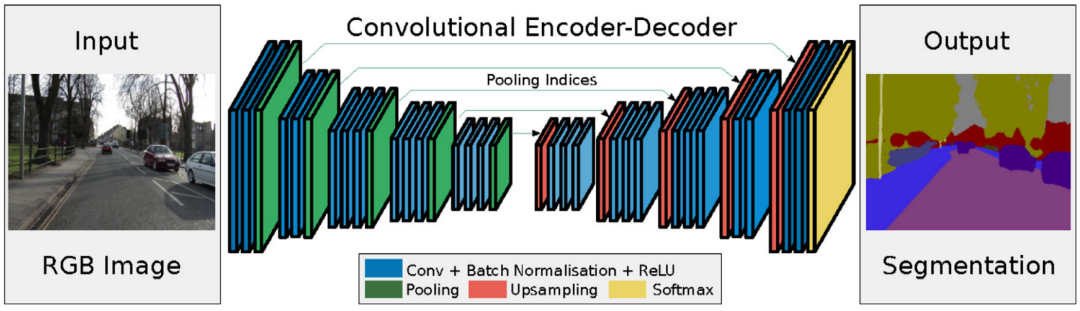
2.2.2 SegNet

Architektura Segnet byla publikována v roce 2016 a stejně jako U-Net se řadí do kategorie konvolučních neuronových sítí. Oba uvedené modely jsou si ve skutečnosti velmi podobné, což je zřetelné i při pohledu na schéma Segnet 2.5.

Podobně jako v předchozím případě, i zde je možné provést rozdělení modelu na dvě hlavní části, tedy enkodér a dekodér. Na straně enkodéru prochází vstupní snímek postupně pěti bloky, které se skládají z několika dílčích operací. Každý z bloků představuje konvoluci, normalizaci dávky, aktivační funkci ReLU a následný maxpooling. Opakování tohoto postupu vede opět k nárůstu abstrakce učitelných prvků a redukci původního rozlišení snímku.

Po přechodu dat na stranu dekodéru dochází postupně ke zpětnému nárůstu rozlišení. Na rozdíl od architektury U-net v tomto případě nedochází mezi stranou enkodéru a dekodéru přímo k přenosu dat, ale přenášena je pouze informace o tom, ze kterých pozic byly extrahovány hodnoty pomocí operace maxpooling [12]. Tento postup pomáhá kvalitně segmentovat jemné detaily snímku s ohledem na efektivní využití výpočetního výkonu [13].

Jako poslední část modelu je ve schématu zaznačena operace softmax, která slouží k finální úpravě výstupního tenzoru. Výstupem sítě je v případě podobných modelů tenzor, jehož rozměry ve dvou osách odpovídají rozlišení vstupního snímku. Velikost třetího rozměru tohoto výstupu může být různá a obecně odpovídá počtu tříd, které model rozlišuje. Operace softmax zajišťuje, že hodnoty výstupního tenzoru budou reprezentovat pravděpodobnost, se kterou každý z pixelů náleží jednotlivým třídám.



Obrázek 2.5: Architektura SegNet, zdroj: [12]

2.2.3 PSPNet

Tento model neuronové sítě poprvé představil Hengshuang Zhao a spol. v roce 2016 publikací článku [14]. Název této architektury je odvozen z anglického "pyramid scene parsing network". Hlavním účelem této práce bylo vytvoření segmentačního modelu, který by byl schopen pracovat nejen s globálním kontextem celého snímku, ale i s kontexty jeho jednotlivých částí.

Stejně jako předešlé modely, i tento je založen na principu konvoluční neuronové sítě. Na schématu 2.6 je zkratkou "CNN"označen právě konvoluční model, který slouží k vytvoření mapy prvků. Autor používá před-trénovaný model ResNet [14], obecně je ale možné i využití jiných modelů.

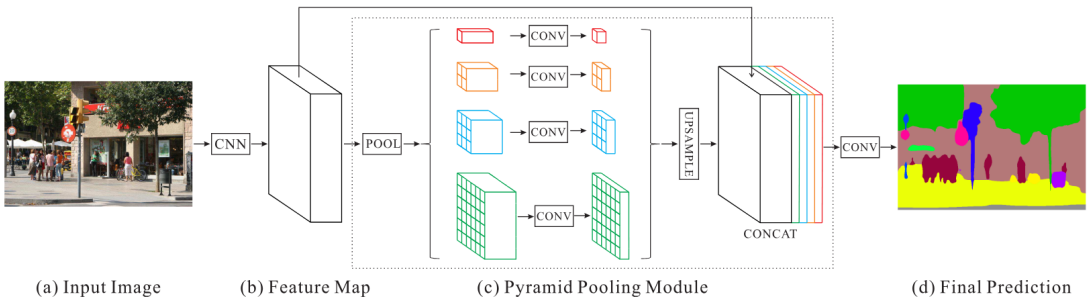
Extrahovaná mapa prvků potom vstupuje do modulu, který je pro tuto architekturu unikátní. Kolektiv autorů jej pojmenoval "pyramid pooling module", což by se dalo přeložit jako pyramidální modul pro sdružování. Základní princip spočívá v aplikaci operace pooling (sdružování) na různé části vstupních dat.

Jak je naznačeno na schématu, nejprve je operace pooling provedena pro celý zpracovaný snímek, poté je proces opakován pro snímek rozdělený do 4 oblastí, následně do 9 a dále. Takto upravená data jsou následně podrobena další konvoluci a následně nad-vzorkována (je zvýšeno jejich rozlišení). Nad-vzorkování probíhá pro každou verzi dat takovým způsobem, aby dosáhla shodného rozlišení jako mapa prvků před vstupem do tohoto modulu. Právě s touto původní mapou prvků jsou potom data ze všech verzí operace pooling složena a výsledný tenzor postupuje do další fáze zpracování [14].

Jakmile je složení veškerých dat provedeno, následuje poslední konvoluční vrstva této sítě, která generuje finální výstupní tenzor na základě požadovaného počtu tříd. Poslední provedenou operací je potom softmax, tento krok je nutný, abychom výstup převedli na tenzor pravděpodobností. Příkazem argmax, který je součástí knihovny numpy, je následně možné vygenerovat výslednou mapu tříd.

Výhodou tohoto přístupu je lepší vyhodnocení lokálního kontextu snímku za cenu pouze malého nárůstu požadavků na výpočetní výkon oproti samotné ResNet.

Pozn. Kolektiv autorů ve své práci testuje implementaci operace max-pooling i average-pooling, proto je ve schématu uvedeno pouze obecné označení pooling. Co se kvality segmentace týče, modely využívající druhou zmíněnou variantu dosahovaly nepatrně lepších výsledků v porovnání se svými protějšky [14].



Obrázek 2.6: Architektura PSPNet, zdroj: [14]

2.2.4 Další architektury

Pro sémantickou segmentaci obrazu je dále možné využít celou řadu dalších architektur, jako například ResNet, Deeplab, RefineNet, AlexNet, FCN a jiné. Tento výčet může posloužit jako základní přehled dnes známých metod, které jsou pro konkrétní aplikace často dále upravovány.

Rozsáhlejší přehled jednotlivých architektur je možné dohledat v literatuře [12], stejně jako rozsáhlou meta analýzu vědeckých prací, která sleduje moderní trendy v sémantické segmentaci pomocí neuronových sítí [15].

2.3 Datové sady

Datové sady jsou nedílnou součástí procesu, který zahrnuje vývoj, trénování a validaci jednotlivých modelů neuronových sítí. Jejich úkolem je poskytnout modelu dostatečné množství vstupních dat (originálních snímků) a požadovaných výstupů (anotovaných masek). Rozsah a kvalita trénovacích dat se řadí mezi významné faktory, které ovlivňují výslednou kvalitu segmentace.

V obecné rovině existuje několik rozdílných přístupů k segmentaci obrazových dat, může se jednat o detekci objektů, segmentaci instancí, pan-optickou segmentaci či sémantickou segmentaci. Jednotlivé datové sady jsou potom konkrétnímu přístupu přizpůsobeny. V souladu se zadáním této práce poskytuje tato kapitola základní přehled o datových sadách, které je možné využít v oblasti sémantické segmentace se zaměřením na snímky zemského povrchu.

Jednotlivé datové sady, nebo též datasety, pro sémantickou segmentaci povrchu země je možné rozdělit podle způsobu pořízení originálních snímků na satelitní, letecké a zachycené dronem. Zatímco u satelitních a leteckých snímků se zpravidla jedná o pohled kolmo na terén, u dronu se setkáváme s různými úhly natočení kamery. S tímto rozdělením potom souvisí další, podstatnější, parametr každého datasetu a tím je měřítko. V dané problematice se ovšem častěji používá pojem velikost pixelu, který udává reálné rozměry zemského povrchu zachyceného jedním pixelem.

Standardně se setkáme s daty, která odpovídají spektru viditelných barev. Ve výjimečných případech jsou dále u jednotlivých snímků dostupná data z termokamery, výška dronu nad povrchem nebo jeho poloha.

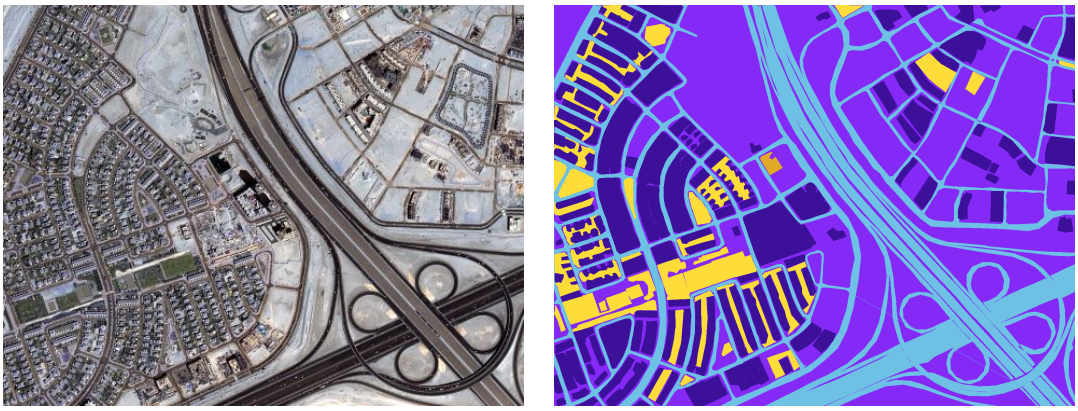
Obrazová data bývají uložena v nejrůznějších formátech, kterými jsou např. JPG, PNG, TIF, JSON nebo DICOM. Nepsaným pravidlem, kterým se velká část datasetů řídí, je použití formátu JPG pro originální snímky a PNG pro anotace.

2.3.1 Semantic segmentation of aerial imagery

Tato datová sada, zachycující město Dubaj, je hojně využívána pro výukové materiály a experimenty. Řada internetových návodů tento soubor dat používá pro demonstraci základních postupů pro trénování neuronových sítí.

Hlavním důvodem, který činí z této datové sady vhodného kandidáta na demonstraci postupů, je její malá velikost dosahující pouhých 32 MB. Tato skutečnost umožňuje svižné trénování některých sítí, a to i v případě, že je uživatel odkázán na výkon svého CPU bez možnosti využití výkonné grafické karty. Na straně druhé se zároveň jedná o zásadní nevýhodu, jelikož právě omezené množství dat následně limituje přesnost a adaptabilitu natrénovaného modelu.

Veškerá data jsou zde rozdělena do 8 složek, přičemž každá z nich obsahuje originální snímky, které tvoří jednu mapovou dlaždici. Rozměry jednotlivých dlaždic se výrazně liší, konzistentní je ovšem velikost pixelů, která je pro celý dataset 1 x 1 m. Originální snímky jsou uloženy ve formátu JPG. Kromě originálních snímků jsou samozřejmě k dispozici anotované masky, které dělí zachycený povrch do 6 následujících tříd: budova, země, cesta, vegetace, voda, neoznačeno. Datová sada je volně dostupná na webových stránkách kaggle.com [16]. Ukázku originálního snímku včetně odpovídající anotace nalezneme na obrázku 2.7.



(a) (b)

Obrázek 2.7: Ukázka (a) originálního snímku a (b) sémantické masky z datové sady Semantic segmentation of aerial imagery

2.3.2 Landcover.ai

Datová sada Landcover.ai zachycuje povrch několika různých oblastí Polska o celkové rozloze více než 200 km^2. Oproti předešlému zmíněnému zástupci se jedná o poněkud větší soubor dat o celkové velikosti přesahující 1,4 GB. Původní letecké snímky i vytvořené anotace jsou uloženy ve formátu TIF. Ačkoli větší část dat reprezentuje polskou přírodu, opomenuty nejsou ani části městské zástavby.

Pro datasety, které jsou vytvořeny na základě leteckých snímků, je typické, že reálná velikost pixelu je napříč všemi snímky konstantní. V tomto případě ovšem dostáváme k dispozici 33 snímků o rozlišení přibližně 9000 x 9500 pixelů, kde každý pixel představuje část povrchu o velikosti 25 x 25 cm. Dále je potom k dispozici dalších 8 dlaždic s rozlišením okolo 4200 x 4700 pixelů, pro které platí velikost pixelu 50 x 50 cm.

Anotované masky v tomto případě obsahují pouze 5 tříd, kterými jsou: cesta, les, voda, budova a pozadí. Ukázka datové sady Lancover.ai se nachází na obrázku 2.8. S ohledem na rozlišení a objem dat originálních souborů přikládám pouze jejich malé ústřižky.

Anotované masky jsou v TIF souborech uloženy jako matice, které obsahují celá čísla. Tato čísla symbolizují příslušnost pixelu k jednotlivým třídám a sama o sobě nemají význam konkrétní barvy. Pro lepší představu jsem pro tuto ukázku (viz obrázek 2.8) zvolil podobnou barevnou interpretaci, kterou využívá i předchozí dataset Semantic segmentation of aerial imagery.

Kompletní datová sada byla publikována ve vědecké práci [17], která mimo jiné uvádí i odkaz, ze kterého je možné dotyčná data získat.

(a) (b)

Obrázek 2.8: Ukázka (a) originálního snímku a (b) anotované masky z datové sady Landcover.ai

2.3.3 Massachusetts Roads Dataset

Jak je patrné ze samotného názvu, v této sekci prozkoumáme datovou sadu z prostředí amerického státu Massachusetts. Tento soubor dat byl představen v disertační práci [18] a zahrnuje letecké snímky z městského prostředí i venkovské krajiny.

Tento soubor dat svojí velikostí celkem 6 GB několikanásobně překonává zmíněný Landcover.ai a mapuje úctyhodných 2600 km^2 terénu. Původní snímky a k nim příslušné anotace zahrnuté v této datové sadě jsou rozděleny na dlaždice o velikosti 1500 x 1500 pixelů, při reálné velikosti pixelu 1 x 1 m potom každá dlaždice reprezentuje 2,25 km^2 povrchu Země.

Anotace jsou zde stejně jako v případě Landcover.ai uloženy v souborech s příponou .tif, zatímco přípona u originálních snímků je .tiff. Tato odlišnost v některých případech umožňuje využití jednoduššího algoritmu pro načítání dat.

(a) (b)

Obrázek 2.9: Ukázka (a) originálního snímku a (b) anotované masky z datové sady Massachusetts Roads Dataset

Název Massachusetts Roads Dataset je ve skutečnosti velmi výstižný, jelikož kromě lokace specifikuje velmi přesně i obsažená data. Jak je možné si povšimnout na obrázku 2.9, anotované masky obsahují pouze kategorie cesta a pozadí. Tato koncepce má potom určité důsledky pro implementaci konkrétního segmentačního modelu, které budou dále popsány v sekci 2.5.2. Kompletní datová sada je dostupná ke stažení na webových stránkách [18].

2.3.4 Semantic drone dataset

Po několika představených zástupcích z oblasti leteckých a satelitních snímků se dostáváme k datové sadě, která je založena na snímcích pořízených z dronu. Použitý dron prolétá městským prostředím v proměnné výšce asi 5 - 30 m a pomocí kamery zachycuje pohled kolmo dolů.

Značnou výhodou této datové sady je rozmanitost poskytovaných dat. Kromě barevných originálních snímků poskytuje například i snímky z termokamery. Na straně anotací je potom kromě sémantických masek možné naleznout i tzv. bounding boxes, které značí jednotlivé části snímku, na nichž se vyskytují osoby. Anotovaná data jsou k dispozici hned v několika formátech a Semantic drone dataset díky nim získává přesah i do oblasti detekce objektů.

Celková velikost této datové sady činí asi 4 GB, jednotlivé snímky terénu mají rozlišení 4000 x 6000 pixelů a jsou anotovány do celkem 20 tříd. Rozsah anotace je vskutku široký a kromě obvyklých tříd, kterými jsou voda, tráva, strom nebo zpevněná plocha, narazíme i na méně obvyklé kategorie jako například bazén, okno, dveře nebo pes.

Základní data pro sémantickou segmentaci jsou uložena ve formátech JPG na straně původních snímků a PNG v případě anotovaných masek. Autoři bohužel neposkytují informaci o rozsahu, ve kterém se pohybuje reálná velikost jednotlivých pixelů a z dostupných dat ji ani není možné přesně určit.

Doplňující informace k této datové sadě je možné dohledat na webových stránkách Technische Universität Graz [19], prostřednictvím kterých je možné také zažádat o její zpřístupnění. Datová sada je dále k dispozici na webu kaggle.com [16].

(a) (b) (c)

Obrázek 2.10: Ukázka (a) originálního snímku, (b) sémantické masky a (c) bounding boxes z datové sady Semantic drone dataset

2.3.5 Varied Drone Dataset for Semantic Segmentation

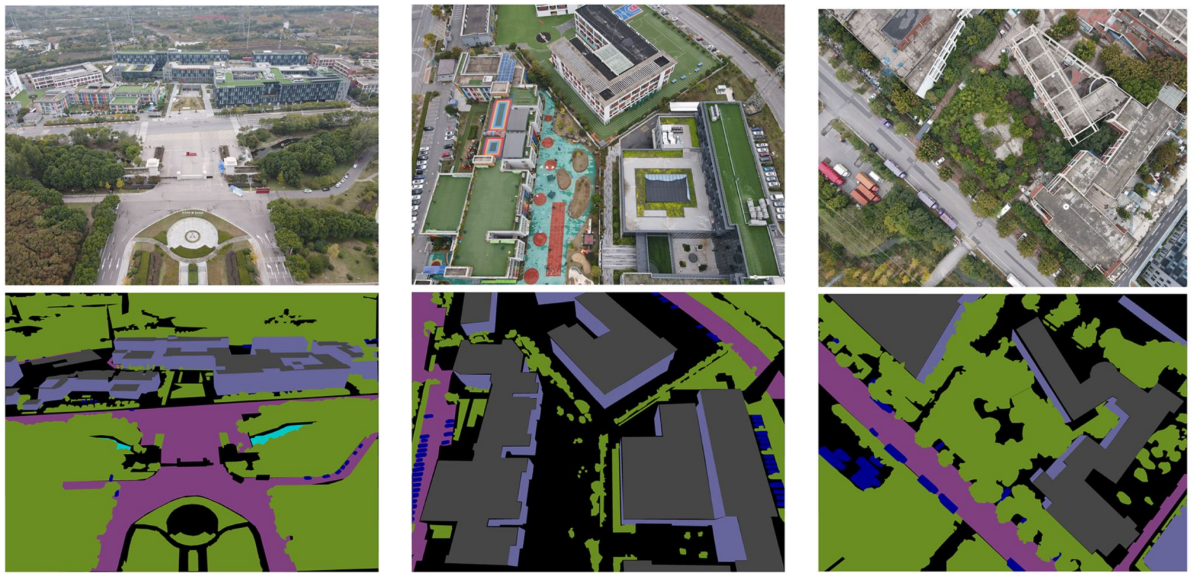
VDD neboli Varied Drone Dataset for Semantic Segmentation obsahuje celkem 400 snímků zemského povrchu a stejný počet odpovídajících anotovaných masek. Originální data byla zachycena pomocí dronu v oblasti východní Číny a obsahují scény z městské zástavby i venkovské krajiny. Pořízené snímky mají rozlišení 4000 x 3000 pixelů.

Jak napovídá název, tato datová sada klade důraz zejména na rozmanitost obsažených dat. Skupina autorů si stanovila jasný cíl - vytvořit takový dataset, na kterém by bylo možné natrénovat vysoce adaptabilní neuronovou síť s univerzálním použitím pro různé kombinace podmínek.

První zajímavostí je variabilní sklon kamery v průběhu pořizování jednotlivých snímků terénu. Úhel, který svírala osa kamery s povrchem Země, činil nejprve 30°, následně 60° a na závěr 90°. Dalším rysem zmíněné rozmanitosti je potom snaha o rovnoměrné rozložení dat mezi scény městské a přírodní povahy. Tato vlastnost je u datových sad poměrně běžná, nikoli však samozřejmá.

Důraz při sběru dat byl potom kladen i na různé světelné podmínky. Jak uvádí skupina autorů, snímky byly pořizovány v časovém rozmezí od léta 2022 do jara 2023 a to i v různých denních dobách kromě noci.

Veškeré původní snímky byly následně anotovány týmem odborníků pomocí specializovaného nástroje Labelme. Výsledné masky obsahují celkem 7 tříd segmentace, mezi které patří: vegetace, cesta, vozidlo, vodní plocha, střecha, zeď a ostatní. Kompletní informace ohledně této datové sady včetně její dostupnosti je možné dohledat v článku [20].



(a) (b) (c)

Obrázek 2.11: Datová sada Varied Drone Dataset for Semantic Segmentation: ukázka originálních snímků terénu včetně příslušných anotovaných masek pro sklon osy kamery vůči povrchu (a) 30°, (b) 60°, (c) 90°

2.4 Evaluační metriky

Nedílnou součástí každé výzkumné činnosti jsou jasně definovaná pravidla, která slouží k objektivnímu posouzení dosažených výsledků. Pro vyhodnocení kvality sémantické segmentace existuje několik evaluačních metrik, které číselně popisují správnost, se kterou neuronová síť generuje své predikce.

Správnost predikce je zpravidla určována porovnáním s odpovídající anotací. Anotované snímky ovšem neodpovídají realitě zcela přesně, a proto mohou být některé části predikce, které jsou správné, vyhodnoceny jako chybné. Tuto skutečnost je dobré mít na paměti s tím, že údaje o přesnosti nejsou absolutní a závisí na kvalitě poskytnutých dat.

Evaluačních metrik existuje celá řada a vhodnost jejich použití je vždy potřeba pečlivě posoudit podle konkrétní aplikace. V následujících odstavcích uvádím stručný popis základních evaluačních metrik, které odpovídají zaměření mé práce.

2.4.1 Accuracy

Accuracy neboli přesnost se řadí mezi nejjednodušší evaluační metriky a poskytuje základní přehled o kvalitě generovaných predikcí. Je definována jako podíl správně predikovaných pixelů ku jejich celkovému počtu. K výpočtu potom slouží rovnice 2.1.

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}} \tag{2.1}$$

Slabina tohoto přístupu se často projevuje při segmentaci snímků v případě, že míra zastoupení jednotlivých tříd je silně nevyvážená. Dobrým příkladem je úzká cesta, která vede pásem vegetace a z celkové rozlohy snímku zabírá jen velmi malou část. I kdyby taková cesta nebyla vůbec detekována a celý snímek by byl vyhodnocen jako jednolitý kus vegetace, přesnost může dosahovat hodnoty přes 90%, přestože daný výsledek je pro nás zcela nežádoucí [21].

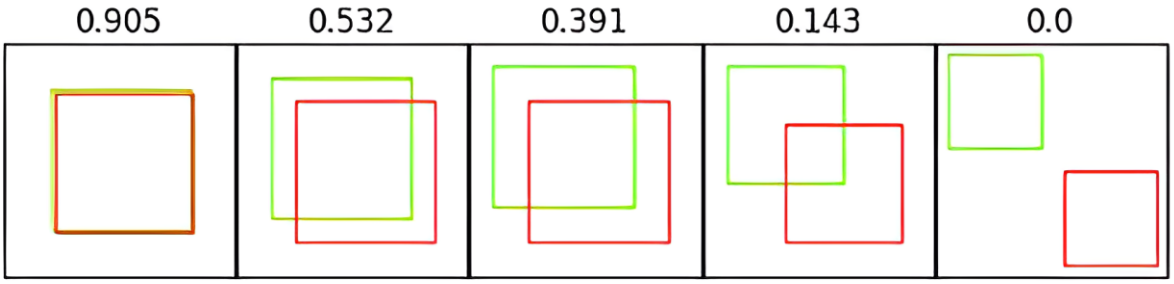
2.4.2 Mean IoU

Celým názvem Mean intersection over union neboli Střední hodnota průniku ku sjednocení je univerzálním hodnotícím kritériem pro sémantickou segmentaci do více tříd. V literatuře se často vyskytuje také pod názvem Jaccard coefficient nebo Jaccard index.

Tato evaluační metrika je založena na hodnocení každé třídy zvlášť. Pro predikci i pro anotaci jsou zjištěny oblasti, které náleží dané třídě. V dalším kroku je vypočtena velikost průniku a sjednocení těchto oblastí. Pro každou třídu je potom vypočten podíl průniku a sjednocení (IoU), přičemž aritmetický průměr těchto získaných hodnot je potom žádaným výsledkem [21]. Výpočet je matematicky popsán rovnicí 2.2, obrázek 2.12 ukazuje několik příkladů hodnoty IoU pro různou míru překrytí dvou oblastí.

$$IoU(A, P) = \frac{|A \cap P|}{|A| + |P| - |A \cap P|} \tag{2.2}$$

Pokud bychom tuto metriku aplikovali na problematický případ zmíněný v 2.4.1, dosažená hodnota Mean IoU by byla výrazně nižší než Accuracy a dávala by jasný signál o nesprávné segmentaci. Právě kombinace Accuracy a Mean IoU je běžnou volbou pro sledování správnosti, se kterou používaný model pracuje.



Obrázek 2.12: Hodnoty IoU při různém překryvu dvou oblastí, zdroj: [22]

2.4.3 Ostatní evaluační metriky

Kromě dvou již zmíněných metrik se v praxi využívá i řada dalších, kterými jsou například Precision, Recall nebo F1 score. Shrnutí jednotlivých postupů včetně způsobu výpočtu je k dohledání v literatuře [21].

2.5 Před-trénované modely

Jak bylo již zmíněno v kapitole 2.2, často je možné využít již natrénovaný model a tím se vyhnout výpočetně náročnému procesu trénování.

Pro potřeby mojí práce je bohužel k dispozici minimální množství volně dostupných před-trénovaných modelů. Z tohoto důvodu jsou si architektury obou vybraných modelů velmi podobné.

2.5.1 Model A

První nalezený model, který odpovídá mému zaměření, byl publikován na webu Towards Data Science [23], který podává základní informace o modelu a jeho trénování. Veškeré doplňující podklady jsou dostupné v github repositáři autora [24]. Jedná se o model založený na architektuře U-Net, která je doplněna o některé modifikace. Trénování proběhlo na datové sadě Semantic segmentation of aerial imagery.

První změnou oproti originálu jsou dvě konvoluční operace místo jedné v každém stupni enkodéru i dekodéru. Mezi každou dvojicí těchto operací je vždy zařazena funkce Dropout, která se používá zejména při trénovaní na menších datových sadách jako prevence přetrénování.

Další změnou je potom proces Rescaling, který normalizuje hodnoty vstupního tenzoru na jednotkový rozsah. Jako vstup je tedy v tomto případě možné použít i obrazová data s hodnotami typu uint, která jsou následně konvertována na typ float.

Model je navržen tak, aby segmentoval vstupní RGB snímky o velikosti 160 x 160 pixelů do 6 tříd. Důležité je poznamenat, že autor pro načítání dat používá knihovnu opencv, která pracuje s pořadím kanálů BGR. Snímky jsou tedy načteny ve formátu BGR, poté dochází k jejich konverzi na RGB a teprve v následujícím kroku vstupují do modelu. Při využití stejné knihovny je tedy nutné pro správnou funkčnost modelu dodržet celý postup včetně konverze.

2.5.2 Model B

Tento model je opět založen na populární architektuře U-Net, která je autorem doplněna o řadu dalších operací.

Stejně jako v případě Model A, i zde probíhá normalizace dat ihned po vstupu do modelu a veškeré hodnoty vstupního tenzoru jsou převedeny na jednotkový rozsah. Dalším společným rysem jsou i 2 konvoluční operace v každém stupni enkodéru i dekodéru včetně již zmíněné operace dropout.

V tomto případě je ovšem do struktury modelu zařazena za každou konvolucí ještě operace batch normalization, která umožňuje například rychlejší trénování nebo obecnější vnímání kontextu [25].

Konkrétní konfigurace tohoto modelu je uzpůsobena pro klasifikaci barevných snímků o velikosti 256 x 256 pixelů. Velikost hrany 256 pixelů je z praktického hlediska rozumnější než 160 pixelů v případě Model A. Tento rozměr je každou konvolucí typicky zmenšen na polovinu. Takových operací je v enkodéru provedena celá řada a v dekodéru potom dochází ke zpětnému zvětšení těchto rozměrů. Rozměry vstupního snímku je tedy výhodné volit vždy obecně 2^n, aby nedocházelo k jejich deformaci. Tento problém se samozřejmě netýká počtu kanálů.

Dalším specifikem tohoto modelu je trénování v barevném formátu BGR. Autor používá pro načítání originálních snímků oblíbenou knihovnu opencv, která čte jednotlivé barvy právě v pořadí BGR, nikoli RGB. Součástí následného postupu není konverze dat do RGB formátu a do modelu tudíž vstupují v pořadí kanálů BGR. Tuto skutečnost je dobré mít na paměti, jelikož správné pořadí jednotlivých barevných kanálů je zásadní pro funkčnost modelu.

Trénování proběhlo na datové sadě Massachusetts Roads Dataset tak, aby byl model schopen segmentovat vstupní snímky do dvou tříd: cesta a pozadí. Jelikož se jedná o úlohu binární klasifikace, výstupem není tenzor, jehož hloubka odpovídá počtu tříd, ale pouze matice s hodnotami od 0 do 1. Čím je daná hodnota blíže číslu 1, s tím větší jistotou model predikuje, že dotyčný pixel je ve skutečnosti součástí cesty. Pomocí vhodně zvoleného prahu je potom škála hodnot převedena na binární masku, která je konečným požadovaným výstupem.

Popsaný model včetně doplňující dokumentace je k dispozici na webu github.com v repositáři autora [26].

2.6 Plánování trasy

Dosavadní část rešerše se věnovala sémantické segmentaci snímků zemského povrchu se zaměřením na identifikaci pozemních komunikací. Aby bylo možné na základě provedené segmentace naplánovat trasu terénem pro mobilního robota, je nutné zvolit pro tento úkol správný algoritmus.

Dostupných algoritmů pro plánování trasy existuje celá řada, přičemž mezi nejznámější patří zástupci jako BFS, Dijkstra's, A* nebo HPA*. Výběr konkretního algoritmu pro moji aplikaci jsem omezil pouze na ty, které zaručují nalezení nejkratší cesty. Dalším významným parametrem každého přístupu je také efektivita, se kterou využívá výpočetní výkon.

Na základě dostupných informací jsem zvolil algoritmus A*, který podle analýzy [27] dosahuje výborné úrovně efektivity a zároveň je velice jednoduchý na implementaci.

2.6.1 A* algoritmus

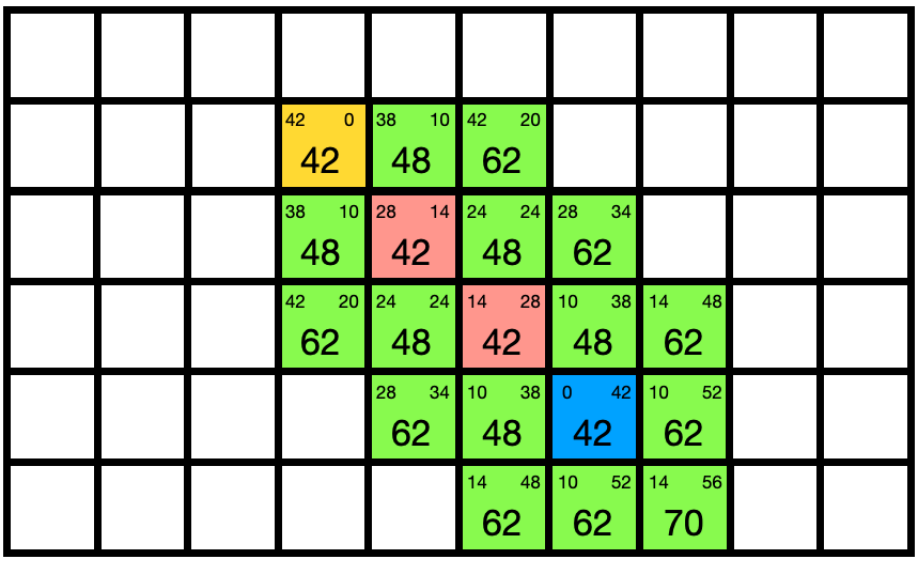
Tento algoritmus pro plánování trasy má za úkol dostat se z počátečního uzlového bodu do bodu konečného za využití minimálních nákladů. Minimální náklady mají v tomto kontextu stejný význam jako nejkratší uražená vzdálenost.

Celý proces začíná v prvním zadaném bodě. V tomto bodě se algoritmus podívá na všechny sousední uzlové body a vyhodnotí jejich vhodnost. Aktuální pozice je potom nastavena na uzel, který je vyhodnocen jako nejvhodnější. Tento proces se opakuje, dokud není dosaženo cílového bodu.

Vhodnost jednotlivých bodů je posuzována podle celkových nákladů na dosažení cíle, tuto hodnotu označujeme názvem F cost a jedná se o součet dílčích nákladů, které označujeme jako G cost a H cost. G cost je definována jako přesná hodnota nákladů, které jsou zapotřebí pro dosažení dané pozice z počátečního bodu. H cost potom představuje odhad nákladů na cestu z posuzovaného bodu do cíle a zpravidla je vypočtena jako vzdálenost do cíle "vzdušnou čarou", tedy jako kdyby v cestě nestály žádné překážky. Uzlový bod s minimální hodnotou F cost je potom v každém kroku zvolen jako směr posunu.

Obrázek 2.13 znázorňuje postup algoritmu na jednoduchém příkladu. Podrobný popis algoritmu A* a několika dalších je možné dohledat v literatuře [27].

Velmi snadnou implementaci tohoto algoritmu v programovacím jazyku python umožňuje knihovna pathfinding.



Obrázek 2.13: Jednoduchá ukázka algoritmu A*. Hodnota vlevo nahoře v každém bodě znázorňuje G cost, zatímco H cost je na straně pravé. Celkové náklady F cost jsou potom dole uprostřed. Zdroj: [28]

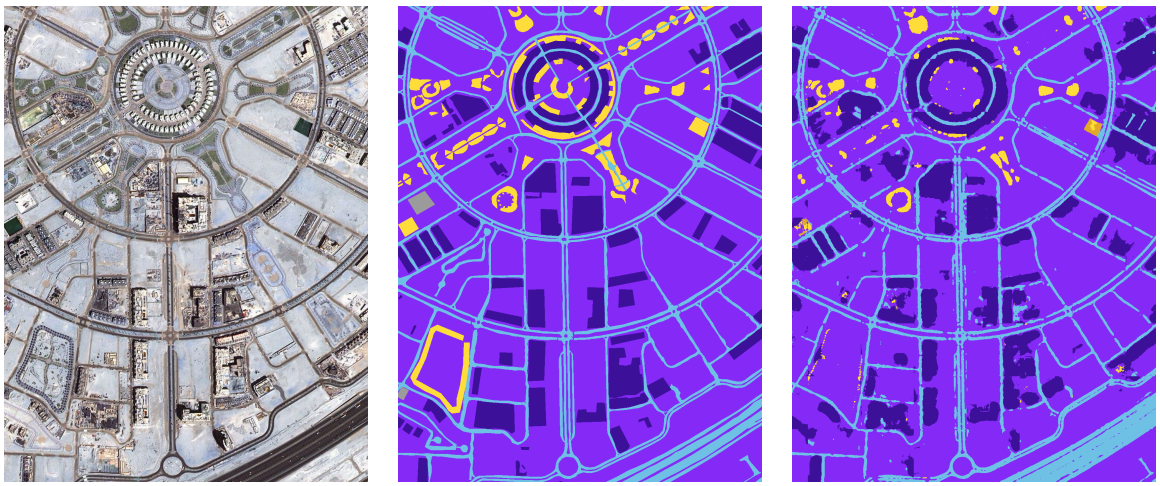
3 Postup a výsledky řešení

3.1 Testování dostupných modelů

3.1.1 Model A

První sada testování proběhla na před-trénovaném modelu, jehož architektura je popsána v kapitole Rešerše v sekci Model A.

Funkčnost modelu jsem otestoval nejprve na datové sadě Semantic segmentation of aerial imagery, která posloužila k jeho trénování. Výsledek testování ilustruje obrázek 3.1 a na první pohled odpovídá výsledkům uvedeným v článku [23]. Při srovnání obrázků (b) a (c) je zřetelné, že predikce sice poskytuje dobrý přehled o výskytu jednotlivých tříd na originálním snímku (a), ale obsahuje celou řadu nedokonalostí. Při pečlivém pohledu je zřejmé, že dochází například k častému přehlížení vegetace nebo nesouvislé segmentaci některých cest.



(a) (b) (c)

Obrázek 3.1: Testování Modelu A na datové sadě Semantic segmentation of aerial imagery: (a) originální snímek, (b) anotovaná maska, (c) vygenerovaná predikce

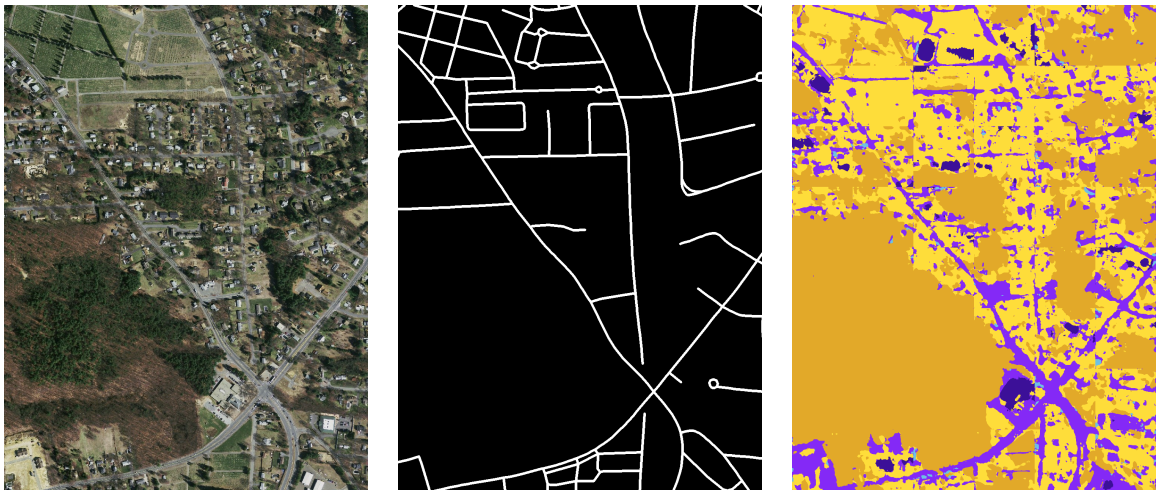
Kromě grafické ukázky uvádí autor také číselné hodnoty evaluačních metrik, kterých dosáhl během trénování tohoto modelu. Pro účely vyhodnocení byla datová sada rozdělena do dvou částí: 90% dat bylo uloženo do trénovací množiny, zatímco 10% bylo vyčleněno pro validaci výsledků. Po skončení každé epochy trénování proběhlo vyhodnocení modelu právě na množině validačních dat. S tímto přístupem autor uvádí konečné dosažené hodnoty Accuracy = 0,8616 a Jaccard index = 0,6599. Jelikož byla datová sada do trénovací a validační množiny rozdělena náhodně, nebylo možné tento proces přesně replikovat. Vyhodnocení kvality segmentace jsem tedy provedl na celé datové sadě s dosaženými výsledky Accuracy = 0,8551 a Jaccard index = 0,5154. Zatímco vyhodnocení

Accuracy dopadlo velmi podobně, Jaccard index se znatelně liší. Značný vliv zde může mít množství testovaných dat, jelikož validační množina obsahující pouhých 10% snímků nemusí být dostatečně kvalitním reprezentativním vzorkem, zejména v případě takto malé datové sady.

Dalším krokem potom bylo testování stejného modelu na datové sadě Massachusetts Roads Dataset. Hlavním cílem bylo zjistit, zda bude tento model, natrénovaný na pouštních podmínkách, schopen kvalitní segmentace ve výrazně odlišném prostředí.

Při prvním pohledu na obrázek 3.2 se zdá, že vygenerovaná predikce (c) zachycuje přinejmenším jakousi základní strukturu odpovídající originálnímu snímku (a). V tento okamžik je ovšem nutné poznamenat, že veškeré oblasti, které jsou vyznačeny tmavší žlutou barvou, jsou modelem vyhodnoceny jako vodní plocha. Stejně tak by se mohlo zdát, že světle fialová barva znázorňuje cesty, ta ale ve skutečnosti reprezentuje třídu země (land v originálním znění), zatímco pro cesty je vyhrazena barva světle modrá. Světle žlutá barva má v podaní tohoto segmentačního modelu význam vegetace a tmavě modrá potom zobrazuje budovy.

Z výše uvedených informací je tedy jasné, že výsledek na obrázku 3.2 je zcela neuspokojivý a správně segmentovány jsou pouze některé části vegetace. Tento závěr ovšem není nikterak překvapivý, jelikož podmínky datové sady pro trénování jsou velmi odlišné od testovaného snímku.



(a) (b) (c)

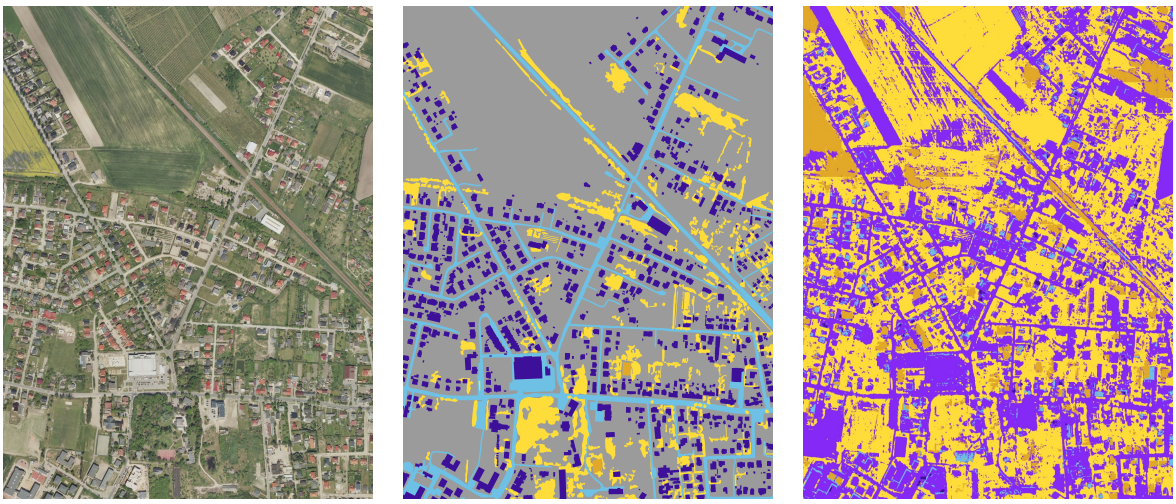
Obrázek 3.2: Testování Modelu A na datové sadě Massachusetts Roads Dataset: (a) originální snímek, (b) anotovaná maska, (c) vygenerovaná predikce

V posledním kroku jsem Model A otestoval na datové sadě Landcover.ai, která odpovídá našim geografickým a klimatickým podmínkám. Barevná interpretace anotovaných masek je uzpůsobena barvám, se kterými pracuje použitý model.

Originální snímek včetně příslušné anotace je zobrazen na obrázku 3.3 spolu s predikcí a podobně jako v případě testování na Massachusetts Roads Dataset se struktura predikce do určité míry podobá skutečnému stavu. Rozdělení pixelů do jednotlivých tříd ovšem realitě zdaleka neodpovídá.

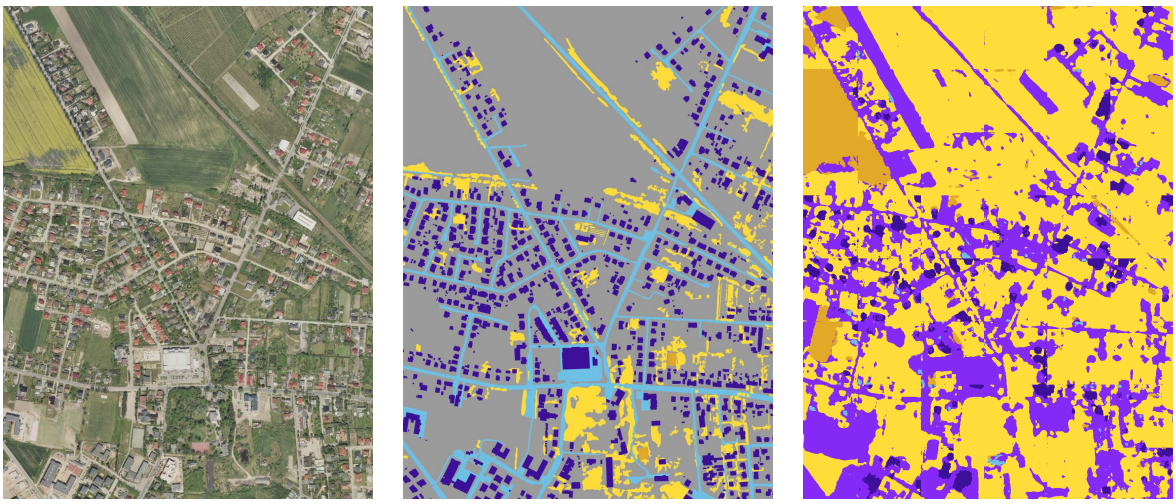
Špatná kvalita segmentace je v tomto případě opět odůvodnitelná rozdílným prostředím trénovací a testovací datové sady. Dalším faktorem, který mohl výsledky negativně ovlivnit je potom rozdílné měřítko. Reálná velikost pixelů v testovacím snímku je totiž

25 x 25 cm oproti 1 x 1 m na straně snímků použitých pro trénování. Na obrázku 3.4 je tedy možné vidět výsledek testování s upraveným měřítkem. Tento experiment ovšem nevedl k výraznému zlepšení a nezbývá tedy než konstatovat, že před-trénovaný Model A není pro tyto podmínky použitelný.



(a) (b) (c)

Obrázek 3.3: Testování Modelu A na datové sadě Landcover.ai s původním měřítkem: (a) originální snímek, (b) anotovaná maska, (c) vygenerovaná predikce



(a) (b) (c)

Obrázek 3.4: Testování Modelu A na datové sadě Landcover.ai s upraveným měřítkem: (a) originální snímek, (b) anotovaná maska, (c) vygenerovaná predikce

Aby bylo možné porovnat oba testované modely z hlediska náročnosti na výpočetní výkon, bude v obou případech změřen čas potřebný pro zpracování referenční fotografie. Pro tyto účely byl zvolen snímek z datové sady Massachusetts Roads Dataset o rozlišení 1500 x 1500 pixelů. V případě tohoto modelu byly provedeny celkem 3 experimenty pro eliminaci náhodné chyby. Průměrný čas segmentace dosáhl nakonec hodnoty 4,3 sekundy.

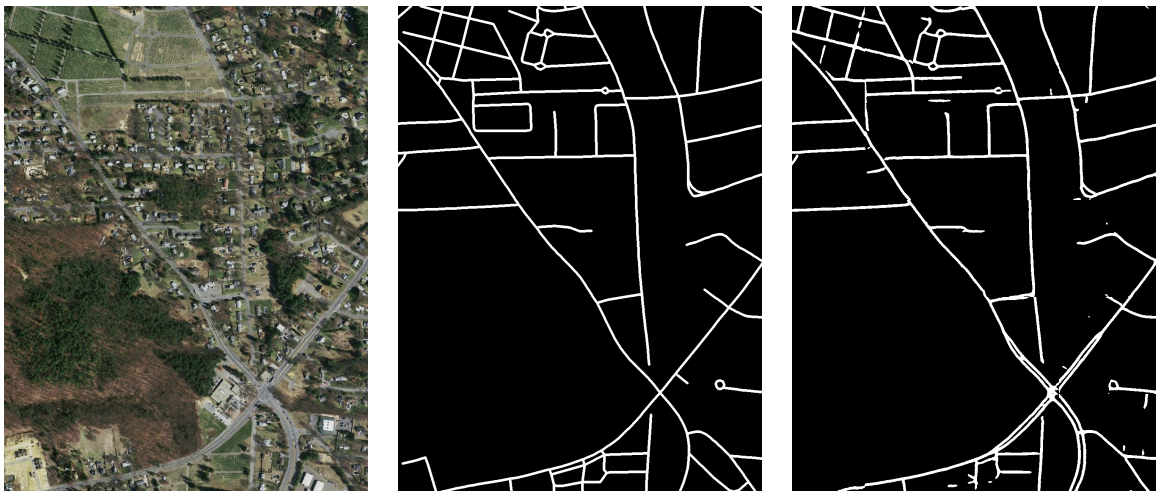
Dříve než započalo měření času, bylo provedeno nutné zpracování vstupního snímku včetně oříznutí na 1440 x 1440 pixelů, aby mohlo dojít k následnému dělení na dílčí záplaty o velikosti 160 x 160 pixelů. Jelikož výsledná predikce je zpětně složena z těchto záplat, její rozlišení odpovídá snímku oříznutému, nikoli původnímu.

Měřený čas potom zahrnuje vytvoření predikce pro každou ze záplat a jejich zpětné složení. Experiment byl proveden na CPU značky Intel, konkrétně na modelu i7-12700H.

3.1.2 Model B

Stejně jako v předešlém případě, i tento model je založen na populární architektuře U-Net a je blíže popsán v samostatné sekci Model B. Pro trénování této neuronové sítě zvolil autor datovou sadu Massachusetts Roads Dataset, na které také proběhlo první kolo testování.

Výsledek experimentu je možné vidět na obrázku 3.5 a na první pohled je zřetelná velmi dobrá kvalita segmentace. V některých místech vykazuje predikce určité nedokonalosti, ale najdou se i scénáře, kdy vystihuje realitu lépe než samotná anotovaná maska. Zejména při pohledu na víceproudové silnice je možné si povšimnout, že segmentační model správně rozlišuje dvě souběžné části komunikace a prostor mezi nimi. Na straně anotovaných masek se autor datové sady dopouští jistého zjednodušení a tento scénář zachycuje stejným způsobem jako jednoduchou pozemní komunikaci.



(a) (b) (c)

Obrázek 3.5: Testování Modelu B na datové sadě Massachusetts Roads Dataset: (a) originální snímek, (b) anotovaná maska, (c) vygenerovaná predikce

Velkou výhodou oproti Modelu A je souvislost segmentovaných cest. Výstupní snímky jsou v tomto případě dostatečně kvalitní na to, aby bylo možné je použít pro plánování trasy. Takové úrovně se v případě Modelu A nepodařilo dosáhnout na žádném z testovaných datasetů.

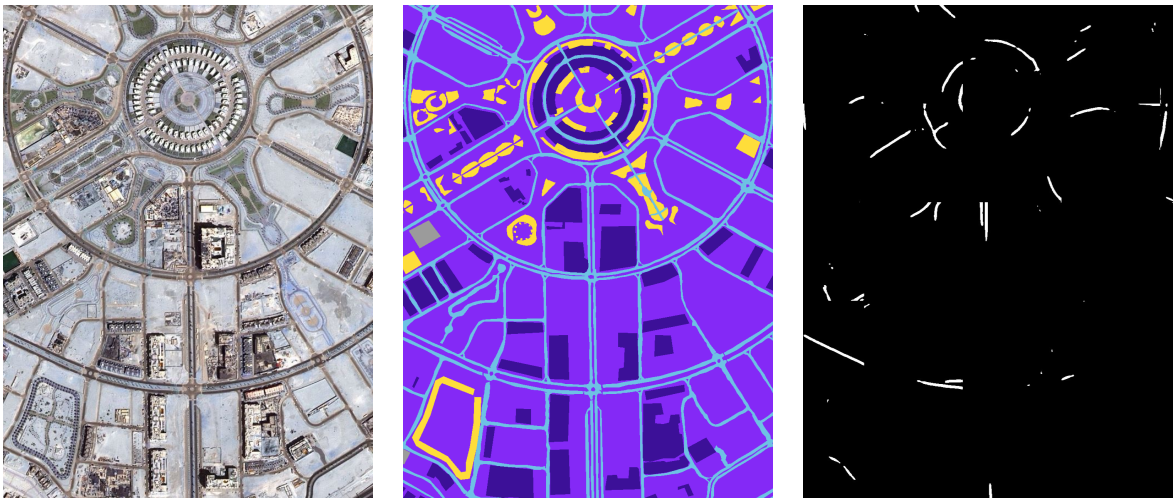
Při výpočtu evaluačních metrik se potom dostáváme do poněkud paradoxní situace, jelikož oblasti predikce, které vystihují realitu lépe než tytéž části anotované masky, budou vyhodnoceny jako chybné. Jak bylo již uvedeno v sekci Evaluační metriky, je dobré mít na paměti, že vypočtené hodnoty jsou závislé nejen na kvalitě výstupní predikce, ale taky na kvalitě anotací.

Experiment byl proveden pro celkem 14 testovacích snímků, které jsou součástí této datové sady, přičemž bylo dosaženo hodnot Accuracy = 0,9836 a Jaccard index = 0,8024, které potvrzují vysokou kvalitu segmentace.

Autor dává k dispozici ukázku několika segmentovaných snímků, které svou kvalitou odpovídají experimentálně vygenerovaným predikcím. Výsledné hodnoty evaluačních metrik, které byly dosaženy během trénování, ovšem dostupné nejsou a není tedy možné je porovnat.

Model B byl dále otestován na datové sadě Semantic segmentation of aerial imagery, která reprezentuje poněkud odlišné klimatické podmínky, než na kterých byl tento model původně natrénován.

Jak je možné vidět na obrázku 3.6, některé části cest byly segmentovány správně, ovšem jedná se pouze o zlomek skutečné dopravní sítě, která je na původním snímku zachycena. Výstupní segmentované snímky v tomto případě nejsou ani zdaleka použitelné pro plánování trasy, což ovšem není překvapivé vzhledem k rozdílné povaze trénovací a testovací datové sady.



(a) (b) (c)

Obrázek 3.6: Testování Modelu B na datové sadě Semantic segmentation of aerial imagery: (a) originální snímek, (b) anotovaná maska, (c) vygenerovaná predikce

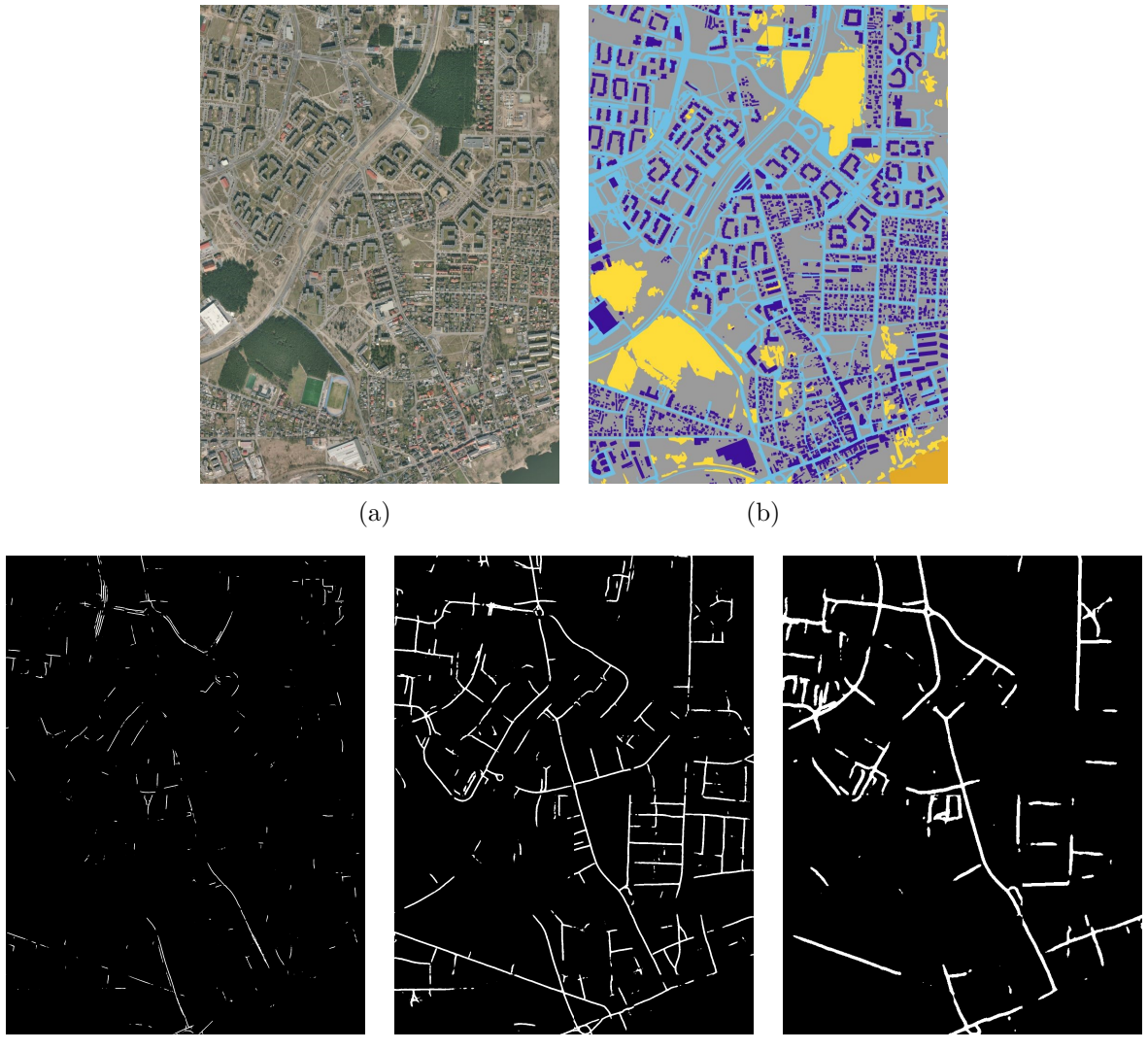
Pro Model B jsem nakonec jako vstupní data použil snímky z datové sady Landcover.ai, abych otestoval jeho funkčnost v podmínkách, které jsou nám blízké. Pro jednoduchou orientaci je barevná interpretace anotovaných masek ponechána stejná jako v případě testování Modelu A.

Experiment byl tentokrát proveden se snímkem, u kterého je velikost jednotlivých pixelů 50 x 50 cm. Testovaný model byl ovšem natrénován na datově sadě, která je složena ze snímků o velikosti pixelu 1 x 1 m. Aby bylo možné posoudit vliv měřítka na kvalitu segmentace, testování bylo provedeno pro 3 verze vstupních dat. V prvním případě byl vstupní snímek ponechán zcela beze změny, dále byla velikost pixelu upravena na 1 x 1 m a v posledním kroku došlo k dalšímu zvětšení pixelů na celkovou velikost 2 x 2 m. Výsledky experimentu je možné vidět na obrázku 3.7.

Nejlepší kvalita segmentace byla podle očekávání dosažena pro shodné měřítko testovací i trénovací datové sady. Příslušná predikce je v obrázku umístěna na pozici (d).

V obou případech, kdy byla velikost pixelů menší (c) i větší (e), potom dochází k určitému poklesu přesnosti. Důvod tohoto poklesu vychází ze skutečnosti, že neuronová síť vyhodnocuje jednotlivé pixely nejen na základě jejich barvy, ale bere v potaz i strukturu okolního prostředí a celkový kontext snímku.

V tomto okamžiku je vhodné podotknout, že umělá změna měřítka byla provedena jednoduše sloučením čtyr pixelů do jednoho. Velikost zemského povrchu, která je zachycena originálním snímkem, tedy zůstává stále stejná, ovšem rozlišení snímku se mění. V tomto případě je původní rozlišení snímku 4096 x 4096 pixelů, poté dochází ke snížení na 2024 x 2024 pixelů a v dalším kroku až na 1024 x 1024 pixelů. Zde je důležité si uvědomit, že snímek je před vstupem do modelu rozdělen na záplaty o velikosti 256 x 256 pixelů. V každém případě tedy do modelu vstupuje dlaždice se stejným rozlišením ale jiným měřítkem, což znamená jinou reálnou velikostí pixelů. Z této skutečnosti vyplývá, že neuronová síť dostává pokaždé jinak velkou část zemského povrchu a tedy i rozdílné množství kontextu.



(a) (b)

(c) (d) (e)

Obrázek 3.7: Testování Modelu B na datové sadě Landcover.ai: (a) originální snímek, (b) anotovaná maska, dále predikce pro různou velikost pixelu: (c) 50 x 50 cm, (d) 1 x 1 m, (e) 2 x 2 m

Na této datové sadě se bohužel nepodařilo dosáhnout stejné kvality, jako v případě Massachusetts Roads Dataset. Ve všech testovaných případech je jen sotva možné využít segmentované snímky za účelem plánování trasy pro mobilního robota. V tomto případě by ovšem bylo možné provést ladění před-trénovaného modelu, které bylo stručně popsáno v kapitole Neuronové sítě.

I pro tento model bylo provedeno měření času segmentace za stejných podmínek jako v předchozím případě. Oproti neuronové síti Model A bylo dosaženo znatelné časové úspory, jelikož průměrná doba segmentace referenčního snímku činila pouze 1,8 sekundy. Rozlišení predikce je v tomto případě 1280 x 1280 pixelů, jelikož vstupní vrstva neuronové sítě vyžaduje dělení na záplaty o velikosti 256 x 256 pixelů.

3.1.3 Shrnutí výsledků testování

V následujících odstavcích přikládám stručné shrnutí výsledků, kterých bylo dosaženo během testování. Získané poznatky poslouží k přehlednějšímu srovnání modelů a vyvození důsledků pro další část práce.

Model A se ukázal jako zcela nevhodný pro segmentaci snímků z prostředí datových sad Landcover.ai a Massachusetts Roads Dataset, jelikož vygenerované predikce ani zdaleka neodpovídají reálnému členění terénu. V případě datasetu Semantic segmentation of aerial imagery bylo dosaženo výrazně lepší kvality segmentace. Ve výsledných predikcích se ovšem vyskytovalo značné množství nedokonalostí, a proto ani tyto snímky není možné použít pro plánování trasy.

Model B dosáhl velmi dobrých výsledků zejména na své původní datové sadě Massachusetts Roads Dataset. Spolehlivost generovaných predikcí je v tomto případě dostatečná na to, aby bylo možné je použít pro plánování trasy původním terénem. Experimenty na zbývajících dvou datových sadách takové úrovně bohužel nedosáhly, ale při testování na Landcover.ai se podařilo jasně prokázat vliv měřítka na kvalitu segmentace. Kromě obrázků, které jsou uvedeny v této práci, je možné další řadu příkladů dohledat ve vytvořeném repositáři na webu github.com [29].

Z hlediska výpočetního výkonu se na referenčním snímku ukázal jako efektivnější Model B. Uvedené neuronové sítě bohužel nebylo možné srovnat na snímcích z různých ročních období, jelikož pro tyto účely nebyla nalezena vhodná datová sada.

3.2 Tvorba vlastní datové sady

Hlavním cílem mé práce je segmentovat letecké snímky zemského povrchu takovým způsobem, aby bylo možné výsledným snímkem naplánovat trasu pro mobilního robota. Pro tento účel by bylo vhodné, aby segmentovaný snímek obsahoval alespoň základní informace o nalezených cestách - tedy jestli jsou zpevněné či nezpevněné. Volně k dispozici bohužel není žádný před-trénovaný model, který by byl schopen tímto způsobem cesty rozlišovat, a neexistuje ani vhodná datová sada, která by umožnila takový model natrénovat. V této kapitole čtenář nalezne kompletní postup, který vedl k vytvoření vlastní datové sady, včetně stručného popisu výsledků a názorné ukázky.

Jako základ pro moji datovou sadu posloužily letecké snímky České republiky, které na svých webových stránkách poskytuje Český úřad zeměměřický a katastrální. V aplikaci Geoprohlížeč, která je dostupná na adrese [30], je možné stahovat velkoformátové mapové dlaždice o rozlišení 16000 x 20000 pixelů, přičemž jeden pixel představuje 12,5 x 12,5 cm zemského povrchu. Každá dostupná dlaždice tedy pokrývá plochu

2 x 2,5 km a ke stažení je k dispozici ve formátu JPG. Jednotlivé dlaždice jsem vybíral tak, aby byly pro účely anotace zachyceny zejména lokality s rozmanitým členěním terénu. Tyto lokality byly na snímku zachovány a zbylá část oříznuta. Získané části snímku byly poté rozděleny na dílčí záplaty o velikosti 1024 x 1024 pixelů.

Dalším krokem zvoleného postupu byla samotná anotace jednotlivých fragmentů původního snímku. Jako anotační nástroj jsem zvolil produkt Darwin od společnosti V7 labs a to zejména díky funkci poloautomatické segmentace. Tato funkce je založena na modelu neuronové sítě, který pomáhá označovat oblasti pixelů s podobnými vlastnostmi. V praxi je výhodné tento model využít jako asistenta pro anotaci budov nebo homogenních pásů vegetace, ovšem například pro polní cesty je často efektivnější anotace manuální. Každý snímek jsem rozdělil do celkem sedmi kategorií, kterými jsou: vegetace, zpevněná cesta, nezpevněná cesta, budova, vodní plocha, pole bez vegetace, neoznačeno/ostatní. Ukázku, jak vypadá snímek v průběhu anotace je možné vidět obrázku 3.8. Zelenou barvou je zde anotována vegetace, červenou barvou zpevněná cesta a žlutou cesta nezpevněná.

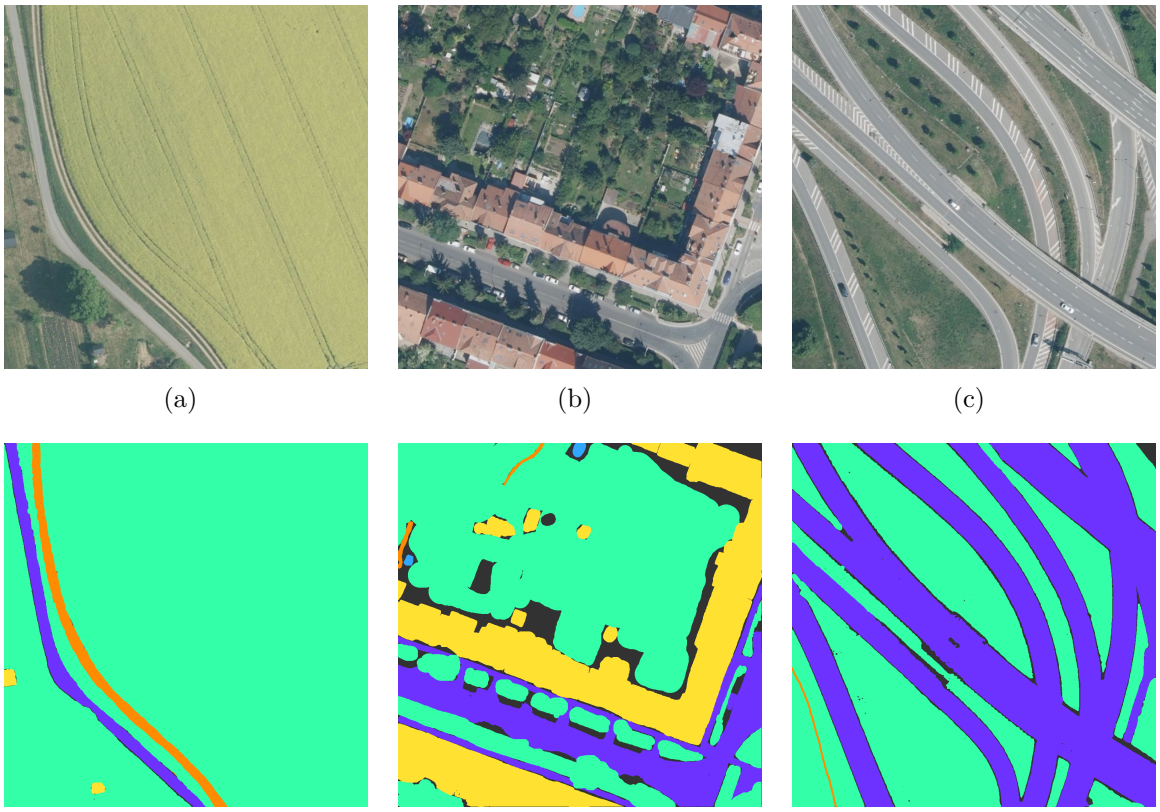
Obrázek 3.8: Ukázka snímku během procesu anotace, publikováno se souhlasem V7 labs

Struktura výsledného souboru je podobná jako v případě Semantic segmentation of aerial imagery a základní dělení zahrnuje 23 složek, přičemž každá z nich slouží jako úložiště dat z jedné lokace. Každá z těchto lokací se dále dělí na dvě složky, přičemž jedna obsahuje části originálního snímku ve formátu JPG a ve druhé jsou jako PNG soubory uloženy příslušné anotované masky. Celkem bylo anotováno 224 záplat o velikosti 1024 x 1024 pixelů.

Celý postup byl inspirován datovou sadou Semantic segmentation of aerial imagery, která ukazuje, že i se svou velikostí pouhých 32 MB může posloužit k trénování funkčního modelu neuronové sítě. Moje datová sada s celkovým objemem dat 62 MB tohoto zástupce překonává téměř dvojnásobně, stále se ovšem řadí mezi velmi malé datasety a v kontextu trénování neuronových sítí poskytuje velmi omezené množství informací. Několik ukázek

originálního snímku a odpovídající anotace je možné vidět na obrázku 3.9. Celá datová sada je potom k dispozici v github repositáři [29].

Na obrázku 3.9, zejména potom v sekci (e) je možné povšimnout si typických nedokonalostí anotovaných masek. Vyobrazené prostředí obsahuje velké množství drobných prvků, které často nelze jednoznačně přiřadit k žádné z používaných tříd nebo z časových důvodů není možné se věnovat každému z nich zvlášť.



(a) (b) (c)

(d) (e) (f)

Obrázek 3.9: Ukázka několika záplat originálních snímků (a), (b), (c) a k nim příslušných anotací (d), (e), (f)

3.3 Trénování neuronové sítě

Aby bylo možné natrénovat neuronovou síť na vytvořené datové sadě, rozhodl jsem se použít model, který je velmi podobný již zmíněné architektuře Model A. Oba modely jsou prakticky totožné, rozdílné je pouze jejich provedení v programovacím jazyku python. Originální verzi použitého modelu je možné dohledat v repositáři na webových stránkách github.com [31]. Autor v tomto případě implementoval proměnnou velikost vstupní vrstvy neuronové sítě na základě zvoleného parametru, díky tomu je snadné vytvořit několik verzí modelu pro rozdílné velikosti vstupního snímku a jejich následné porovnání. Další výhodou je potom možnost pozměnit na základě vstupního parametru počet tříd, které bude model rozlišovat. Pro trénování na vlastní datové sadě tedy nebyl problém upravit tuto neuronovou síť pro segmentaci do sedmi tříd.

Celý proces trénování byl realizován v programovacím jazyku python za použití dostupných knihoven pro zpracování dat a pro práci s neuronovými sítěmi. V prvním kroku

probíhá načtení veškerých snímků obsažených v datové sadě pomocí knihovny opencv. Po načtení jsou obrazová data upravena tak, aby pořadí kanálů odpovídalo standardu RGB. Součástí modelu není příkaz pro automatickou normalizaci RGB dat a tento proces je tedy nutné provést ještě před vstupem původního snímku do neuronové sítě.

V dalším kroku je nutné rozdělit načtené snímky a anotace takovým způsobem, aby jejich velikost odpovídala velikosti vstupní vrstvy neuronové sítě. Jak bylo již zmíněno v kapitole Rešerše, při práci s konvolučními operacemi na stranách enkodéru i dekodéru je rozumné volit rozlišení vstupního snímku tak, aby velikost jeho hrany byla obecně 2^n, v praxi se typicky setkáváme s velikostí hrany snímku 128 nebo 256 pixelů. Na tuto skutečnost byl brán ohled při tvorbě vlastní datové sady, která zahrnuje obrazová data o konstantním rozlišení 1024 x 1024 pixelů. Každý z načtených snímků je tedy možné jednoduše rozdělit na několik částí pomocí knihovny patchify bez nutnosti snímky ořezávat. Stejný postup platí i pro veškeré anotace.

Než započne samotné trénování modelu, dochází ještě ke změně formátu anotovaných masek. Z obrazových dat, která jsou v tomto případě ve formátu RGB, je nejprve vytvořena matice celých čísel. Velikost této matice odpovídá rozlišení příslušné anotace a obsažená čísla symbolizují příslušnost pixelu na dané pozici ke konkrétní třídě. Tento formát je běžně nazýván "2D label". Požadujeme-li, aby trénovaný model segmentoval vstupní snímek obecně do k tříd, je tato 2D matice dále převedena na tenzor o k vrstvách. Každá vrstva tohoto tenzoru představuje jednu třídu a svými rozměry odpovídá zmíněné 2D matici. K-tá vrstva je tedy matice, která určuje příslušnost všech pixelů k-té třídě pomocí logických hodnot 1 nebo 0. Tento formát je označován jako "one-hot encode"a svými rozměry odpovídá výstupní vrstvě neuronové sítě.

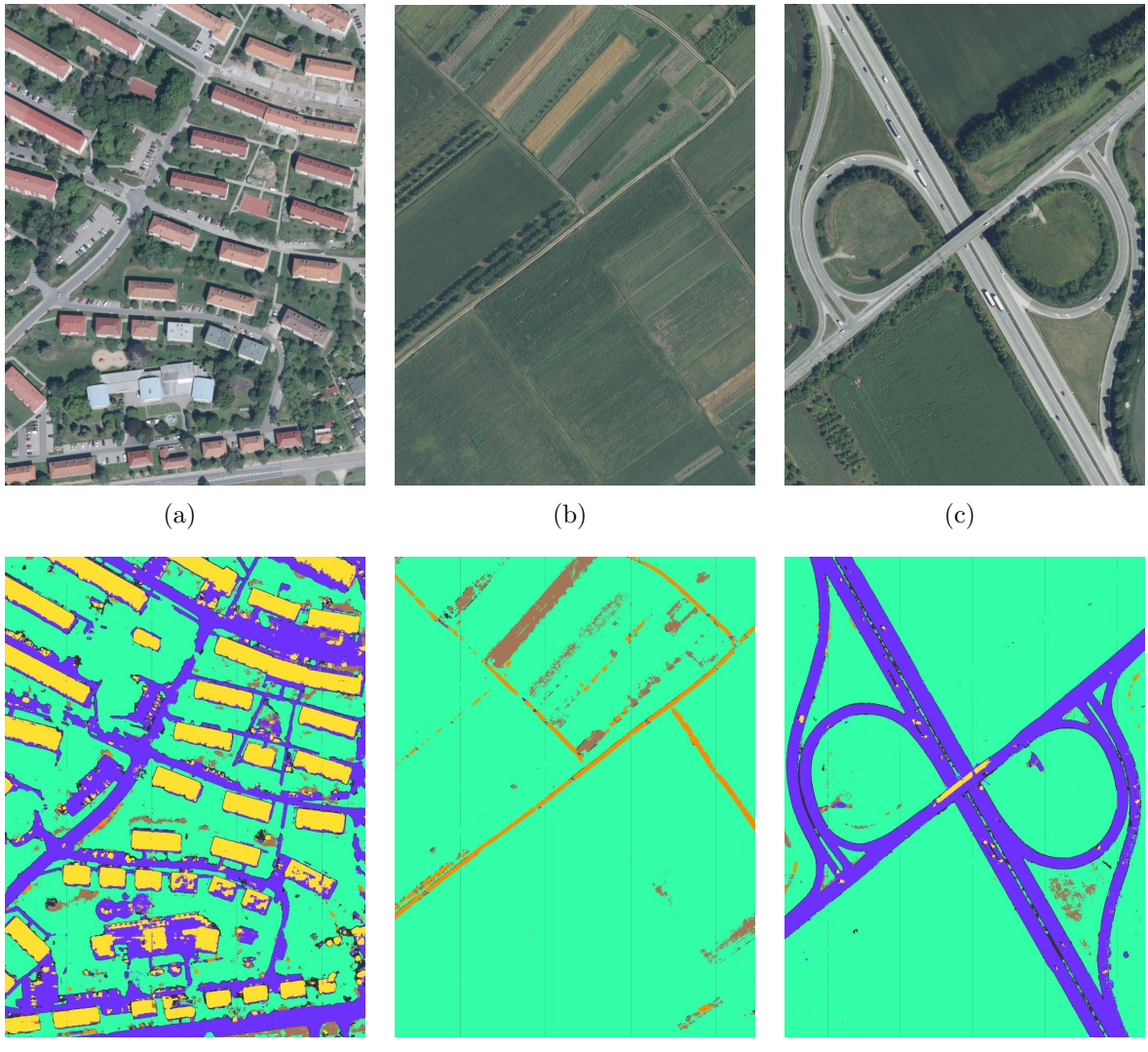
Posledním krokem ve zpracování dat je jejich rozdělení do trénovací a validační množiny. Jak napovídá označení, trénovací množina slouží k samotnému procesu trénování, zatímco množina validační je použita pro vyhodnocení aktuální kvality segmentace. Ve své práci jsem pro trénování vyčlenil 80% datové sady, zbylých 20% potom posloužilo pro validaci daného modelu.

Takto zpracovaná data potom vstupují do samotného modelu a pomocí známých procesů, které je možné dohledat v literatuře [6], probíhá jeho trénování. Pro práci s uvedeným modelem jsem ve své práci využil knihovnu tensorflow. Celý proces i výsledný model je ovlivněn několika významnými parametry, kterými jsou například velikost vstupního snímku, počet epoch, velikost dávky (batch size) nebo použitá ztrátová funkce (loss function). Tyto parametry jsem zpočátku zvolil stejné jako autor modelu. Na základě informací v článku [7] jsem následně provedl několik experimentů pro každý parametr a podle dosažených výsledků jsem zvolil jejich nejlepší kombinaci.

Určitými změnami procházel také použitý dataset, ze kterého byly například odstraněny velké vodní plochy. Modely, které byly natrénovány na kompletní datové sadě často zaměňovaly některé části vegetace právě za vodní plochu. Po odstranění dotyčných snímků a anotací došlo k výraznému zlepšení situace. Výsledná verze použité datové sady byla tedy o něco menší než verze původní a její celková velikost činila přibližně 50 MB.

Nejlepší model, který se podařilo natrénovat, dosahoval na validační množině dat hodnot Accuracy = 0,894 a Jaccard index = 0,763. Těchto výsledků se podařilo dosáhnout při velikosti vstupního snímku 512 x 512 pixelů a celkem 40 epochách trénování. Kompletní proces trénování včetně všech použitých parametrů je možné dohledat v již zmíněném repositáři na webových stránkách github.com [29].

Na obrázcích 3.10 a 3.11 je možné si prohlédnout snímky, které byly segmentovány pomocí nejlepšího natrénovaného modelu. Obrázek 3.10 zachycuje větší části terénu pro lepší představu o celkové přesnosti, zatímco na pozici 3.11 je k dipozici detailní pohled na chybně vyhodnocené oblasti. Z důvodu maximálního povoleného objemu dat se jedná o ukázky menšího formátu s uměle sníženým rozlišením. Celá řada segmentovaných snímků v plném rozlišení je k dispozici v repositáři [29].



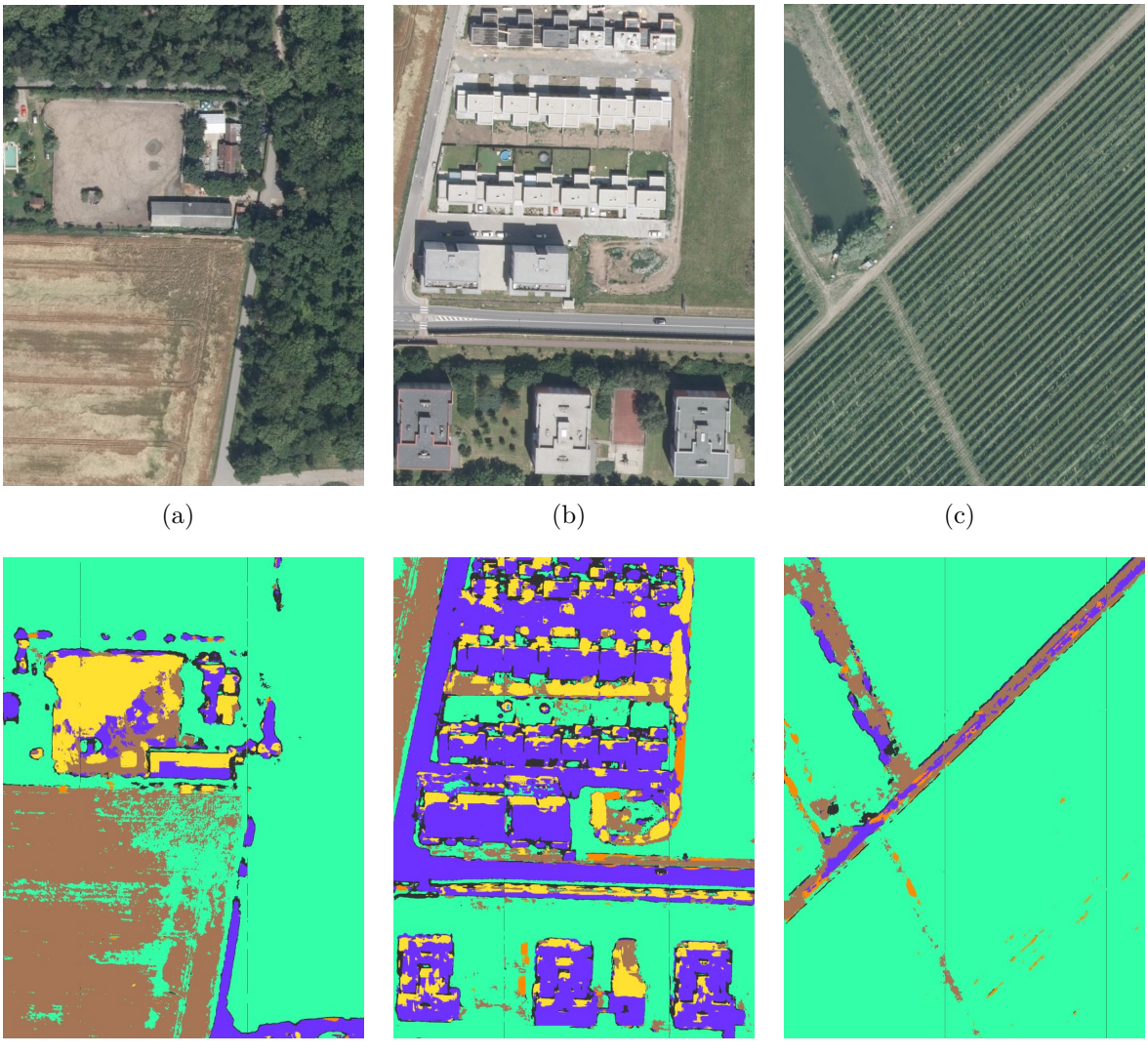
(a) (b) (c)

(d) (e) (f)

Obrázek 3.10: Ukázka několika snímků zemského povrchu (a), (b), (c) a příslušných predikcí natrénovaného modelu (d), (e), (f)

Kromě toho, že výsledný model dosáhl uspokojivých výsledků na validační množině dat během trénování, podává poměrně dobré výkony i při dalším testování. Jelikož hlavním úkolem modelu je správná identifikace zpevněných a nezpevněných cest, zaměříme se nejprve na vyhodnocení těchto dvou tříd. Pro lepší orientaci v uvedených příkladech uvádím výčet tříd včetně jejich barevné interpretace: vegetace (zelená), budova (žlutá), zpevněná cesta (fialová), nezpevněná cesta (oranžová), pole bez vegetace (hnědá), vodní plocha (světle modrá), neoznačeno/ostatní (šedá).

S vysokou přesností jsou zpravidla segmentovány ty zpevněné cesty, které jsou na originálním snímku dobře viditelné, problém ovšem nastává v případě, kdy je cesta překryta například korunou stromu nebo jakýmkoli stínem. Zejména na obrázku 3.11 (d) si můžeme všimnout, že jakkoli překrytá cesta není vyhodnocena správně. O něco horší je potom situace u cest nezpevněných, které jsou například v případě obrázku 3.11 (f) segmentovány nesprávně i v případě dobré viditelnosti.



(a) (b) (c)

(d) (e) (f)

Obrázek 3.11: Ukázka několika snímků zemského povrchu (a), (b), (c) a chybných predikcí natrénovaného modelu (d), (e), (f)

Poměrně rozporuplných výsledků dosahuje model při segmentaci budov. Natrénovaná neuronová síť identifikuje budovu podle střechy, která je na leteckém snímku zachycena, a různé typy střech segmentuje s různou přesností. Správně jsou vyhodnoceny zpravidla budovy, které mají při pohledu shora oranžovou nebo červenou barvu. Časté problémy se naopak vyskytují u objektů, jejichž střecha je šedá. V takových případech je celá budova nebo její část označena jako zpevněná cesta - například na obrázku 3.11 (e).

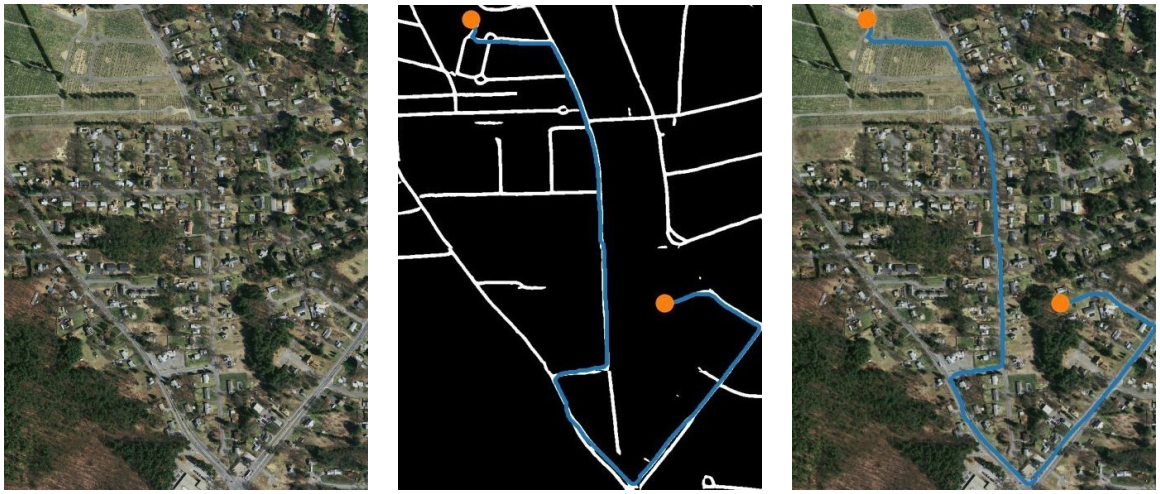
Za zmínku dále stojí třída vegetace, jejíž predikovaný výskyt velmi dobře odpovídá realitě. Opačných výsledků potom dosahuje třída vodní plocha, která není správně vy-

hodnocena téměř nikdy. Tento efekt můžeme pozorovat například na obrázku 3.11 (f). Výskyt těchto chyb je ovšem očekávaný, jelikož mezi trénovacími daty jsou vodní plochy zastoupeny minimálně.

Hlavní příčinou jmenovaných nedokonalostí je velmi omezené množství dat, na kterých byl použitý model natrénován. Stejně jako byl postupně zdokonalován proces trénování, i datová sada procházela jistým vývojem. Z tohoto důvodu není kvalita datové sady v celém jejím rozsahu konzistentní, což dále rozšiřuje prostor pro její zlepšení.

3.4 Plánování trasy

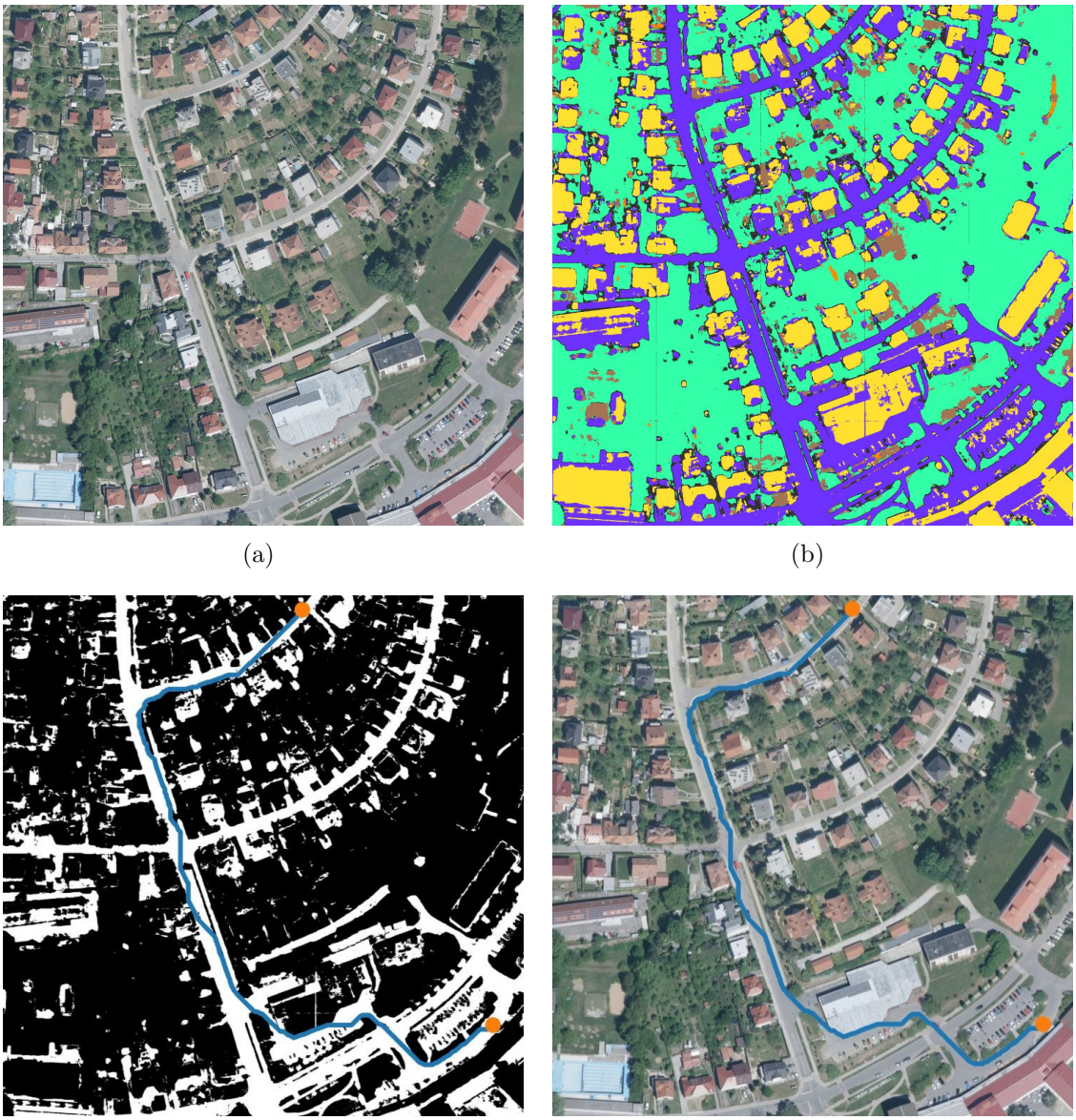
Na základě teorie, které je zmíněna v sekci A* algoritmus, jsem na závěr vybral několik správně segmentovaných snímků, na kterých jsem demonstroval možnost plánování trasy pro mobilního robota. Uvedené příklady jsou realizovány pomocí knihovny pathfinding, která je schopna rozlišovat pouze průjezdné a neprůjezdné oblasti. V případě snímků, které byly segmentovány Modelem B, je tato rozlišovací schopnost plně dostačující. Pro snímky, které zachycují zpevněné a nezpevněné cesty odděleně, by bylo dále možné algoritmus modifikovat takovým způsobem, aby upřednostňoval lépe průjezdné komunikace.



(a) (b) (c)

Obrázek 3.12: Příklad plánování trasy pomocí A* algoritmu: (a) Originální snímek z datové sady Massachusetts Roads Dataset, (b) predikce Modelu B s nalezenou trasou, (c) originální snímek s nalezenou trasou

V případě snímků, jenž jsou segmentovány pomocí vlastního natrénovaného modelu je nutné převést predikci neuronové sítě na binární masku. Tento převod provedeme jednoduše tak, že veškeré cesty označíme jako průjezdnou oblast a veškeré ostatní třídy jako oblast neprůjezdnou. Pokud by byla trasa plánována pro vozidlo s vysokou průchodností terénem, bylo by možné do průjezdných oblastí zahrnout i třídu pole bez vegetace. Zmíněný postup je znázorněn na obrázku 3.13



(a) (b)

(c) (d)

Obrázek 3.13: Příklad plánování trasy pomocí A* algoritmu: (a) Originální snímek z vytvořené datové sady, (b) predikce vlastního natrénovaného modelu, (c) predikce převedená na binární masku s nalezenou trasou, (d) originální snímek s nalezenou trasou

4 Závěr

Kapitola Rešerše představuje průzkum dostupných metod a datových sad relevantních pro sémantickou segmentaci leteckých snímků. Dva perspektivní před-trénované modely neuronových sítí byly potom vybrány k dalšímu testování.

Jako první prošel řadou experimentů Model A, který dosahoval obstojných výsledků na snímcích z datové sady Semantic segmentation of aerial imagery, na které byl také natrénován. Autor tohoto modelu dosáhnul během trénování hodnot Accuracy = 0,8616 a Jaccard index = 0,6599 na validační množině dat. V této práci vykazoval Model A během testování Accuracy = 0,8551 a Jaccard index = 0,5154. Testování bylo provedeno na celé datové sadě Semantic segmentation of aerial imagery. Stejný model následně posloužil pro segmentaci několika snímků z datových sad Massachusetts Roads Dataset a Landcover.ai. Predikce neuronové sítě ovšem zdaleka neodpovídaly realitě ani v jednom případě. Model A byl tedy vyhodnocen jako použitelný pouze pro jeho původní datovou sadu. Přesnost segmentace ale nebyla dostatečná pro plánování trasy segmentovanými snímky.

Na stejné trojici datových sad proběhlo testování i pro Model B, který byl natrénován na Massachusetts Roads Dataset. Tato neuronová síť dosáhla na svém původním datasetu pozoruhodné kvality segmentace, což dokládají i evaluační metriky Accuracy = 0,9836 a Jaccard index = 0,8024. U snímků z Landcover.ai obsahují výstupní snímky řadu nedokonalostí a dopravní síť je segmentována nesouvisle. V případě datové sady Semantic segmentation of aerial imagery jsou potom správně segmentovány pouze útržky cest. V průběhu testování byl prokázán značný vliv měřítka vstupních snímků na kvalitu segmentace. Pro optimální výsledky je nutné zachovat stejné měřítko jako během trénování. Při srovnání z hlediska výpočetního výkonu se tento model ukázal jako efektivnější než Model A. Modely bohužel nebylo možné otestovat v podmínkách různých ročních období, jelikož pro tyto účely nebyla nalezena vhodná datová sada.

Pro účely plánování trasy by bylo vhodné, aby výstupní snímky obsahovaly alespoň základní informace o segmentovaných cestách - tedy zda se jedná o povrch zpevněný či nezpevněný. Volně k dispozici ovšem není vhodný před-trénovaný model, ani datová sada, na které by bylo možné takový model natrénovat. Dalším krokem bylo tedy vytvoření vlastní datové sady, která je popsána kapitole 3.2. Tato datová sada dělí povrch originálních snímků do 7 tříd, mezi kterými jsou i dva druhy cest - zpevněné a nezpevněné. Celkový objem dat činí asi 62 MB a kompletní verzi datasetu je možné nalézt ve vytvořeném repositáři na webových stránkách github.com [29].

Součástí této práce je i trénování existující neuronové sítě, které je popsáno v kapitole 3.3. Během trénování na vlastní datové sadě se podařilo dosáhnout Accuracy = 0,894 a Jaccard index = 0,763 na validační množině dat. Při vizuálním ověření na testovacích datech se ukázalo, že získaný model je spolehlivý zejména pro segmentaci dobře viditelných zpevněných cest, zatímco cesty nezpevněné jsou ve výstupních predikcích zpravidla nesouvislé. Významným nedostatkem modelu je časté selhání při segmentaci budov s šedou střechou, které jsou v některých případech vyhodnoceny jako zpevněné cesty.

V závěru práce je potom demonstrována možnost plánování trasy na základě některých snímků, které jsou segmentovány s dostatečnou přesností. Konkrétní implementace algoritmu A* v případě použité knihovny ovšem není schopna odlišit více druhů cest a pracuje pouze s průjezdnou a neprůjezdnou oblastí snímku.

Jelikož se výsledný natrénovaný model ukázal jako funkční, nabízí se s výhledem do budoucna tuto práci dále rozvíjet a zdokonalovat. Velký význam by mělo rozšíření vlastní datové sady, jejíž omezený rozsah je hlavním limitem při trénování jakékoli neuronové sítě. Pro optimální výsledky by dále bylo vhodné natrénovat více modelů s rozdílnými architekturami a porovnat jejich vhodnost. Po dosažení uspokojivé kvality segmentace by dále bylo vhodné modifikovat algoritmus A* takovým způsobem, aby při plánování trasy upřednostňoval zpevněné komunikace před nezpevněnými.

Seznam použitých zdrojů

[1] BENCHAMARDIMATH, Basavaprasad; HEGADI, Ravindra. A Survey on Traditional and Graph Theoretical Techniques for Image Segmentation. *International Journal of Computer Applications* [online]. 2014, č. 0975 – 8887, s. 38–46 [cit. 2024-04-09]. Dostupné z: `https://www.researchgate.net/publication/274270045_A_Survey_on_Traditional_and_Graph_Theoretical_Techniques_for_Image_Segmentation`.

[2] *Image Thresholding Based on Otsu's Method using OpenCV* [online]. 2021. [cit. 2024-04-09]. Dostupné z: `https://lindevs.com/image-thresholding-based-on-otsus-method-using-opencv`.

[3] KOSARAJU, Vineet; RAGLAND, Davy; TRUONG, Adrien; NEHORAN, Effie; TOYUNGYERNSUB, Maneekwan. *Lecture 10: Semantic Segmentation and Clustering* [online]. 2005. [cit. 2024-05-22]. Dostupné z: `http://vision.stanford.edu/teaching/cs131_fall1718/files/10_notes.pdf`.

[4] IKONOMATAKIS, N.; PLATANIOTIS, K.N.; ZERVAKIS, M.; VENETSANOPOULOS, A.N. Region growing and region merging image segmentation. *Proceedings of 13th International Conference on Digital Signal Processing* [online]. 1997, s. 299–302 [cit. 2024-04-09]. ISBN 0-7803-4137-6. Dostupné z DOI: `10.1109/ICDSP.1997.628077`.

[5] PREETHA, Mary Synthuja Jain; SURESH, Padma; BOSCO, John. Image segmentation using seeded region growing. *International Conference on Computing, Electronics and Electrical Technologies (ICCEET)* [online]. 2012, roč. 2012, s. 576–583 [cit. 2024-04-09]. ISBN 978-1-4673-0212-8. Dostupné z DOI: `10.1109/ICCEET.2012.6203897`.

[6] CHOLLET, François. *Deep learning v jazyku Python: knihovny Keras, Tensorflow.* Grada Publishing 2019. Praha: Grada Publishing, 2019. ISBN 978-80-247-3100-1.

[7] LI, Zewen; LIU, Fan; YANG, Wenjie; PENG, Shouheng; ZHOU, Jun. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems* [online]. 2022, roč. 33, č. 12, s. 6999–7019 [cit. 2024-05-21]. ISSN 2162-237X. Dostupné z DOI: 10.1109/TNNLS.2021.3084827.

[8] RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Computer Vision and Pattern Recognition* [online]. 2015, č. 1505.04597, s. 4 [cit. 2024-04-13]. Dostupné z: https://arxiv.org/abs/1505.04597.

[9] ABDERRAHIM, Norel Ya Qine; ABDERRAHIM, Saadane; RIDA, Azmi. Road Segmentation using U-Net architecture. *2020 IEEE International conference of Moroccan Geomatics (Morgeo)* [online]. 2020, s. 1–4 [cit. 2024-04-13]. ISBN 978-1-7281-5806-8. Dostupné z DOI: 10.1109/Morgeo49228.2020.9121887.

[10] STEWART, Matthew. *Simple Introduction to Convolutional Neural Networks* [online]. 2019. [cit. 2024-05-22]. Dostupné z: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac.

[11] ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. *International Conference on Engineering and Technology (ICET)* [online]. 2017, s. 1–6 [cit. 2024-04-30]. ISBN 978-1-5386-1949-0. Dostupné z DOI: 10.1109/ICEngTechnol.2017.8308186.

[12] SULTANA, Farhana; SUFIAN, Abu; DUTTA, Paramartha. Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. *Knowledge-Based Systems* [online]. 2020, č. 201-202, s. 3–7 [cit. 2024-04-09]. ISSN 09507051. Dostupné z DOI: 10.1016/j.knosys.2020.106062.

[13] HESRAKI, Saba. *SegNET*. Sv. 2023 [online]. 2023. [cit. 2024-04-13]. Dostupné z: https://medium.com/@saba99/segnet-a139ce77b570.

[14] ZHAO, Hengshuang; SHI, Jianping; QI, Xiaojuan; WANG, Xiaogang; JIA, Jiaya. Pyramid Scene Parsing Network. *Computer Vision and Pattern Recognition* [online]. 2017, roč. 2017, č. 1612.01105, s. 1–11 [cit. 2024-05-22]. Dostupné z DOI: 10.48550/arXiv.1612.01105.

[15] NEUPANE, Bipul; HORANONT, Teerayut; ARYAL, Jagannath. Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis. *Remote Sensing* [online]. 2021, roč. 13, č. 4, s. 1–41 [cit. 2024-05-22]. ISSN 2072-4292. Dostupné z DOI: 10.3390/rs13040808.

[16] *Semantic segmentation of aerial imagery* [online]. 2020. [cit. 2024-04-13]. Dostupné z: https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery.

[17] BOGUSZEWSKI, Adrian; BATORSKI, Dominik; ZIEMBA-JANKOWSKA, Natalia; DZIEDZIC, Tomasz; ZAMBRZYCKA, Anna. LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands, Water and Roads from Aerial Imagery. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* [online]. 2021, roč. 2021, s. 1102–1110 [cit. 2024-04-13]. ISBN 978-1-6654-4899-4. Dostupné z DOI: 10.1109/CVPRW53098.2021.00121.

[18] MNIH, Volodymyr. *Machine Learning for Aerial Image Labeling* [online]. Toronto, Canada, 2013 [cit. 2024-04-13]. Dostupné z: https://www.cs.toronto.edu/~vmnih/data/. A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy. University of Toronto.

[19] MOSTEGEL, Christian; MAURER, Michael; HERAN, Nikolaus; PUERTA, Jesus Pestana; FRAUNDORFER, Friedrich. *Semantic drone dataset* [online]. 2019. [cit. 2024-04-13]. Dostupné z: https://www.tugraz.at/institute/icg/research/team-fraundorfer/software-media/dronedataset.

[20] CAI, Wenxiao; JIN, Ke; HOU, Jinyan; GUO, Cong; WU, Letian; YANG, Wankou. *VDD: Varied Drone Dataset for Semantic Segmentation*. 2023. Dostupné z arXiv: 2305.13608 [cs.CV].

[21] ALOUINI, YASSINE. *All the segmentation metrics!* [online]. 2022. [cit. 2024-04-14]. Dostupné z: https://www.kaggle.com/code/yassinealouini/all-the-segmentation-metrics%5C#What-about-Jaccard-and-IoU?.

[22] RESTREPO, Ronny. *Intersect over Union (IoU)* [online]. [cit. 2024-05-04]. Dostupné z: http://ronny.rest/tutorials/module/localization_001/iou/.

[23] DAVIES, Andrew Joseph. *Semantic Segmentation of Aerial Imagery Using U-Net in Python* [online]. 2022. [cit. 2024-04-14]. Dostupné z: https://towardsdatascience.com/semantic-segmentation-of-aerial-imagery-using-u-net-in-python-552705238514.

[24] DAVIES, Andrew Joseph. *U-net-aerial-imagery-segmentation* [online]. 2022. [cit. 2024-05-22]. Dostupné z: https://github.com/ad-1/u-net-aerial-imagery-segmentation.

[25] KLINGLER, Nico. *The Role of Batch Normalization in CNNs* [online]. [cit. 2024-05-05]. Dostupné z: https://viso.ai/deep-learning/batch-normalization/.

[26] SHAH, Parshwa. *Map-Segmentation* [online]. 2020. [cit. 2024-04-14]. Dostupné z: https://github.com/parshwa1999/Map-Segmentation/tree/master/docs/static/models/Massachusetts_Roads_and_Building_Dataset.

[27] ZAREMBO, Imants; KODORS, Sergejs. Pathfinding Algorithm Efficiency Analysis in 2D Grid. *Environment. Technology. Resources. Proceedings of the International Scientific and Practical Conference* [online]. 2015-08-08, roč. 2015, č. vol. 2, s. 46–50 [cit. 2024-04-19]. ISSN 2256-070X. Dostupné z DOI: 10.17770/etr2013vol2.868.

[28] SEBO, Sarah. *Class Meeting 08: Path Finding* [online]. 2011. [cit. 2024-05-22]. Dostupné z: https://classes.cs.uchicago.edu/archive/2022/spring/20600-1/class_meeting_08.html.

[29] PAZDERA, Jiří. *Semantic-segmentation-of-aerial-images* [online]. 2024. [cit. 2024-05-22]. Dostupné z: https://github.com/JiriPazdera/Semantic-segmentation-of-aerial-images.

[30] *Geoprohlížeč* [online]. 2018. [cit. 2024-05-22]. Dostupné z: https://ags.cuzk.cz/geoprohlizec/.

[31] BHATTIPROLU, Sreenivas. *Python for microscopists* [online]. 2021. [cit. 2024-04-13]. Dostupné z: https://github.com/bnsreenu/python_for_microscopists.

Seznam obrázků