



Zadání diplomové práce

| | |
|-----------------------------|--|
| Název: | Identifikace ochranných hrází a náspů liniových staveb |
| Student: | Bc. Jan Kostecký |
| Vedoucí: | Ing. Petr Pulc |
| Studijní program: | Informatika |
| Obor / specializace: | Znalostní inženýrství |
| Katedra: | Katedra aplikované matematiky |
| Platnost zadání: | do konce letního semestru 2020/2021 |

Pokyny pro vypracování

V kvalitních digitálních výškových modelech (DEM) s horizontálním rozlišením pod 10 m lze jasně identifikovat liniové struktury ochranných hrází, železnic a silnic v říčních nivách, které významně ovlivňují rozsah záplav. Tyto struktury jsou podstatné pro modelování povodňových rizik, proto je vhodné tyto struktury identifikovat a vektorizovat. Pro zpřesnění je možné využít satelitní obrazové snímky (např. mise Sentinel-2), zejména v případech, kdy je šířka stavby užší než rozlišení digitálního modelu terénu.

- Prozkoumejte dodané digitální výškové modely a satelitní snímky.
 - Navrhněte metodu zarovnání dat.
 - Prozkoumejte metody strojového učení vhodné pro zpracování dat (např. konvoluční neuronové sítě).
 - Navrhněte metodu rozpoznávání a odhadu polohy liniových staveb v lokálně degradovaném výškovém modelu.
 - Navrhněte a implementujte nástroj, který vytvoří reálnou, ale co nejspojitější vektorovou reprezentaci vedení korun hrází nebo náspů ve vhodném GIS formátu.
-



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Diplomová práce

Identifikace ochranných hrází a náspů liniových staveb

Bc. Jan Kostecký

Katedra aplikovaná matematika

Vedoucí práce: Ing. Petr Pulc

6. května 2021

Poděkování

Děkuji vedoucímu práce Ing. Petru Pulcovi za pomoc.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 6. května 2021

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Jan Kostecký. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Kostecký, Jan. *Identifikace ochranných hrází a naspů liniových staveb*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Ochranné hráze a násypy liniových staveb výrazně ovlivňují rozsah záplav. Tato práce navrhuje postup, jak tyto struktury identifikovat z dat dálkového průzkumu Země a vektorizovat je do GIS formátu. Vstupem pro identifikaci jsou především digitální výškové modely; pro další zpřesnění je pak otestováno použití satelitních snímků mise Sentinel-2. K rozpoznání ochranných hrází a liniových staveb jsou využity konvoluční neuronové sítě.

Klíčová slova Konvoluční neuronová síť, U-net, Sentinel-2, Digitální výškový model

Abstract

Levees and embankments have a significant effect on the magnitude of flooding. This work deals with the identification of such structures and their conversion to vector GIS format. The inputs for the proposed algorithm are primarily digital terrain models; for further improvements, Sentinel-2 satellite images are used. For the detection of levees and embankments, an approach based on convolutional neural networks is proposed.

Keywords Convolutional neural net, U-net, Sentinel-2, Digital terrain model

Obsah

| | |
|--|-----------|
| Úvod | 1 |
| 1 Základní pojmy | 3 |
| 1.1 Digitální výškové modely | 3 |
| 1.2 Souřadnicové systémy | 4 |
| 1.3 Sentinel | 5 |
| 1.4 Formáty ukládání dat | 6 |
| 1.4.1 Rastrové formáty | 6 |
| 1.4.1.1 TIFF, GeoTIFF | 6 |
| 1.4.1.2 JPEG 2000 | 7 |
| 1.4.2 Vektorové formáty | 7 |
| 1.4.2.1 GeoJSON | 7 |
| 1.4.2.2 Esri Shapefile | 8 |
| 1.5 Ochranné hráže a náspy liniových staveb | 9 |
| 2 Algoritmy | 11 |
| 2.1 Strojové učení | 11 |
| 2.1.1 Přístupy k učení | 11 |
| 2.1.2 Rozdělení učení podle cíle | 11 |
| 2.1.2.1 Regrese | 11 |
| 2.1.2.2 Klasifikace | 12 |
| 2.1.2.3 Shlukování | 12 |
| 2.2 Algoritmy počítačového vidění | 12 |
| 2.2.1 Sémantická segmentace | 12 |
| 2.3 Umělé neuronové sítě | 13 |
| 2.3.1 Perceptron | 13 |
| 2.3.2 Vícevrstvá neuronová síť | 14 |
| 2.3.2.1 Aktivační funkce pro skryté vrstvy | 14 |
| 2.3.2.2 Aktivační funkce pro výstupní vrstvy | 15 |

| | | |
|----------|---|-----------|
| 2.4 | Hluboké učení | 16 |
| 2.5 | Konvoluční neuronové sítě | 16 |
| 2.5.1 | Konvoluční vrstva | 17 |
| 2.5.2 | Pooling vrstva | 17 |
| 2.6 | Architektury konvolučních sítí vhodných pro sémantickou segmentaci | 17 |
| 2.6.1 | U-Net | 18 |
| 2.6.2 | LinkNet | 18 |
| 2.6.3 | PSPnet | 19 |
| 2.7 | Vyhodnocení úspěšnosti modelů | 19 |
| 2.7.1 | Matice záměn | 21 |
| 2.7.2 | Intersection over Union | 22 |
| 2.8 | Účelová funkce | 22 |
| 2.9 | Morfologické operace | 22 |
| 2.9.1 | Dilatace | 22 |
| 2.9.2 | Eroze | 23 |
| 2.9.3 | Skeletonizace | 23 |
| 3 | Analýza problému a návrh řešení | 25 |
| 3.1 | Vstupní data | 25 |
| 3.1.1 | Digitální výškové modely | 25 |
| 3.1.2 | Satelitní snímky | 26 |
| 3.1.3 | Ochranné hráze a násypy liniových staveb | 26 |
| 3.2 | Možná řešení | 26 |
| 3.2.1 | Morfometrické charakteristiky | 28 |
| 3.2.2 | Sémantická segmentace | 28 |
| 3.3 | Zvolené řešení | 28 |
| 4 | Implementace | 31 |
| 4.1 | Použité knihovny | 31 |
| 4.2 | Předzpracování vstupních dat | 32 |
| 4.2.1 | Rasterizace vektorů ochranných hrází | 32 |
| 4.2.2 | Sloučení satelitních snímků | 32 |
| 4.2.3 | Převod na stejný SRS | 33 |
| 4.2.4 | Prozkoumání předzpracovaných vstupních dat | 33 |
| 4.3 | Přehled datových sad | 34 |
| 4.3.1 | Augmentace dat | 35 |
| 4.4 | Trénování CNN pro sémantickou segmentaci | 35 |
| 4.4.1 | Parametry modelů typu CNN | 35 |
| 4.4.2 | Model založený na digitálním výškovém modelu | 36 |
| 4.4.3 | Model založený na digitálním výškovém modelu a satelitních snímcích | 36 |
| 4.5 | Návrh vlastní CNN typu U-net | 38 |
| 4.5.1 | Model založený na DEM | 38 |

| | | |
|----------------------------------|--|-----------|
| 4.5.1.1 | Zhodnocení a interpretace výsledků | 38 |
| 4.5.2 | Model založený na DEM a satelitních snímcích | 39 |
| 4.5.2.1 | Zhodnocení a interpretace výsledků | 41 |
| 4.6 | Predikce | 42 |
| 4.7 | Post-processing | 43 |
| 4.7.1 | Eroze | 43 |
| 4.7.2 | Dilatace | 44 |
| 4.7.3 | Úprava dle DEM | 44 |
| 4.7.4 | Skeletonizace | 45 |
| 4.8 | Transformace do vektorového formátu | 46 |
| 4.9 | Zhodnocení výsledků | 46 |
| 4.10 | Návrhy na možné budoucí vylepšení | 47 |
| Závěr | | 51 |
| Literatura | | 53 |
| A Seznam použitých zkratk | | 57 |
| B Obsah příloženého CD | | 59 |

Seznam obrázků

| | | |
|------|--|----|
| 1.1 | DEM vs. DSM [1] | 4 |
| 2.1 | Příklad sémantické segmentace [2] | 13 |
| 2.2 | jednovrstvý perceptron [3] | 14 |
| 2.3 | Logistická funkce | 16 |
| 2.4 | Vícevrstvá neuronová síť [3] | 16 |
| 2.5 | Konvoluční neuronová síť [4]. | 17 |
| 2.6 | Schéma plně konvoluční neuronové sítě [5]. | 18 |
| 2.7 | Příklad U-net architektury převzatý z [6]. | 19 |
| 2.8 | Architektura LinkNet [7]. | 20 |
| 2.9 | Architektura PSPnet [8]. | 20 |
| 2.10 | Vlevo původní stav, vpravo stav po aplikaci dilatace. [9] | 23 |
| 2.11 | Vlevo původní stav, vpravo stav po aplikaci eroze. [9] | 23 |
| 2.12 | Vlevo původní stav, vpravo stav po skeletonizaci. [?] | 24 |
| 3.1 | Vizualizace dat z digitálního výškového modelu | 26 |
| 3.2 | RGB satelitní snímek | 27 |
| 3.3 | Příklad vektorů hrází a náspů liniových staveb | 27 |
| 4.1 | Vlevo vektory ochranných hrází, vpravo po rasterizaci. Jako pozadí je použit digitální výškový model. | 32 |
| 4.2 | Příklad pravděpodobně nesprávně anotovaných dat | 34 |
| 4.3 | Průběh trénování modelu založeném na DEM. Trénováno na datech <code>austria_01</code> , <code>austria_02</code> a otestováno na <code>austria_03</code> . | 39 |
| 4.4 | Úspěšná predikce modelu založeném na DEM. Vlevo anotované hráze, uprostřed predikce, vpravo DEM. | 40 |
| 4.5 | Neúspěšná predikce modelu založeném na DEM. Vlevo anotované hráze, uprostřed predikce, vpravo DEM. | 40 |
| 4.6 | Průběh trénování modelu založeném na DEM a satelitních snímcích. Trénováno na datech <code>austria_01</code> , <code>austria_02</code> a otestováno na <code>austria_03</code> . | 41 |

| | | |
|------|--|----|
| 4.7 | Ukázka predikcí, kde jsou satelitních snímky přínosné. Zleva – anotovaná data, predikce využívající satelitní snímky, predikce využívající pouze DEM, satelitní snímek, DEM. | 42 |
| 4.8 | Vlevo po ukázka predikce, kde každý pixel nese hodnotu pravděpodobnosti, že se na jeho poloze nachází hráz. Vpravo ukázka predikce po binarizaci. | 43 |
| 4.9 | Vlevo stav před aplikací eroze, vpravo po. | 44 |
| 4.10 | Vlevo stav před aplikací dilatace, vpravo po. | 44 |
| 4.11 | Vlevo problém, který může nastat pokud je koryto řeky úzké a hráze jsou na obou stranách řeky. Vpravo opravení odečtením bodů, které jsou nižší, než jejich okolí. | 45 |
| 4.12 | Vlevo stav před skeletonizací, vpravo stav po skeletonizaci. | 45 |

Seznam tabulek

| | | |
|-----|--|----|
| 2.1 | Matice záměn | 21 |
| 4.1 | Model s použitím DEM jako vstupu, natrénován na datech <code>austria_01</code> , <code>austria_02</code> a otestován na <code>austria_03</code> | 37 |
| 4.2 | Model s použitím DEM jako vstupu, natrénován na datech <code>poland</code> a otestován na <code>czech</code> | 37 |
| 4.3 | Model s použitím DEM a satelitních snímků Sentinel-2, natrénován na datech <code>austria_01</code> , <code>austria_02</code> a otestován na <code>austria_03</code> | 37 |
| 4.4 | Model s použitím DEM a satelitních snímků Sentinel-2, natrénován na datech <code>poland</code> a otestován na <code>czech</code> | 37 |
| 4.5 | Úspěšnost modelu založeném na DEM, natrénovaném a otesto- vaném na různých datech. | 39 |
| 4.6 | Úspěšnost modelu založeném na DEM a satelitních snímcích, natrénovaném a otestovaném na různých datech. | 42 |

Úvod

Ochranné hráze a násypy liniových staveb (jako jsou silnice a železnice) výrazně ovlivňují rozsah záplav, především pak v říčních nivách. Pro zpřesnění modelu rizik a rozsahu případných povodní je velmi užitečné mít jako jeden ze vstupů povodňového modelu takovéto struktury ve vektorovém formátu. V digitálních výškových modelech (DEM) s vysokým horizontálním rozlišením je možné liniové stavby přímo rozpoznat. Ne vždy je ovšem dostatečně vysoké rozlišení digitálního výškového modelu dostupné. V tom případě je možné využít pro zpřesnění obrazové satelitní snímky pořízené satelity z mise Sentinel-2. Ty mohou pomoci především v těch případech, kdy jsou ochranné hráze či násypy liniových staveb výrazně užší, než rozlišení DEM.

Cílem této práce je prozkoumat metody identifikace vyvýšených liniových staveb v digitálním výškovém modelu. A následně ověřit, zda je přínosné jako další vstup pro zpřesnění použít obrazové satelitní snímky. Z takto identifikovaných staveb pak vytvořit co nejspojitéjší, ale zároveň co nejvíce reálnou, vektorovou reprezentaci a následně převést do vhodného vektorového GIS formátu.

V první kapitole je poskytnut přehled základních pojmů, kterým je potřeba porozumět pro pochopení problematiky. V druhé kapitole jsou definovány a rozebrány používané metody a algoritmy. Třetí kapitola se zabývá analýzou řešeného problému, přehledem možných metod řešení a volbou vhodného řešení pro tuto práci. Čtvrtá kapitola obsahuje popis procesu předzpracování dat, implementační detaily a informace o trénování modelů. Dále je ve čtvrté kapitole řešena úprava a vektorizace výstupu modelu a vyhodnocení úspěšnosti celkové úspěšnosti.

Základní pojmy

1.1 Digitální výškové modely

Digitální výškové modely (digital elevation model, DEM) slouží k popisu výškové struktury povrchu zkoumaného objektu, zpravidla (a také v této práci) jde o povrch Země.

Výškový model může být reprezentován například rastrovou mřížkou s informací o své poloze (zeměpisné šířce a délce hranic mřížky), kde každý bod této pravidelné čtvercové mřížky má přiřazenou informaci o nadmořské výšce.

Jinou možností, jak reprezentovat výškový model, je vektorová metoda zvaná TIN (triangulated irregular network). V té je povrch objektu reprezentován nepravidelnou sítí trojúhelníků, kde má zeměpisnou šířku, délku a elevaci definovaný každý uzel sítě. Tento typ výškového modelu ovšem není v této práci používán a nebude dále diskutován.

Při popisu digitálních výškových modelů se běžně užívají tři zkratky:

DEM Digital elevation model – Digitální výškový model

DTM Digital terrain model – Digitální model terénu

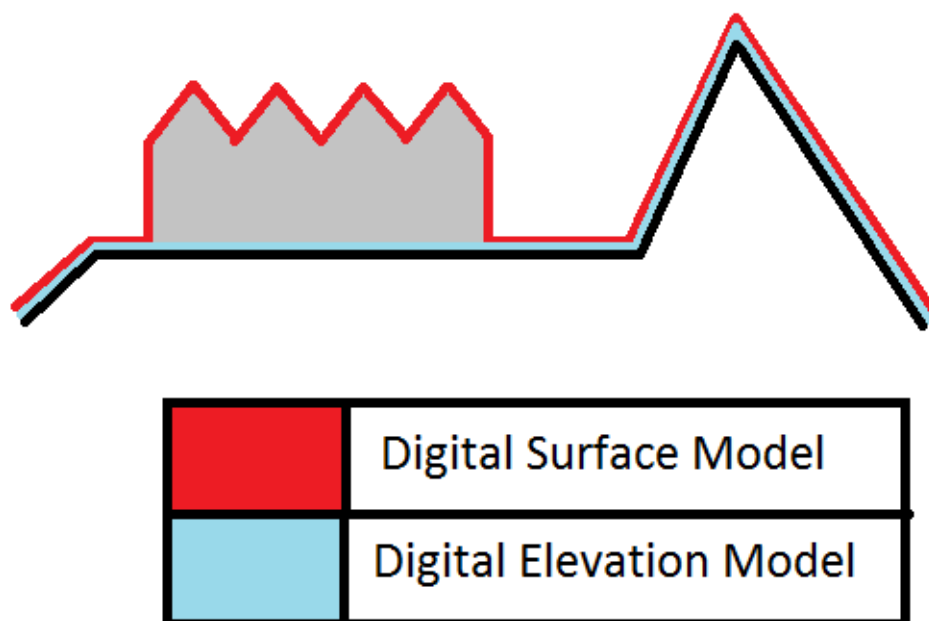
DSM Digital surface model – Digitální model povrchu

Napříč literaturou nejsou tyto pojmy striktně a jasně definovány, někdy je DEM (digital elevation model) používán jako obecný výraz pro digitální výškový model, jindy je zase používán jako synonymum k DTM (digital terrain model).

DTM obvykle popisuje nadmořskou výšku terénu bez člověkem uměle vytvořených staveb, ale i bez dalších přírodních objektů jako jsou například stromy. Někdy je ovšem DTM obohacen vektorovými informacemi o poloze řek, příkopů či hřebenů.

DSM (digital surface model) pak typicky popisuje výšku povrchu včetně staveb a všech dalších objektů, které odráží světlo.

Rozdíl mezi DTM a DSM ilustruje obrázek 1.1



Obrázek 1.1: DEM vs. DSM [1]

1.2 Souřadnicové systémy

Souřadnicový systém (SRS – spatial reference system, nebo také CRS – coordinate reference system) definuje, jak se budou geografická data matematicky transformovat tak, aby byla s co nejmenším zkreslením zobrazitelná na ploché mapě.

Každý souřadnicový systém je určen několika parametry:

- referenční těleso – Slouží jako aproximace popisu tvaru země, případně prostoru na který chceme polohově odkazovat. Typicky jde o elipsoid, kouli či rovinu.
- poloha nultého poledníku – U globálních souřadnicových systémů je většinou použit mezinárodní nultý Greenwichský poledník.
- typ zobrazení referenční plochy na plochu rovinnou
- definice počátku, jednotky míry a orientace systému souřadnic

Který souřadnicový systém použít závisí na tom, na jakou plochu na zemi chceme polohově odkazovat. Například souřadnicový systém S-JTSK (systém jednotné trigonometrické sítě katastrální), který vychází z Křovákova zobrazení, má definované zobrazení pouze pro body v České a Slovenské republice

a okolí, ale zato je pro definovaný prostor výrazně přesnější, než systémy globální.

Všechny běžně používané souřadnicové systémy jsou jednoznačně identifikované pomocí SRID (spatial reference identifier). Například souřadnicový systém S-JTSK s Greenwichským nultým poledníkem má autoritou EPSG (European Petroleum Survey Group) přidělený kód 5514. Díky tomu je možné v GIS datech přesně specifikovat jaký souřadnicový systém používají.

1.3 Sentinel

Sentinel je skupina misí ESA (evropské kosmické agentury), které slouží pro účely programu Copernicus.

Program Copernicus obsahuje komplexní systémy zajišťující monitorování země. Tyto systémy obstarávají sběr dat jak ze satelitů, jako v případě mise Sentinel, tak z pozemních stanic, letadel a dalších zdrojů.

Mise Sentinel je rozdělena do několika částí a každá z nich sbírá různá data:

- Sentinel-1 – Dvě družice, které monitorují jak pevninu, tak moře, oceány a stavy mořských ledů.
- Sentinel-2 – Provádí multispektrální snímkování krajiny, slouží převážně k monitorování změn ploch a změn jejich využití.
- Sentinel-3 – Mise má za úkol měřit topografii povrchu moře, teplotu a barvu moře i země.
- Sentinel-4 – Není samostatná družice, ale jde o modul umístěný na meteorologické družici MTG. Monitoruje stav atmosféry a plynů, které jsou určující pro kvalitu ovzduší.
- Sentinel-5 – Monitoruje atmosféru, zjišťuje její chemické složení, provádění měření související s vrstvou ozónu a UV zářením.

Pro tuto práci je zajímavá především mise Sentinel-2, která poskytuje snímky země ve 13 spektrálních pásmech s prostorovým rozlišením 10–60 m na pixel.

V nejjemnějším rozlišení 10 m na pixel jsou k dispozici především 3 pásma viditelného spektra, a to červené (se střední vlnovou délkou 665 nm), zelené (560 nm) a modré (490 nm). Při vhodném složení informací z těchto pásem a atmosférické korekci vzniká takzvaný barevně věrný snímek (true color composite), který barevně odpovídá pozorování lidským okem z úrovně terénu.

1.4 Formáty ukládání dat

V oboru GIS existuje množství způsobů jak ukládat geografická data. Jaký formát zvolit obvykle záleží na tom, jak bude s daty dále nakládáno a v jakých softwarových nástrojích s nimi budeme pracovat.

Formáty můžeme v základu rozdělit na rastrové a vektorové. Rastrové formáty mají fixní mřížku jednotlivých pixelů, u vektorových formátů jsou data uložena jako popisy křivek.

1.4.1 Rastrové formáty

1.4.1.1 TIFF, GeoTIFF

Formát TIFF (Tagged Image File Format) je jedním z nejpoužívanějších formátů rastrové grafiky v různých oborech, a to především kvůli své flexibilitě.

Tento formát umožňuje ukládat jak s kompresí, tak bez komprese. Oproti jiným běžným rastrovým formátům umožňuje uložit více samostatných obrazových vrstev v jednom souboru.

Nevýhodou grafiky ve formátu TIFF je typicky větší velikost souborů a omezená maximální velikost souboru na 4 GB.

GeoTIFF je rozšíření formátu TIFF o geografická metadata. Příklad takových metadat poskytuje ukázka 1.1. Důležitá je například informace o použitém souřadnicovém systému (řádky 5–17), velikost jednoho pixelu ve skutečnosti v metrech (řádek 19) a souřadnice všech rohů obrázku v použitém souřadnicovém systému (řádky 25–28).

Ukázka 1.1: Metadata GeoTIFF.

```
1 Driver: GTiff/GeoTIFF
2 Files: Morava_10m.tif
3     Morava_10m.tif.aux.xml
4 Size is 2501, 7426
5 Coordinate System is:
6 LOCAL_CS["S-JTSK_Krovak_East_North",
7     GEOGCS["S-JTSK",
8         DATUM["System_Jednotne_Trigonometricke_Site_Katastralni",
9             SPHEROID["Bessel 1841",6377397.155,299.1528128000033,
10                 AUTHORITY["EPSG","7004"]],
11                 AUTHORITY["EPSG","6156"]],
12                 PRIMEM["Greenwich",0],
13                 UNIT["degree",0.0174532925199433],
14                 AUTHORITY["EPSG","4156"]],
15                 AUTHORITY["EPSG","5514"],
16                 UNIT["metre",1,
17                     AUTHORITY["EPSG","9001"]]]]
18 Origin = (-554197.5000000000000000,-1122092.5000000000000000)
19 Pixel Size = (10.0000000000000000,-10.0000000000000000)
20 Metadata:
21     AREA_OR_POINT=Area
22 Image Structure Metadata:
```

```

23 INTERLEAVE=BAND
24 Corner Coordinates:
25 Upper Left ( -554197.500 , -1122092.500)
26 Lower Left ( -554197.500 , -1196352.500)
27 Upper Right ( -529187.500 , -1122092.500)
28 Lower Right ( -529187.500 , -1196352.500)
29 Center ( -541692.500 , -1159222.500)
30 Band 1 Block=128x128 Type=Float32, ColorInterp=Gray
31 Min=160.365 Max=355.605
32 Minimum=160.365, Maximum=355.605, Mean=204.073, StdDev=25.989
33 NoData Value=-3.4028234663852886e+38
34 Metadata:
35 STATISTICS_MAXIMUM=355.60498046875
36 STATISTICS_MEAN=204.07262062397
37 STATISTICS_MINIMUM=160.36500549316
38 STATISTICS_SKIPFACTORX=1
39 STATISTICS_SKIPFACTORY=1
40 STATISTICS_STDDEV=25.989020052022

```

1.4.1.2 JPEG 2000

JPEG 2000 je následníkem známého a rozšířeného formátu JPEG. Jsou u něj používány techniky komprese založené na vlnkové kompresi [10], což umožňuje oproti původnímu JPEG lepší zachování kvality grafiky i při stejné velikosti souboru. Formát umožňuje ale i bezztrátovou kompresi. Dle [10] byly při tvorbě tohoto standardu ukládání dat zohledněny také požadavky GIS komunity, díky čemuž je možné využít rychlý přístup k prostorovým umístěním, větší bitovou hloubku či rozdělení obrazu na několik dlaždic s kterými je pak možno pracovat nezávisle.

1.4.2 Vektorové formáty

1.4.2.1 GeoJSON

GeoJSON je prostý textový formát používající syntaxi formátu JSON (JavaScript Object Notation). Nad rámec tohoto formátu, který obsahuje pouze definice zápisu běžných datových typů a kolekcí, pak GeoJSON definuje několik typů objektů (např. Position, Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon), které jsou blíže specifikovány v [11].

Pro tuto práci je důležitý především typ objektu LineString (lomená čára), jelikož námi prováděná vektorizace poskytuje právě čárové segmenty. Objekt se pak definuje pomocí seznamu souřadnic bodů, které tuto lomenou čáru tvoří. Příklad takového zápisu je uvedený v ukázce 1.2.

Ukázka 1.2: Ukázka objektu typu LineString s definicí souřadnic bodů.

```

1 {
2   "geometry": {
3     "type": "LineString",
4     "coordinates": [[0,10],[1,11],[2,11]]

```



```
5   },
6   "properties": {
7     "name": "Hráz"
8   }
9 }
```

1.4.2.2 Esri Shapefile

Otevřený formát pro ukládání vektorových GIS dat Shapefile byl vyvinut společností Esri a je podporován většinou komerčních i opensource nástrojů. I proto je velmi populární volbou.

Formát je uložen jako sada souborů se stejným názvem ale různými příponami. Povinné jsou tři soubory:

- `.shp` – hlavní soubor, který obsahuje záznam o každém objektu, uložený jako seznam bodů, který tento objekt definují
- `.shx` – index záznamů v hlavním `.shp` souboru, obsahuje 100 bajtovou hlavičku a poté 8-bajtové záznamy o umístění každého objektu v `.shp` souboru
- `.dbf` – dodatečné atributy ve formátu dBase [12] pro objekty v `.shp` souboru

Kromě těchto tří povinných souborů je možné použít i několik dalších, volitelných:

- `.prj` – informace o souřadnicovém systému a geodetickému datu [13], příklad v ukázce 1.3
- `.cpg` – specifikace kódování znaků v souboru `.dbf`
- `.shp.xml` – metadata ve formátu XML

Ukázka 1.3: Informace k souřadnému systému v Esri Shapefile.

```
1 PROJCS["S-JTSK_Krovak_East_North",
2     GEOGCS["GCS_S_JTSK",
3         DATUM["D_S_JTSK",SPHEROID
4             ["Bessel_1841",6377397.155,299.1528128]
5         ],
6         PRIMEM["Greenwich",0.0],
7         UNIT["Degree",0.0174532925199433]
8     ],
9     PROJECTION["Krovak"],
10    PARAMETER["False_Easting",0.0],
11    PARAMETER["False_Northing",0.0],
12    PARAMETER["Pseudo_Standard_Parallel_1",78.5],
13    PARAMETER["Scale_Factor",0.9999],
14    PARAMETER["Azimuth",30.28813975277778],
```

```
15     PARAMETER["Longitude_of_Center",24.83333333333333],
16     PARAMETER["Latitude_of_Center",49.5],
17     PARAMETER["X_Scale",-1.0],
18     PARAMETER["Y_Scale",1.0],
19     PARAMETER["XY_Plane_Rotation",90.0],
20     UNIT["Meter",1.0]
21 ]
```

1.5 Ochranné hráze a náspy liniových staveb

Obecně je liniová stavba definovaná jako stavba, kde podstatně převažuje délka nad výškou a šířkou. V kontextu této práce jsou pak zajímavé především takové liniové stavby, které jsou oproti okolí vyvýšené, a v případě povodní tvoří překážku pro rozlévání vody do okolí.

Algoritmy

2.1 Strojové učení

Strojové učení je podoblastí umělé inteligence (AI), která se zabývá schopností počítače řešit úlohy samostatně bez explicitního naprogramování.

2.1.1 Přístupy k učení

Algoritmy strojového učení můžeme rozdělit na tři základní podoblasti podle toho, jakým způsobem je k učení přistupováno.

- **Učení bez učitele** (Unsupervised learning) – jsou k dispozici pouze vstupní data, není znám správný výstup. Nelze tedy obecně a snadno určit jaká je úspěšnost učení.
- **Učení s učitelem** (Supervised learning) – ke vstupním datům je znám i správný výstup. Je tedy možné vyhodnotit jak bylo učení úspěšné a podle toho dále upravovat parametry naučeného modelu.
- **Zpětnovazební učení** (Reinforcement learning) – v této technice strojového učení se učí agent, kterému je umožněno interagovat s okolním prostředím a učit se na základě svých chyb a zkušeností. Zatím co u učení s učitelem by agent dostal seznam správných výstupů ke vstupům, u zpětnovazebního učení dostává buď pozitivní nebo negativní ohodnocení svých akcí, na jehož základě provádí kroky další.

2.1.2 Rozdělení učení podle cíle

2.1.2.1 Regrese

Cílem regrese je odhadnout funkci, která ke každému vstupu přiřadí jednu výstupní číselnou hodnotu, v obecném případě typicky reálné číslo.

2.1.2.2 Klasifikace

Klasifikace spočívá v přiřazení každého vstupu k nějaké třídě. V případě binární klasifikace jde o třídy dvě. Pokud je tříd více mluvíme pak o klasifikaci multinomiální.

2.1.2.3 Shlukování

Shlukovací algoritmy jsou jedním z typických zástupců algoritmů učení bez učitele. Shlukování rozděljuje vstupní data do několika podmnožin tak, aby si data v jedné podmnožině byla podle definované podobnostní funkce dostatečně podobná a zároveň byla od dat v ostatních podmnožinách dostatečně rozdílná. Tato podobnost je pak definována nějakou metrikou nebo mírou podobnosti. Více o problematice shlukování popisují ve své bakalářské práci [14].

2.2 Algoritmy počítačového vidění

Počítačové vidění je jedním z nových a rychle se vyvíjejících oborů s velkým praktickým využitím v mnoha oblastech. Cílem je, aby byl počítač schopný porozumět tomu, co obrazová data reprezentují, a získat z nich užitečné informace. Použití je opravdu široké: mnoho lidí každý den odemyká svůj mobilní telefon pomocí funkce rozpoznávání obličeje, což je jen důmyslná aplikace metod počítačového vidění a binární klasifikace. Důležitá je například i aplikace v medicíně, kdy jsou pomocí počítačového vidění analyzována nebo zvýrazňována obrazová data z radiologických vyšetření a je tak ušetřen čas lékařů.

2.2.1 Sémantická segmentace

V oboru počítačového vidění je několik úrovní, na kterých mohou počítače analyzovat obrazová data. V případě sémantické segmentace jde o úroveň jednotlivých pixelů, kdy každý pixel obrázku je klasifikován zvlášť. Příklad vstupu a výstupu takové segmentace je možné vidět na obrázku 2.1.

Sémantická segmentace je metodou využívanou například k rozdělení obrazového satelitního snímku povrchu Země do několika kategorií na základě využití dané plochy [15]. Jednou z kategorií mohou být komunikace (silnice, železnice), což jsou také liniové stavby, ovšem v této práci je další požadavek, aby takovéto stavby byly oproti okolí zvýšeny.

K samotné realizace sémantické segmentace je často využíváno především neuronových a konvolučních neuronových sítí, které jsou popsány dále.



Obrázek 2.1: Příklad sémantické segmentace [2]

2.3 Umělé neuronové sítě

Umělé neuronové sítě (ANN – Artificial Neural Networks) jsou statistickým modelem, který je inspirován biologickými neurony a jejich propojením. Základní stavební jednotka neuronové sítě je neuron[3], který na vstupu přijímá vektor reálných čísel a na výstupu vrací jedno reálné číslo.

2.3.1 Perceptron

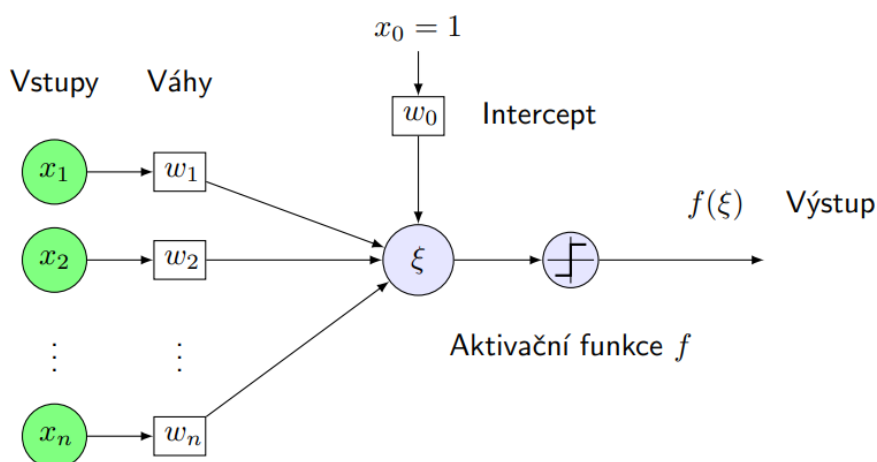
Nejjednodušším modelem umělé neuronové sítě je perceptron představený F. Rosenblattem již roku 1957. Perceptron je možné použít například k binární klasifikaci dat, úspěšně ale umí klasifikovat pouze množiny, které jsou lineárně separovatelné. Tedy při projekci dat do vhodné dvojdimenzionální roviny je schopen perceptron vybírat pouze z rozhodovacích hranic tvořených přímkou.

Schéma perceptronu je znázorněno na obrázku 2.2. Perceptron má na vstupu vektor reálných čísel (x_1, x_2, \dots, x_n) , učení perceptronu pak probíhá pomocí ladění vektoru vah (w_1, w_2, \dots, w_n) a prahu w_0 . Vnitřní potenciál perceptronu označený ξ spočteme vztahem 2.1.

$$\xi = w_0 + \sum_{i=1}^n w_i x_i \quad (2.1)$$

Výstupem perceptronu je pak hodnota funkce $f(\xi)$. Této funkci se říká aktivační funkce a v případě perceptronu je to funkce skoková jejíž předpis je určen vztahem 2.2.

$$f(\xi) = \begin{cases} 1 & \text{když } \xi \geq 0 \\ 0 & \text{když } \xi < 0 \end{cases} \quad (2.2)$$



Obrázek 2.2: jednovrstvý perceptron [3]

2.3.2 Vícevrstvá neuronová síť

Vícevrstvá neuronová síť, běžně označována jako MLP (multilayer perceptron), vzniká propojením několika vrstev neuronů (nebo perceptronů). Propojení vzniká tak, že výstup neuronu z jedné vrstvy slouží jako vstup do jednoho nebo více neuronů ve vrstvě jiné (nebo i stejné v případě rekurentních sítí). První vrstvu označujeme jako vstupní, její jedinou úlohou je předat vstupní data do dalších vrstev sítě které označujeme jako skryté. Poslední vrstvu sítě z pohledu toku dat označujeme jako výstupní. Pokud neurony v rámci jedné vrstvy mezi sebou nemají žádné propojení a jsou naopak propojeny se všemi neurony z vrstvy následující, mluvíme o plně propojené dopředné neuronové síti. Příklad takové topologie ukazuje schéma 2.4.

Oproti jednoduchému modelu perceptronu je vícevrstvá neuronová síť schopna klasifikovat množiny i pomocí složitějších rozhodovacích hranic, než je pouze přímka.

Jedním z nejčastěji používaných algoritmů k učení takto zkonstruované neuronové sítě je algoritmus zpětné propagace (back-propagation) [3], který vyžaduje, aby byla neuronová síť jako funkce jejich parametrů, tedy vah, diferencovatelná. Toho docílíme vhodnou volbou aktivačních funkcí.

2.3.2.1 Aktivační funkce pro skryté vrstvy

U neuronů ve skrytých vrstvách se v současné době nejčastěji používá funkce RELU (Rectified Linear Unit) a její další varianty (ELU, LeakyReLU).

- RELU

$$f(\xi) = \begin{cases} \xi & \text{když } \xi \geq 0 \\ 0 & \text{když } \xi < 0 \end{cases} \quad (2.3)$$

- ELU

$$f(\xi) = \begin{cases} \xi & \text{když } \xi \geq 0 \\ \alpha(e^\xi - 1) & \text{když } \xi < 0 \end{cases} \quad (2.4)$$

- LeakyReLU

$$f(\xi) = \begin{cases} \xi & \text{když } \xi \geq 0 \\ \alpha\xi & \text{když } \xi < 0 \end{cases} \quad (2.5)$$

- Hyperbolický tangens

$$f(\xi) = \frac{e^\xi - e^{-\xi}}{e^\xi + e^{-\xi}} \quad (2.6)$$

2.3.2.2 Aktivační funkce pro výstupní vrstvy

Cílem výstupní vrstvy je transformace hodnot z předchozích vrstev na hodnotu, která odpovídá požadovanému výstupu. Konkrétní podobu aktivační funkce volíme podle toho, zda řešíme úlohu regrese, binární klasifikace či multinomiální klasifikace.

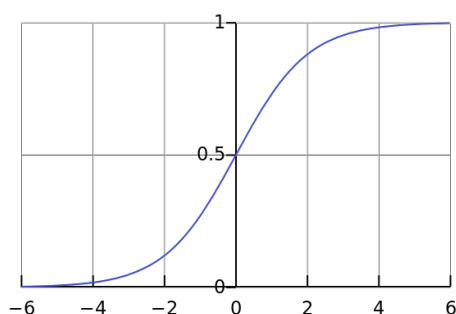
- **Binární klasifikace** – výstupní vrstva obsahuje jeden neuron, jako aktivační funkce je typicky volena logistická funkce (sigmoid) 2.7, viz obrázek 2.3. Obor hodnot takovéto funkce je $\langle 0, 1 \rangle$ a její hodnotu můžeme interpretovat jako pravděpodobnost příslušnosti k třídě 1.

$$f(\xi) = \frac{1}{1 + e^{-\xi}} \quad (2.7)$$

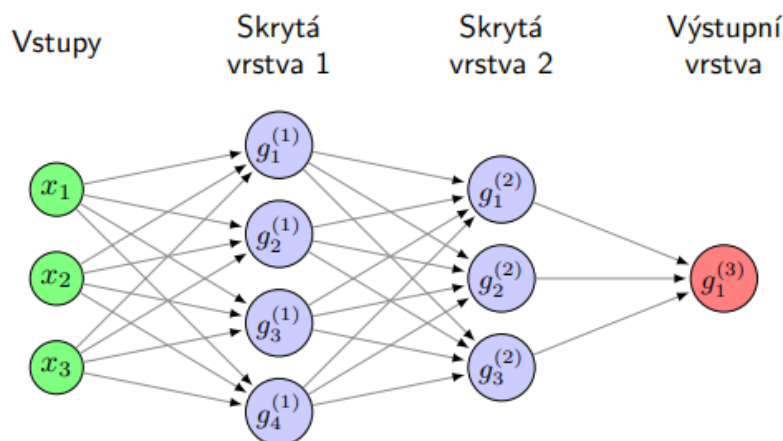
- **Multinomiální klasifikace** – je-li cílem klasifikace do c tříd, je na výstupní vrstvě c neuronů s aktivační funkcí softmax 2.8. Hodnota aktivační funkce i -tého neuronu je pak považována za pravděpodobnost příslušnosti k i -té třídě. Výslednou predikcí je pak třída jejíž odpovídající neuron má nejvyšší hodnotu aktivační funkce.

$$f(\xi)_i = \frac{e^{\xi_i}}{e^{\xi_1} + \dots + e^{\xi_c}} \quad (2.8)$$

- **Regrese** – na výstupu sítě je jeden neuron bez aktivační funkce (neboli s lineární aktivační funkcí $f(\xi) = \xi$), a tedy vrací vnitřní potenciál neuronu.



Obrázek 2.3: Logistická funkce



Obrázek 2.4: Vícevrstvá neuronová síť [3]

2.4 Hluboké učení

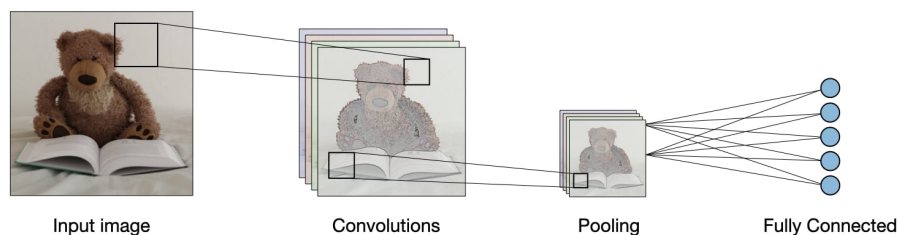
Hluboké učení je pojmenování pro podmnožinu metod strojového učení. Neexistuje ovšem přesný konsenzus o jaké metody jde, někdy se o hlubokém učení mluví jako o neuronových sítích s větším počtem skrytých vrstev, typicky minimálně tři až pět, poté jde o tzv. hlubokou neuronovou síť (DNN – deep neural network).

Jindy je jako hluboké učení označováno širší spektrum metod, které pracují s velkým počtem parametrů a velkými objemy dat.

2.5 Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN, ConvNet – convolutional neural network) jsou jedním ze speciálních případů hlubokých neuronových sítí. Jsou používány

2.6. Architektury konvolučních sítí vhodných pro sémantickou segmentaci



Obrázek 2.5: Konvoluční neuronová síť [4].

především při analýze a zpracování obrazových dat v rastrovém formátu, případně i jiných dat, která je možné do rastrové mřížky uspořádat.

Základem konvolučních neuronových sítí jsou konvoluční vrstvy [4]. Kromě nich pak mezi konvolučními vrstvami CNN obvykle obsahuje pooling vrstvy a před výstupní vrstvou i několik plně propojených vrstev. Základní struktura CNN je na obrázku 2.5.

2.5.1 Konvoluční vrstva

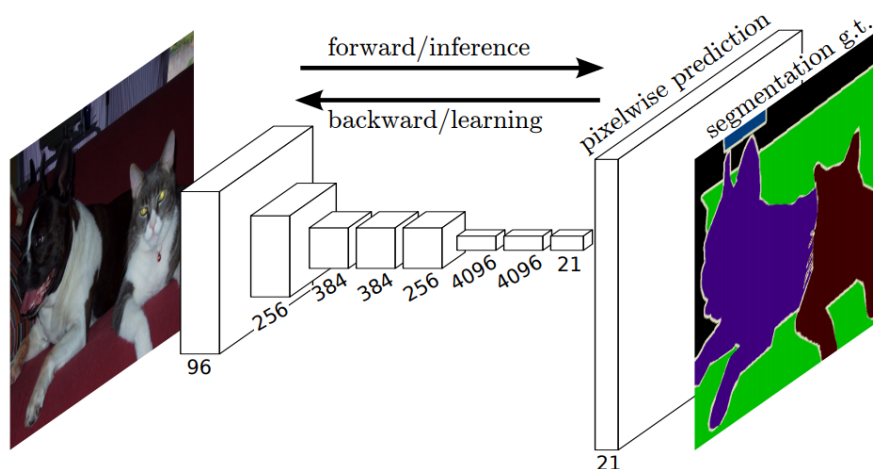
Konvoluční vrstva spočívá v konvoluční transformaci všech prvků ze vstupu společně s jejich logickým okolím pomocí konvolučních jader. Velikost konvolučního jádra a jejich počet jsou hyperparametry CNN, samotné váhy v konvolučním jádru jsou trénovatelným parametrem neuronové sítě. Výstupy konvoluční vrstvy jsou označovány jako příznakové mapy a mají za cíl ve vstupních datech zvýraznit důležité příznaky.

2.5.2 Pooling vrstva

Pooling vrstva následuje typicky po vrstvě konvoluční a slouží ke snížení počtu příznaků tak, že redukuje dimenzi příznakových map. Používají se převážně dvě varianty: Max Pooling a Average Pooling. Princip vrstvy je takový, že oknem vhodné velikosti jsou postupně překrývány příznakové mapy a z oblasti tohoto okna je vybrána buď maximální hodnota v případě Max Poolingu, nebo je do výstupního rastru vložena průměrná hodnota. Pooling vrstvy tak neobsahují žádné trénovatelné parametry, pouze hyperparametry.

2.6 Architektury konvolučních sítí vhodných pro sémantickou segmentaci

Pro sémantickou segmentaci je dle [5] jedním z nejvhodnějších přístupů použití plně konvolučních neuronových sítí (FCN – fully convolutional networks). Plně konvoluční neuronové sítě obsahují oproti standardním CNN navíc upsampling vrstvy [5], které provádí operaci opačnou než pooling vrstvy, tedy zvětšují



Obrázek 2.6: Schéma plně konvoluční neuronové sítě [5].

rozlišení příznakových map. Tímto postupem je tak možné získat výstup požadované velikosti, viz obrázek 2.6.

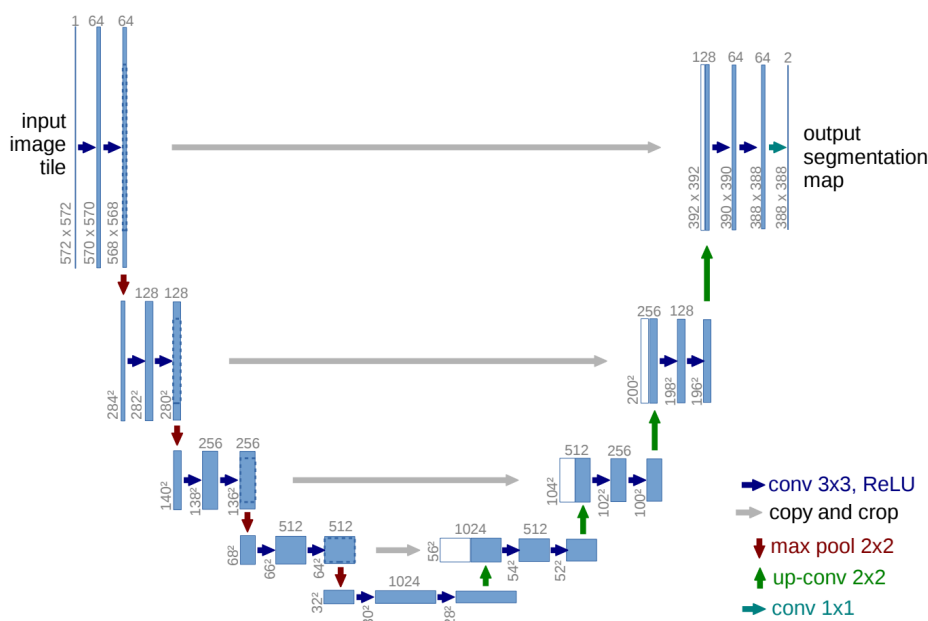
Dále je představeno několik nejzákladnějších topologií konvoluční neuronových sítí, které se v praxi ukázaly jako vhodné pro úlohu sémantické segmentace.

2.6.1 U-Net

U-net, představena autory Olaf Ronneberger, Philipp Fischer a Thomas Brox v roce 2015 [6], je plně konvoluční neuronová síť, která byla původně navržena za účelem segmentace biomedicínských snímků. V původním článku měli autoři k dispozici trénovací množinu o velikosti pouze 35 obrázků, přesto rozšířením vstupní množiny pomocí augmentace dat dosáhli vysoké úspěšnosti. Příklad architektury U-net sítě z [6] je znázorněn na 2.7.

2.6.2 LinkNet

Topologie LinkNet [7] funguje na podobném principu jako U-Net, skládá se taktéž z kodéru a dekodéru. Cílem autorů této architektury bylo zkonstruovat plně konvoluční neuronovou síť pro sémantickou segmentaci, která bude nejen dosahovat dobré přesnosti, ale také bude mít nižší počet parametrů a tím pádem bude rychlejší a efektivnější jak její učení, tak hlavně predikce. Výsledky průchodu obrazových dat touto sítí je poté možné získat velmi rychle a to i na mobilních zařízeních a je možné ji pak používat pro aplikace, kde je výsledek požadován v reálném čase.



Obrázek 2.7: Příklad U-net architektury převzatý z [6].

2.6.3 PSPnet

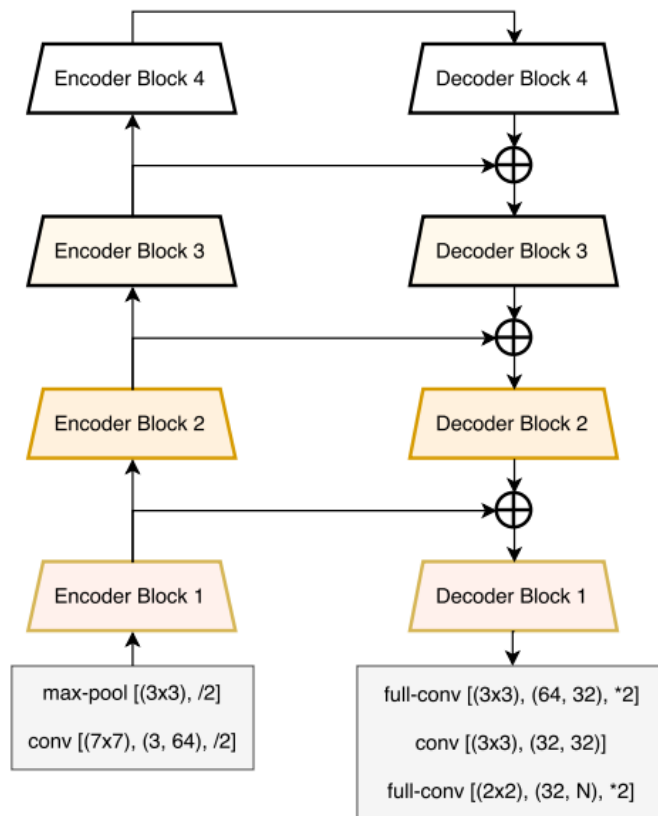
PSPNet – pyramid scene parsing network, je další z konvolučních neuronových sítí určených k sémantické segmentaci. Autoři této architektury si všímají, že správná segmentace nevyžaduje pouze znalost podoby, respektive příznaků jednotlivých objektů, ale také je důležité v jakém kontextu se tyto objekty nacházejí. Uvádějí několik příkladů, k jakým typům chyb může při zanedbání kontextuální informace vést [8].

Pro vyřešení těchto nedostatků přicházejí s Pyramid Pooling Module se schematicou podobou na obrázku 2.9. Jako první se v architektuře nachází kódovací část, která slouží získání základní příznakové mapy o menším rozměru. V [8] autoři jako kódér používají ResNet [16] a vstupní obraz zmenší na osminu šířky a výšky. Takto získaná příznaková mapa je pak vstupem do Pyramid Pooling Module, kde jsou na ni aplikovány konvoluce různých velikostí a zachycují tak různě široký kontext.

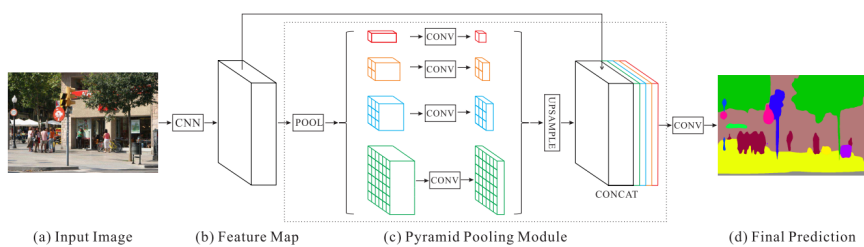
2.7 Vyhodnocení úspěšnosti modelů

Aby bylo možné vyhodnotit jak úspěšný je model, je potřeba definovat metriku pomocí které lze takovou úspěšnost měřit — čím vyšší je hodnota metriky, tím je model úspěšnější. Mezi běžně používané metriky patří např. euklidovská vzdálenost, manhattanská vzdálenost, kosínová podobnost či korelační koefi-

2. ALGORITMY



Obrázek 2.8: Architektura LinkNet [7].



Obrázek 2.9: Architektura PSPnet [8].

Tabulka 2.1: Matice záměn

| Skutečnost | | Předpověď | |
|------------|-----|-----------|----|
| | | ANO | NE |
| | ANO | TP | FN |
| | NE | FP | TN |

cient. Volba té správné metriky závisí na tom jaký mají modelovaná data charakter a je velmi důležitá pro správné vyhodnocení.

2.7.1 Matice záměn

Uvažujme binární klasifikátor, který určuje, zda vstup patří do nějaké třídy nebo ne, výstup je tedy ANO/NE. Nastávají pak čtyři případy, jak klasifikace mohla dopadnout. Tyto možnosti jsou znázorněny v matici záměn 2.7.1. V případě, že klasifikace proběhla správně, je výsledek buď TP (True positive) – klasifikátor predikoval ANO a skutečnost byla stejná, nebo je výsledek TN (True negative) – predikce byla NE a skutečnost také. Pokud byla predikce klasifikátoru nesprávná, pak nastává možnost FN (False negative) – klasifikátor predikoval NE, ale skutečnost byla ANO, nebo případ FP (False positive), kdy je nesprávně predikováno ANO.

Kombinací těchto čtyř možností je možné stanovit metriky kvalit klasifikátoru, tak aby hodnotily ty kvality, které jsou od klasifikátoru pro danou úlohu požadovány. Jako příklad je možné uvést situaci z lékařství, kdy může být požadováno, aby nějaký test s binárním výstupem, který odhaluje závažné onemocnění dělal co nejméně chyb typu false negative (falešně negativní) a naopak nemusí vadit, když udělá několik chyb false positive (falešně pozitivní).

Několik běžně používaných metrik pro binární klasifikaci, které je ale možno rozšířit i pro klasifikaci do více tříd, je představeno dále.

- **Přesnost** (accuracy)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

- **Senzitivita** (sensitivity, recall)

$$sensitivity = \frac{TP}{TP + FN} \quad (2.10)$$

- **Specifita** (specificity)

$$specificity = \frac{TN}{TN + FP} \quad (2.11)$$

2.7.2 Intersection over Union

Intersection over Union (IoU) je název pro metriku hojně využívanou pro hodnocení úspěšnosti sémantické segmentace nebo obecně jakékoliv detekce nejen v obrazových datech. Někde je uváděn také pod jménem Jaccardův index či koeficient.

$$IoU(vzor, predikce) = \frac{|vzor \cap predikce|}{|vzor \cup predikce|} \quad (2.12)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (2.13)$$

Hlavní výhodou této metriky je, že při posuzování úspěšnosti vynechává skutečně negativní datové vzory, které v případě segmentace obrazu nebo detekce stejně z principu nehledáme.

2.8 Účelová funkce

Součástí matematické optimalizace, a tedy i trénování modelu, musí být funkce, pomocí které odhadujeme aktuální chybu modelu. Takové funkci se říká účelová funkce (loss function). Ve strojovém učení je při procesu učení minimalizována hodnota účelové funkce měřené na trénovací (a případně validační) množině. Pro posouzení kvality natrénovaného modelu se pak počítá hodnota účelové funkce na nezávislé testovací množině. Stejně jako u metriky i volba účelové funkce je závislá na charakteru vstupních a výstupních dat a její volba je zásadní pro vytvoření funkčního modelu.

2.9 Morfologické operace

Matematická morfologie je poměrně složitá teorie s velkým praktickým využitím nejen v oblasti počítačového vidění. V kontextu této práce je potřeba pouze několik základních morfologických operací a pro účely použití pro řešený problém nepotřebujeme jejich formální definice, ale vystačíme si pouze s představením praktického fungování.

2.9.1 Dilatace

Jednou ze základních operací matematické morfologie je dilatace. Spočívá v konvoluci s oknem zvolené velikosti, kde je na výstup přiřazena maximální hodnota z uvažovaného okna. Výsledek takovéto operace demonstruje obrázek 2.10. Bílé regiony se rozrostly, operace je tedy vhodná například na vyplnění nechtěných mezer segmentovaných oblastí.



Obrázek 2.10: Vlevo původní stav, vpravo stav po aplikaci dilatace. [9]



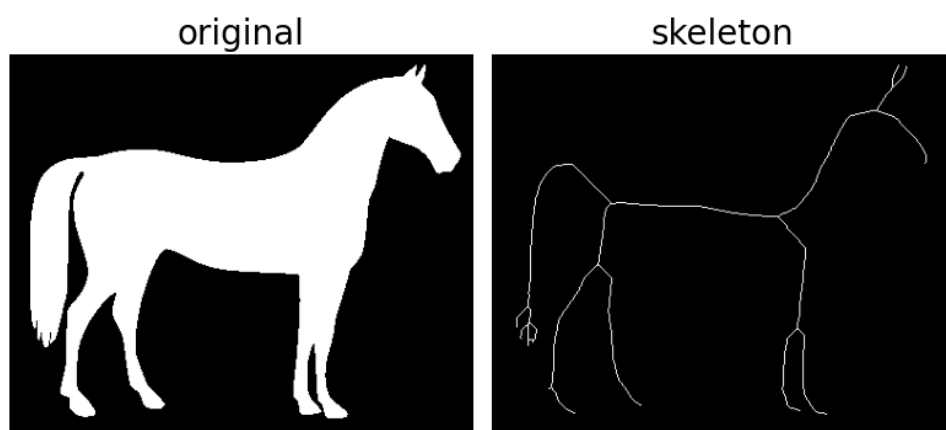
Obrázek 2.11: Vlevo původní stav, vpravo stav po aplikaci eroze. [9]

2.9.2 Eroze

Operace eroze je opakem dilatace, lze ji použít v případě, když se v obrázku chceme zbavit detailů a nebo šumu. Příklad aplikování této operace je zobrazen na obrázku 2.11

2.9.3 Skeletonizace

Cílem skeletonizace je nalézt topologickou kostru objektu, což spočívá v transformaci původního objektu na křivky jednotkové šířky, tak aby původní objekt co nejlépe reprezentovaly. Ukázkou takovéto transformace lze vidět na obrázku 2.12. Skeletonizace může být užitečná pro předzpracování dat při převodu z rastrového do vektorového formátu.



Obrázek 2.12: Vlevo původní stav, vpravo stav po skeletonizaci. [?]

Analýza problému a návrh řešení

Problém řešený v této práci spočívá v identifikaci geografické polohy vyvýšených liniových staveb, jako jsou ochranné hráze, silnice, železnice a další, a to převážně v říčních nivách. Znalost polohy takovýchto struktur může pomoci při modelování povodňových rizik. Pro praktické použití je potřeba identifikované liniové stavby převést do vhodného vektorového GIS formátu.

3.1 Vstupní data

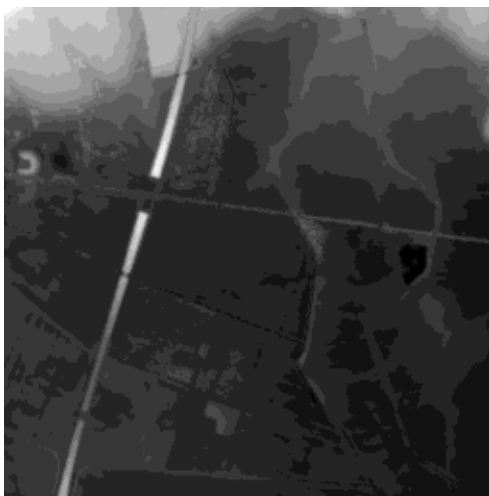
Primárním podkladem pro identifikaci jsou digitální výškové modely (DEM). Jako další data ke zpřesnění identifikace, tam kde informace z DEM nejsou dostatečné, je možné využít satelitní obrazové snímky. Pro účely strojového učení je k dispozici také vektorový popis omezeného počtu liniových staveb tak, jak jsou požadovány jako potencionální výstup.

Data dodaná zadavatelem pochází z několika geografických oblastí, jde o okolí polské řeky Wisla, české řeky Moravy a tří oblastí v Rakousku – okolí řeky Inn v oblasti města Innsbruck, okolí řeky Salzach v oblasti Salzburgu a okolí řeky Dunaj v oblasti města Linz.

3.1.1 Digitální výškové modely

Digitální výškové modely byly poskytnuty zadavatelem práce, jelikož ne pro všechny oblasti jsou tato data volně dostupná v dostatečné kvalitě. Horizontální rozlišení poskytnutých výškových modelů je 10 m a jsou v různých systémech souřadnic podle toho, které území výškově popisují. Formát dodaných souborů je GeoTIFF, což je pro tento typ dat standardní.

Vizualizace digitálního výškového modelu je možná užitím libovolných barevných škál, na obrázku 3.1 je příklad takovéto vizualizace pomocí stupňů šedi.



Obrázek 3.1: Vizualizace dat z digitálního výškového modelu

3.1.2 Satelitní snímky

Satelitní snímky z mise Sentinel-2 poskytované evropskou kosmickou agenturou jsou dostupné z Copernicus Open Access Hub. Použity byly snímky typu Level-2A v nejvyšším dostupném rozlišení, které činí 10 m. V tomto rozlišení je dostupných několik spektrálních pásem. Po ručním prozkoumání snímků a dat se ukázalo, že potencionálně přínosná jsou pouze pásma viditelného spektra (červené, zelené, modré), která tvoří TCI (true color image).

Snímky jsou organizovány v dlaždicích, které pokrývají území různých rozloh a je u nich použit souřadnicový systém UTM WSG84. K uložení těchto snímků je použit formát JPEG 2000, viz sekce 1.4.1.2. Ukázkou části takového snímku je obrázek 3.2.

3.1.3 Ochranné hráze a násypy liniových staveb

Geografická poloha ochranných hrází a násypů liniových staveb je reprezentována jako vektory v GIS formátu Esri Shapefile v různých systémech souřadnic. Tyto vektory byly taktéž dodány zadavatelem a představují výstup, který chceme aby nástroj navrhovaný v této práci generoval.

Příklad části takových vektorů, které polohově odpovídají ukázce satelitního snímku na obrázku 3.2 i ukázce digitálního výškového modelu na obrázku 3.1, lze vidět na obrázku 3.3.

3.2 Možná řešení

V této části bude rozebráno několik možných variant, jak je k řešenému problému možno přistupovat, a jak jsou částečně shodné či alespoň analo-



Obrázek 3.2: RGB satelitní snímek



Obrázek 3.3: Příklad vektorů hrází a náspů liniových staveb

gické problémy řešeny jinými autory. Následně bude představena metoda zvolena pro tuto práci, která byla vyhodnocena jako nejvhodnější vzhledem ke vstupním datům a vzhledem k požadovanému zaměření této práce na metody strojového učení.

3.2.1 Morfometrické charakteristiky

Identifikace ochranných hrází a naspů liniových staveb v digitálních výškových modelech je problém, kterým se zabývá několik prací [17], [18], [19], jejichž společným prvkem je použití různých morfometrických charakteristik. Morfometrie se zabývá číselným vyvozováním charakteristiky tvarů zemského povrchu [20].

V [17] je pro identifikaci využíváno 5 morfometrických charakteristik – relativní výška, sklon, směr sklonu (aspect), profilová křivost (profile curvature) a planární křivost (planar curvature). Jejich detailní popis je uveden v [17]. Takto získané příznaky mohou poté sloužit k identifikaci buď přímo, využitím prahových hodnot pro jednotlivé příznaky, a nebo jako vstup do klasifikátoru.

Ve všech pracích mají autoři k dispozici digitální výškové modely lepšího horizontálního rozlišení, než je 10 m jako v této práci.

3.2.2 Sémantická segmentace

Na identifikaci polohy ochranných hrází je možno nahlížet jako na binární klasifikaci. Pro každý pixel jak digitálního výškového modelu, tak případně i satelitního snímku, provedeme klasifikaci, zda jde o hráz či nikoliv. Výstup takového klasifikátoru je ekvivalentní s výstupem procesu sémantické segmentace obrazu.

Jak již bylo zmíněno, konvoluční neuronové sítě jsou vhodným nástrojem pro sémantickou segmentaci. Možnost použití tohoto přístupu pro identifikaci ochranných hrází je ověřena v [21]. Autoři této práce použili jako vstupní data digitální výškový model s horizontálním rozlišením 5 m, vektory silnic a železnic a vektory říčních sítí. Jako model byla použita konvoluční neuronová síť typu U-net, úspěšnost měřená metrikou IoU 2.7.2 je vyčíslena na 0.73 na trénovacích datech a 0.48 na datech validačních. Data použitá k trénování a k validaci pocházejí dle autorů ze stejného geografického regionu a to ze státu Florida, USA. Pro vytvoření finální predikce poloh ochranných hrází je na výstup U-net aplikováno ještě několik úprav, které jsou detailně popsány v [21].

3.3 Zvolené řešení

Vzhledem k povaze vstupních dat, tedy digitálního výškového modelu, který má menší rozlišení, než mají k dispozici autoři prací, využívající morfometrické charakteristiky a také s ohledem na další vstup, u kterého je potřeba ověřit,

zda může nedostatečnou kvalitu DEM vykompenzovat – obrazové satelitní snímky, a v neposlední řadě s ohledem na apriorní cíl, ověřit vhodnost použití moderních metod strojového učení, je v této práci k implementaci a otestování zvolena metoda sémantické segmentace.

K sémantické segmentaci je využito konvolučních neuronových sítí. Je otestována úspěšnost několika nejpoužívanějších sítí a to U-net, LinkNet a PSP-Net s, na jiných úlohách úspěšně použitými, architektury kódovací části těchto sítí. Dále je vybrána jedna z testovaných konvolučních neuronových sítí a je empiricky optimalizována její architektura.

Důležitou součástí zvolené metody je následné zpracování (post-processing) hrubého výstupu modelu. Jednotlivé kroky jsou popsány a rozebrány na konkrétních příkladech v kapitole zabývající se implementací. Proces závěrečné vektorizace identifikovaných hrází je taktéž prezentován v následující kapitole.

Implementace

K implementaci bylo využito jazyka Python ve verzi 3 a interaktivního webového vývojového prostředí Jupyter notebook. Python je v době psaní této práce bezkonkurenčně nejpoužívanějším jazykem používaným pro implementaci prototypů modelů a algoritmů strojového učení. Mezi jeho přednosti patří jednoduchost a rychlost s jakou je možné programy implementovat a především velké množství různých knihoven. Nevýhodou je pak v určitých případech nižší výkonnost při výpočetně náročných úlohách nebo při práci s velkým množstvím dat.

4.1 Použité knihovny

Implementovat algoritmy strojového učení od základu v samotném jazyku Python by bylo časově velmi náročné a neefektivní, proto je použito několik užitečných knihoven. Výčet a stručný popis několika, z hlediska implementované úlohy nejdůležitějších, je uveden níže.

- **Numpy** – Přidává podporu n-rozměrných polí a matic velkého rozsahu. Umožňuje na těchto datových strukturách efektivně provádět matematické operace.
- **Cv2** – Python interface pro OpenCV. OpenCV (Open Source Computer Vision Library) je opensource knihovna, která slouží k manipulaci s obrazovými daty a poskytuje algoritmy a metody z oblasti počítačového vidění a obecně strojového učení.
- **Gdal** – Gdal je knihovna určená na manipulaci s rastrovými i vektorovými GIS daty. Gdal je knihovna v jazyce C++ je ale poskytováno API pro Python.
- **Tensorflow** - Opensource framework strojového učení, poskytnutý společností Google.



Obrázek 4.1: Vlevo vektory ochranných hrází, vpravo po rasterizaci. Jako pozadí je použit digitální výškový model.

- **Keras** – API pro hluboké učení, používá Tensorflow.
- **Segmentation models** – knihovna, poskytující API k konvolučním neuronovým sítím pro sémantickou segmentaci, využívá Keras a Tensorflow [22].

4.2 Předzpracování vstupních dat

Předtím, než je možné přistoupit k trénování konvolučních neuronových sítí, je potřeba data transformovat do podoby, která je vhodná pro vstup do tohoto typu modelu.

4.2.1 Rasterizace vektorů ochranných hrází

Ochranné hráze jsou k dispozici ve vektorovém formátu, který není pro CNN vhodný, proto je potřeba tato data převést na rastr. K rasterizaci je využito metody `RasterizeLayer` z knihovny `Gdal`. Rasterizace je provedena do stejného rozlišení a stejného souřadnicového systému, který mají odpovídající digitální výškové modely, každá ochranná hráz má tedy pro rasterizaci šířku 1 px. Ukázkou podoby ochranných hrází ve vektorovém formátu a po rasterizaci lze vidět na obrázku 4.1.

4.2.2 Sloučení satelitních snímků

Jelikož jsou satelitní snímky poskytovány v dlaždicích různých rozměrů, a k překrytí území, které odpovídá poskytnutým výškovým modelům je potřeba dlaždic několik, je nezbytné tyto dlaždice sloučit tak, aby byl zachován souřadnicový systém. K tomu je použita utilita `gdal_merge.py`, která je součástí knihovny `Gdal`.

Při stahování satelitních snímků z Copernicus Open Access Hub je podstatné, aby byly všechny použité snímky pořízeny ideálně v co nejkratším

časovém odstupu. Jejich barevná struktura totiž může být závislá na ročním období, například v zimním období může být v určitých oblastech přítomen sníh.

4.2.3 Převod na stejný SRS

Pro snadnější práci se všemi vstupními daty je žádoucí, aby byla všechna projektována stejným souřadnicovým systémem a ve stejném rozlišení. Digitální výškový model a rasterizované hráze již ve stejném souřadnicovém systému a rozlišení jsou po prvním kroku předzpracování. Satelitní snímky již mají také rozlišení 10 m, do stejného souřadnicového systému jsou převedeny metodou `Warp` z knihovny `Gdal`.

4.2.4 Prozkoumání předzpracovaných vstupních dat

Digitální výškové modely, satelitní snímky i rasterizované hráze jsou prozkoumány, aby bylo zjištěno zda si pixely se stejnou polohou v rámci obrazových dat odpovídají i geografickou polohou a mají tedy vzájemně si odpovídající obsah. Zkoumáno je především to, zda se liniová stavba (např. hráz, silnice i železnice) v rasterizovaných datech skutečně nachází tam, kde tomu nasvědčují hodnoty na odpovídajícím pixelu v digitálním výškovém modelu. Stejným způsobem jsou prověřeny i satelitní snímky. Zjištěno je, že si data polohově odpovídají a tak není v tomto směru potřeba již dalších transformací.

Dalším krokem je ruční prozkoumání anotovaných hrází, získání přehledu o tom, kde se obvykle nacházejí a jaké problémy mohou při identifikaci jejich polohy nastat. Bylo zjištěno, že struktury typu ochranných hrází nejsou označeny v celém rozsahu dodaného digitálního výškového modelu. Označeny jsou pouze ty, které jsou z hlediska modelování povodňových rizik zajímavé, typicky tedy takové, které jsou v blízkosti řek. Situace, kdy podoba výškového modelu i satelitních dat, naznačuje, že se na daném místě nachází hrázová struktura, ale v anotovaných datech takové struktura není, je ilustrován příkladem na obrázku 4.2.

Silnice, která je od řeky vlevo nahoře, je ve anotovaných vektorových datech označena jako hráz. V pravém dolním rohu ovšem silnice se stejnou podobou, jak v digitálním výškovém modelu, tak v satelitním snímku, jako hráz označena není. Data s těmito nedostatky mohou být pro trénovaný model matoucí, proto je potřeba je vyfiltrovat a jako vstup do modelu je nepoužívat.

Zajištěno je to tak, že se jako vstup do modelu používají čtvercové vzorky dat o velikosti 64 px a je požadováno aby alespoň 50 px obsahovalo hráz. Čím je zvolený výřez menší, tím je menší šance, že se v něm budou nacházet hrázové struktury, které tak ale nejsou označeny. Velikost 64 pixelů byla zvolena jako kompromis mezi množstvím obsažených informací a šancí, že některá data nejsou správně označena jako hráz. Jelikož jeden pixel je ve skutečnosti deset



Obrázek 4.2: Příklad pravděpodobně nesprávně anotovaných dat

metrů, mělo by pro určení zda je daný pixel hráz ve většině případů stačit jen jeho bezprostřední okolí.

4.3 Přehled datových sad

Pro snadnější odkazování ve zbytku textu je níže zavedeno označení pro datové sady, které jsou použity.

- `austria_01` – 4835x10780 px, okolí řeky Inn
- `austria_02` – 9177x9589 px, okolí řeky Salzach
- `austria_03` – 4263x12977 px, okolí řeky Dunaj
- `czech` – 7426x2501 px, okolí řeky Morava
- `poland` – 28497x22246 px, okolí řeky Wisla

4.3.1 Augmentace dat

Dostupných dat, převážně pak anotovaných vyvýšených liniových struktur, ale i digitálních výškových modelů, je omezené množství, které by nebylo pro natrénování modelu dostačující. Vyřešit tento nedostatek lze augmentací dat.

Mezi základní augmentační přístupy patří otočení a vodorovné či svislé překlopení. Dále je možné přidání šumu, odebrání informací nebo deformace. Takto provedenými transformacemi získáme ze vstupních dat data další, z kterých může trénovaný model vytěžit informace, které se mu z původních dat naučit nepovedlo.

4.4 Trénování CNN pro sémantickou segmentaci

Otestovány byly tři architektury konvolučních neuronových sítí vhodných pro sémantickou segmentaci a to U-net, LinkNet a PSPNet. Pro sestupnou (kódovací) část těchto sítí je využita topologie VGG16 [23]. Otestovány byly i další architektury kódovacích částí: ResNet, DenseNet, EfficientNet. Prezentovány budou v této práci ale pouze výsledky VGG16, která měla úspěšnost nejvyšší. Dále je navíc navržena vlastní upravená architektura CNN.

Pro testování CNN s běžně používanými topologiemi kódovacích částí je využita knihovna Segmentation Models [22].

Natrénovány jsou dva typy modelů, které se liší vstupními daty. První typ je natrénován pouze na základě digitálního výškového modelu. K trénování druhého typu je použit digitální výškový model a zároveň satelitní obrazové snímky. Trénovat model pouze s využitím satelitních snímků nedává smysl, jelikož satelitní snímky samy o sobě neobsahují dostatek informací, které by vedly k identifikaci polohy hrázových struktur.

Jako hlavní kritérium pro hodnocení úspěšnosti modelu je použita metrika Intersection over Union. Jako pomocná měřítka úspěšnosti jsou pak reportovány specifická a senzitivita.

4.4.1 Parametry modelů typu CNN

K trénování konvolučních neuronových sítí je potřeba nastavit několik parametrů v závislosti na jejich architektuře a na podobě vstupních dat. Níže je uveden přehled těch zásadních a je popsáno jakým způsobem byly nastaveny.

- **Metrika** – Slouží k hodnocení úspěšnosti modelu. Zvolena byla metrika IoU.
- **Účelová funkce (loss function)** – Cílem trénování modelu je minimalizace hodnoty účelové funkce. Účelová funkce je zde volena ve tvaru $1 - \text{IoU}$. Kde IoU musí být modifikována tak, aby měla vždy nenulový gradient.

- **Optimalizátor (optimizer)** – Optimalizátor je algoritmus pomocí kterého je optimalizována účelová funkce tak, aby mohly být následně aktualizovány váhy neuronové sítě. V této práci byl zvolen *Nadam*.
- **Velikost dávky (batch size)** – Velikost dávky určuje po kolika vstupech dojde k aktualizaci vah. Jako výchozí hodnota, na které při ladění tohoto parametru začít, se doporučuje 32. Z důvodů výpočetní optimalizace se doporučuje volit velikost jako mocninu dvou. Při trénování modelů byla v této práci použita velikost 16.
- **Míra učení (learning rate)** – Určuje jak velké jsou kroky při hledání minima účelové funkce. Hodnotu je třeba volit také v závislosti na velikosti dávky, pokud je velikost dávky malá, je potřeba volit i nižší míru učení. Při trénování modelů v této práci se ukázaly jako vhodné hodnoty v řádu 0.001–0.0001.
- **Počet epoch** – Jedna epocha odpovídá jednomu použití všech vstupních dat. Počet epoch po které se provádí trénování závisí na tom, zda se úspěšnost modelu zlepšuje na trénovací množině a zároveň nedochází k přeučení, tedy úspěšnost roste i na validačních datech.

4.4.2 Model založený na digitálním výškovém modelu

Vstupem je jednokanálový rastr velikosti 64x64 pixelů, data jsou generována tím způsobem, že je z celého digitálního výškového modelu vybrán výřez této velikosti s náhodným umístěním a poté je provedena augmentace. Aby byl takovýto výřez použit na trénování modelu, je potřeba aby obsahoval alespoň 50px anotovaných hrází.

Výškový digitální model obsahuje informace o absolutních hodnotách nadmořských výšek. Podstatná pro identifikaci hrázových struktur je ovšem relativní výška daného bodu vůči bodům okolním, proto je provedena normalizace digitálního výškového modelu. Pro každý vstupní rastr obsahující informace o nadmořských výškách je zjištěna hodnota minimální výšky a ta je odečtena od všech ostatních.

Jednotlivé architektury modelu jsou natrénovány a vyhodnoceny na různých datech. Výsledky jsou uvedeny v tabulkách 4.1, a 4.2. Vzhledem k tomu, že je v další části navržena vlastní upravená topologie CNN, která je používána pro finální predikci, jsou v této části ponechány výsledky bez dalších komentářů a je pouze konstatováno, že nejlepší úspěšnosti dosahuje síť typu U-net.

4.4.3 Model založený na digitálním výškovém modelu a satelitních snímcích

Další otestovaný model má jako vstup kromě digitálního výškového modelu, který je předzpracován stejným způsobem jako u 4.4.2, také obrazové satelitní snímky. Celý vstup je potom čtyřkanálový rastr o rozměrech 64x64 pixelů, se

4.4. Trénování CNN pro sémantickou segmentaci

Tabulka 4.1: Model s použitím DEM jako vstupu, natrénován na datech `austria_01`, `austria_02` a otestován na `austria_03`.

| | IoU | | Specifcita | | Senzitivita | |
|------------------------|-------|------|------------|------|-------------|------|
| Model | Train | Test | Train | Test | Train | Test |
| U-net (VGG16) | 0.55 | 0.41 | 0.98 | 0.97 | 0.55 | 0.41 |
| LinkNet (VGG16) | 0.63 | 0.38 | 0.97 | 0.97 | 0.47 | 0.38 |
| PSPNet (VGG16) | 0.15 | 0.11 | 0.95 | 0.95 | 0.3 | 0.15 |

Tabulka 4.2: Model s použitím DEM jako vstupu, natrénován na datech `poland` a otestován na `czech`.

| | IoU | | Specifcita | | Senzitivita | |
|------------------------|-------|------|------------|------|-------------|------|
| Model | Train | Test | Train | Test | Train | Test |
| U-net (VGG16) | 0.66 | 0.53 | 0.99 | 0.98 | 0.66 | 0.53 |
| LinkNet (VGG16) | 0.63 | 0.45 | 0.98 | 0.98 | 0.65 | 0.41 |
| PSPNet (VGG16) | 0.5 | 0.36 | 0.97 | 0.97 | 0.64 | 0.3 |

Tabulka 4.3: Model s použitím DEM a satelitních snímků Sentinel-2, natrénován na datech `austria_01`, `austria_02` a otestován na `austria_03`.

| | IoU | | Specifcita | | Senzitivita | |
|------------------------|-------|------|------------|------|-------------|------|
| Model | Train | Test | Train | Test | Train | Test |
| U-net (VGG16) | 0.53 | 0.39 | 0.98 | 0.98 | 0.52 | 0.4 |
| LinkNet (VGG16) | 0.50 | 0.38 | 0.97 | 0.97 | 0.45 | 0.37 |
| PSPNet (VGG16) | 0.13 | 0.09 | 0.95 | 0.9 | 0.28 | 0.14 |

satelitními snímky musí být samozřejmě provedeny shodné augmentační kroky jako s DEM.

Výčet úspěšnosti jednotlivých architektur CNN na různých datech je v tabulkách 4.3 a 4.4.

Tabulka 4.4: Model s použitím DEM a satelitních snímků Sentinel-2, natrénován na datech `poland` a otestován na `czech`

| | IoU | | Specifcita | | Senzitivita | |
|------------------------|-------|------|------------|------|-------------|------|
| Model | Train | Test | Train | Test | Train | Test |
| U-net (VGG16) | 0.61 | 0.5 | 0.99 | 0.99 | 0.62 | 0.51 |
| LinkNet (VGG16) | 0.58 | 0.44 | 0.98 | 0.98 | 0.62 | 0.39 |
| PSPNet (VGG16) | 0.41 | 0.32 | 0.96 | 0.96 | 0.6 | 0.28 |

4.5 Návrh vlastní CNN typu U-net

V této části je otestován vlastní návrh CNN architektury typu U-net. Tento typ architektury byl vybrán, protože se ukázal jako potenciálně nejlepší v předchozí části a zároveň je často a úspěšně využíván v pracích jiných autorů. Opět jsou natrénovány dva modely lišící se typem vstupních dat.

Konkrétní strukturu této konvoluční neuronové sítě je možné vyčíst z výpisu 4.1. Její kódovací část se skládá z devíti konvolučních vrstev s různým počtem konvolučních filtrů. Výpis 4.1 popisuje síť, která byla použita k vytvoření modelu s využitím pouze DEM. Architektura sítě, která kromě DEM pracuje i se satelitními snímky se liší pouze v počtu konvolučních filtrů na jednotlivých vrstvách.

4.5.1 Model založený na DEM

Vstupní data jsou stejná a stejně předzpracovaná jako pro model testovaný v předchozí části 4.4.2.

Trénování modelu je prováděno tak dlouho, dokud roste úspěšnost na validačních datech měřená metrikou IoU, tolerance je 10 po sobě jdoucích epoch v kterých se úspěšnost nemusí zvýšit. Průběh takového trénování na prvních 100 epochách je znázorněn grafem 4.3. Osa x toho grafu obsahuje pořadí epochy na ose y je pak hodnota IoU. Modrou křivkou je znázorněna úspěšnost na trénovacích datech, oranžovou pak na datech validačních.

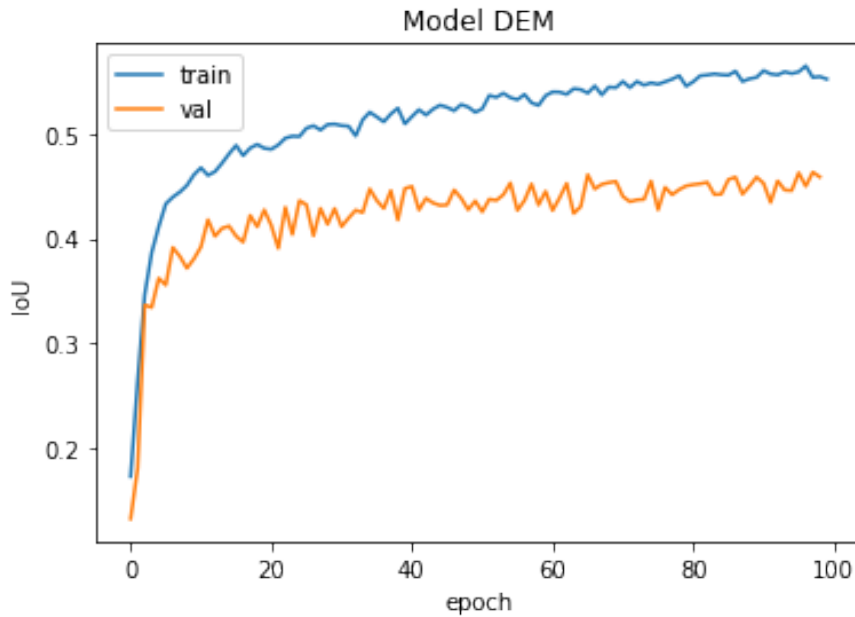
4.5.1.1 Zhodnocení a interpretace výsledků

Souhrn výsledků trénování a testování modelu na různých datech je uveden v tabulce 4.5. Z výsledku lze vyčíst, že největší úspěšnost při testování na datech `czech` má model natrénovaný na `poland` a na datech `poland` má nejlepší výsledek model z dat `czech`.

Pro data `austria_03` má pak nejlepší úspěšnost model, pro jehož trénování jsou použita data `austria_01` a `austria_02`. Přidání dat `czech` nebo `poland` do trénovací množiny naopak úspěšnost zhorší. Toto je možné vysvětlit tím, že profil terénu je v datech z Rakouska oproti Polsku a Česku odlišný, data pocházejí z podhorských oblastí a hrázové struktury zde mohou mít odlišnou podobu na digitálním satelitním snímku. Otázkou je také, jaká je kvalita anotovaných dat. Je možné, že kvalita anotací může být u Rakouska mírně zhoršená, což ale není možné jednoduše posoudit.

Vzhledem k reálnému využití modelu může z těchto výsledků plynout poznatek, že je vhodné mít několik modelů, každý pro různý typ terénu. Toto tvrzení je ovšem potřeba potvrdit na větším množství dat.

Ukázka úspěšné predikce modelu založeného pouze na DEM, kde je z podoby DEM zřetelně vidět, kde se hráz nachází, je na obrázku 4.4. Příklad



Obrázek 4.3: Průběh trénování modelu založeném na DEM. Trénováno na datech `austria_01`, `austria_02` a otestováno na `austria_03`.

Tabulka 4.5: Úspěšnost modelu založeném na DEM, natrénovaném a otestovaném na různých datech.

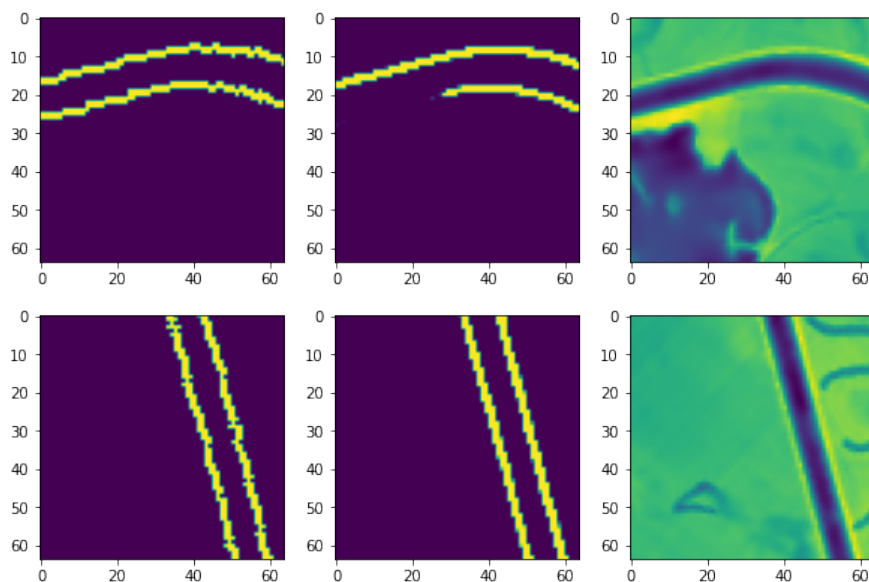
| | | IoU | |
|---|-------------------------|-------|------|
| Trénovací data | Testovací data | Train | Test |
| <code>austria_01</code> , <code>austria_02</code> | <code>austria_03</code> | 0.62 | 0.47 |
| <code>czech</code> , <code>poland</code> | <code>austria_03</code> | 0.67 | 0.33 |
| <code>austria_01</code> , <code>austria_02</code> , <code>czech</code> , <code>poland</code> | <code>austria_03</code> | 0.62 | 0.38 |
| <code>austria_01</code> , <code>austria_02</code> , <code>austria_03</code> | <code>czech</code> | 0.51 | 0.49 |
| <code>austria_01</code> , <code>austria_02</code> , <code>austria_03</code> , <code>poland</code> | <code>czech</code> | 0.48 | 0.54 |
| <code>poland</code> | <code>czech</code> | 0.68 | 0.59 |
| <code>czech</code> | <code>poland</code> | 0.68 | 0.63 |
| <code>austria_01</code> , <code>austria_02</code> , <code>austria_03</code> | <code>poland</code> | 0.51 | 0.53 |
| <code>austria_01</code> , <code>austria_02</code> , <code>austria_03</code> , <code>czech</code> | <code>poland</code> | 0.51 | 0.57 |

predikce, kde z DEM není jasné jaká je poloha hráze a model ji tak neumí predikovat je znázorněn na obrázku 4.5.

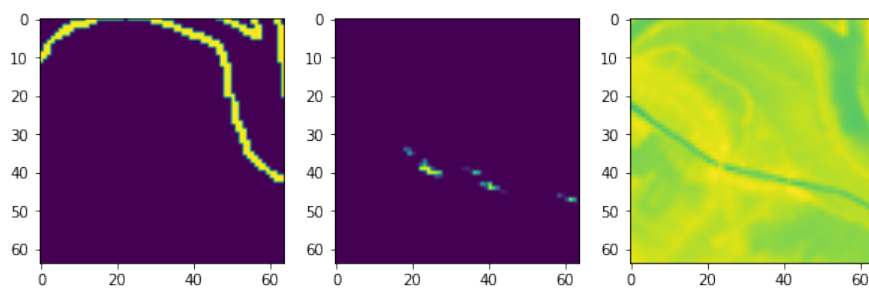
4.5.2 Model založený na DEM a satelitních snímcích

Předzpracování a podoba vstupních dat je stejná jako u 4.4.3. Architektura konvoluční neuronové sítě je shodná jako pro model založený jen na DEM a

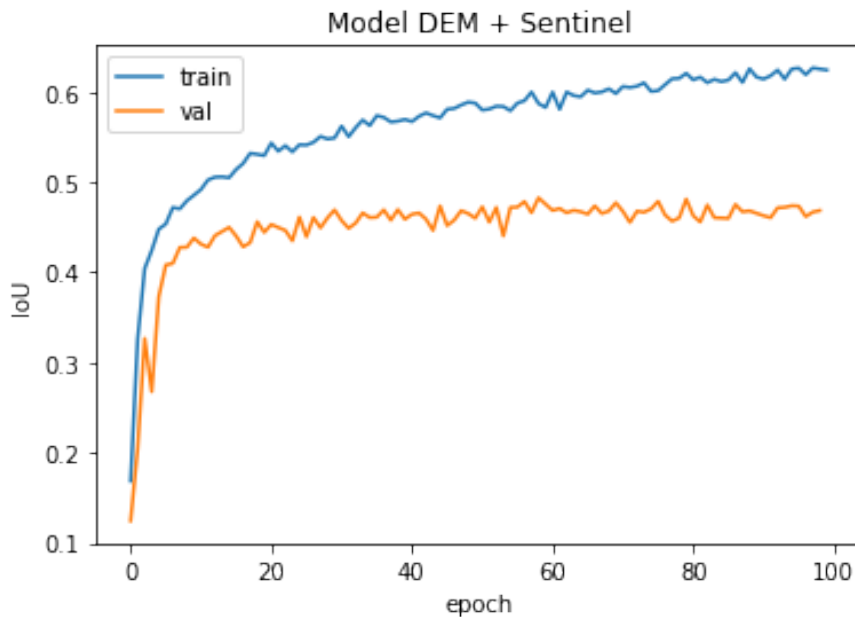
4. IMPLEMENTACE



Obrázek 4.4: Úspěšná predikce modelu založeném na DEM. Vlevo anotované hráze, uprostřed predikce, vpravo DEM.



Obrázek 4.5: Neúspěšná predikce modelu založeném na DEM. Vlevo anotované hráze, uprostřed predikce, vpravo DEM.



Obrázek 4.6: Průběh trénování modelu založeném na DEM a satelitních snímcích. Trénováno na datech `austria_01`, `austria_02` a otestováno na `austria_03`.

je uvedena v ukázce 4.1, obsahuje ale na každé konvoluční vrstvě čtyřnásobný počet konvolučních filtrů, větší je tedy na každé konvoluční vrstvě i počet příznakových masek.

Metodika pro trénování modelu je stejná jako u 4.5.1 – trénování je ukončeno pokud se více jak 10 epoch v řadě nezlepší maximální úspěšnost na validačních datech. Průběh takového trénování je znázorněn grafem 4.6.

4.5.2.1 Zhodnocení a interpretace výsledků

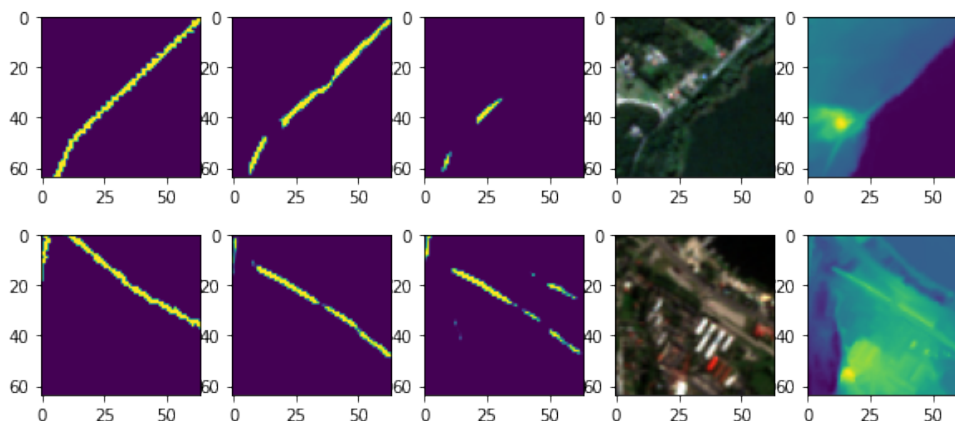
Výsledky trénování a testování modelu založeném na digitálních výškových modelech a satelitních snímcích uvedené v tabulce 4.6 ukazují, že se úspěšnost modelu na testovacích datech ve všech případech mírně zhoršila nebo zůstala přibližně stejná. Po prozkoumání konkrétních predikcí, lze najít takové, kde satelitní snímky mají přínos pro zlepšení identifikace hrázové struktury. Příklady lze vidět na obrázku 4.7. Existují ovšem i takové případy, kde je model s přidáním satelitních snímků horší, než model využívající pouze DEM. Může to být způsobeno protichůdnými informacemi mezi DEM a satelitním snímkem.

Jelikož je metrika IoU měřící úspěšnost modelu v tomto případě horší než pro model DEM, není proto možné jednoznačně prohlásit, že mají satelitní snímky přínos pro rozpoznávání hrází. Úspěšnost ale není výrazně horší a lze objevit i příklady kde jsou satelitní snímky přínosné, není tedy na místě

4. IMPLEMENTACE

Tabulka 4.6: Úspěšnost modelu založeném na DEM a satelitních snímcích, natrénovaném a otestovaném na různých datech.

| | | IoU | |
|--|----------------|-------|------|
| Trénovací data | Testovací data | Train | Test |
| austria_01, austria_02 | austria_03 | 0.58 | 0.45 |
| czech, poland | austria_03 | 0.65 | 0.33 |
| austria_01, austria_02, czech, poland | austria_03 | 0.66 | 0.38 |
| austria_01, austria_02, austria_03 | czech | 0.5 | 0.45 |
| austria_01, austria_02, austria_03, poland | czech | 0.51 | 0.49 |
| poland | czech | 0.67 | 0.57 |
| czech | poland | 0.68 | 0.63 |
| austria_01, austria_02, austria_03 | poland | 0.5 | 0.53 |
| austria_01, austria_02, austria_03, czech | poland | 0.58 | 0.55 |



Obrázek 4.7: Ukázka predikcí, kde jsou satelitních snímky přínosné. Zleva – anotovaná data, predikce využívající satelitní snímky, predikce využívající pouze DEM, satelitní snímek, DEM.

ani definitivně vyloučit přínos satelitních snímků. Je potřeba otestovat model na více odlišných datech, případně navrhnout metodu, která bude jak model založený na DEM, tak model založený na DEM a satelitních snímcích, nějakým způsobem kombinovat.

4.6 Predikce

Modely v předchozích částech jsou natrénovány na datech velikosti 64x64px a jsou proto také schopné provést jednotlivé predikce pouze pro data těchto rozměrů. Pro reálné použití je potřeba jednotlivé predikce sloučit, tak aby byla predikována celá požadovaná oblast. Dosaženo toho je posouváním okna



Obrázek 4.8: Vlevo po ukázka predikce, kde každý pixel nese hodnotu pravděpodobnosti, že se na jeho poloze nachází hráz. Vpravo ukázka predikce po binarizaci.

velikosti 64x64 px přes celé vstupní data s horizontálním i vertikálním krokem 4 px.

Pokud model predikuje pixel jako hráz, je k výsledné hodnotě pixelu přičtena hodnota 1. Mimo krajních 64 px je každý pixel predikován celkem 256 krát. Po provedení celé predikce jsou výsledné hodnoty vyděleny počtem modelů, které se tento pixel pokusily predikovat, tím je hodnota normalizována do intervalu $< 0, 1 >$. Takto upravené hodnoty pixelů můžeme považovat za pravděpodobnost, že je daný pixel hráz. Na levém obrázku 4.8 je toto znázorněno, černá barva znamená hodnotu 1, tedy tady predikoval model s velkou jistotou hráz. Bílá barva znázorňuje hodnotu 0, ostatní hodnoty mají barvy stupně šedi lineárně rozložené v daném intervalu. Výsledná predikce je získána tak, že za hráz jsou prohlášeny ty pixely, které mají pravděpodobnost alespoň 0.5, výsledek takového prahování je na pravém obrázku 4.8.

4.7 Post-processing

Důležitou součástí identifikace polohy ochranných hrází a jejich převodu do vektorového formátu je následné zpracování výstupu z konvolučních neuronových sítí. Výstup z CNN může být v mnoha případech nepřesný. Jelikož je ale známa povaha výsledných dat, je možné navrhnout několik úprav, kterými lze dosáhnout zlepšení, a které jsou popsány dále.

4.7.1 Eroze

Aplikace operace eroze slouží k odstranění případného šumu. Touto operací jsou tedy odstraněny pixely, které jsou označeny jako příslušící ke třídě „hráz“,



Obrázek 4.9: Vlevo stav před aplikací eroze, vpravo po.



Obrázek 4.10: Vlevo stav před aplikací dilatace, vpravo po.

ale ve svém bezprostředním okolí žádné další pixely indikující přítomnost hráze nemají. Stav před a po této operaci je vidět na obrázku 4.9.

4.7.2 Dilatace

Dilatace může pomoci udělat hráze více spojitě tam, kde na výstupu z modelu chybí několik pixelů klasifikovaných jako hráz. Zároveň také opraví případné nesouvislosti vytvořené předchozí operací eroze. Příklad, jak aplikace této operace vypadá ilustruje 4.10.

4.7.3 Úprava dle DEM

V případě, kdy je koryto řeky velmi úzké, což při horizontálním rozlišení dat 10 m na pixel může snadno znamenat koryto o šířce 1 px, a hrázích na obou březích řeky může nastat situace, kdy model predikuje hráz jen jako jednu širokou. Příklad lze vidět na levém obrázku 4.11. Toto je vyřešeno tím způsobem, že jsou z predikce odstraněny všechny pixely u kterých je dle



Obrázek 4.11: Vlevo problém, který může nastat pokud je koryto řeky úzké a hráze jsou na obou stranách řeky. Vpravo opravení odečtením bodů, které jsou nižší, než jejich okolí.



Obrázek 4.12: Vlevo stav před skeletonizací, vpravo stav po skeletonizaci.

DEM evidentní, že o hráz jít nemůže – tedy takové pixely které mají nižší nadmořskou výšku než průměr jejich bezprostředního okolí. Jak se takové úprava projeví je ilustrováno pravým obrázkem 4.11.

4.7.4 Skeletonizace

Před převedením polohy a tvaru hrází do vektorového formátu je žádoucí aby ve formátu rastrovém měla každá hráz šířku jednoho pixelu. Toto zajistíme operací skeletonizace. Stav před a po aplikaci skeletonizace je vyobrazen na obrázku 4.12.

4.8 Transformace do vektorového formátu

Jako vektorový formát, do kterého jsou výsledné identifikované hrázové struktury převedeny, je zvolen Esri Shapefile 1.4.2.2, jelikož v tomto formátu byly anotované hráze dodány. Za předpokladu, že je použit některý ze standardních vektorových formátů, by ale neměla být jeho konkrétní volba podstatná.

Prvním krokem převodu do vektorů je získání seznamu všech bodů, které jsou predikovány jako hráze. V dalším kroku je jeden bod odebrán ze seznamu a je z něj vytvořena vektorová křivka, následně se prochází všechny body v seznamu a kontroluje se, zda s již vytvořenou vektorovou křivkou bezprostředně sousedí, pokud ano, pak je takový bod odebrán ze seznamu a je o něj vektorová křivka prodloužena. Pokud již žádný bod s vektorovou křivkou nesousedí, je ze seznamu vybrán další bod na základě kterého je vytvořena další vektorová křivka. Proces vektorizace skončí až je seznam bodů prázdný.

4.9 Zhodnocení výsledků

V této práci bylo implementováno a otestováno několik modelů pokoušejících se identifikovat polohu ochranných hrází. Jako nejúspěšnější, z hlediska metriky IoU měřené na testovacích datech, se ukázal model CNN, architektury U-net s topologií v ukázce 4.1, využívající pouze digitální výškový model.

Úspěšnost se výrazně liší podle toho na jakých datech je predikce prováděna. Na datech geograficky pocházejících z České republiky a Polska je úspěšnost predikce uspokojivá, na datech z Rakouska je v porovnání s předchozími zeměmi úspěšnost horší. Je třeba konstatovat, že takto prezentovaná úspěšnost je pouze hodnotou jedné metriky modelu a nemusí nutně znamenat reálnou využitelnost či nevyužitelnost predikovaných dat, jejichž vlastnosti se mohou po post-processingových krocích měnit. Míru, do jaké byla predikce úspěšná, je potřeba posoudit v kontextu k čemu mají data sloužit, tedy k zpřesnění modelu povodňových rizik. Zda jsou predikovaná data pro tento účel dostatečně přesná, je potřeba určit lidmi, kteří mají znalost této domény a model povodňových rizik vytvářejí.

Dalším z cílů této práce bylo ověřit, zda je možné identifikaci ochranných hrází zpřesnit s využitím satelitních snímků. Za tímto účelem byl navržen model CNN, jehož vstupem byly digitální výškové modely a satelitní snímky pocházející z mise Sentinel-2. Takto vytvořený model měl na testovacích datech nepatrně horší úspěšnost, než model využívající pouze DEM. Ručním prozkoumáním predikcí se podařilo objevit, že v některých případech satelitní snímky predikci opravdu zpřesní. V průměru je ale celková úspěšnost horší, proto nelze s jistotou tvrdit, že jsou satelitní snímky přínosné.

4.10 Návrhy na možné budoucí vylepšení

Na základě poznatků z implementace a otestování několika přístupů, jak identifikovat ochranné hráze, je možné uvést několik návrhů na další vylepšení, které je možné otestovat, ale v této práci již vyzkoušeny nebyly.

Jelikož model s využitím satelitních snímků v některých případech predikuje lépe než model využívající jen DEM, nabízí se otestovat zkombinování výsledků obou modelů. Jakým způsobem by měla být kombinace provedena je předmětem dalšího testování.

Metrika IoU je sice standardně využívanou metrikou pro sémantickou segmentaci, ovšem v případě kdy anotovaná data tvoří křivky o šířce jednoho pixelu, nemusí být tato metrika vždy ideální. Je možné se zamyslet a zkusit navrhnout metriku, která bude lépe reflektovat povahu segmentovaných dat. Mohlo by jít například o to, že ne každý pixel je vyhodnocován nezávisle, ale jedna hráz je brána jako objekt a je vyhodnocováno do jaké míry je celé hráz například i s nějakou prostorovou tolerancí správně predikována. Zda je takovou metriku možné navrhnout a v praxi by fungovala je potřeba ověřit.

V neposlední řadě je určité prostor postupy navržené v rámci této práce vyzkoušet i na dalších datech, pokud taková data budou dostupná. Mimo jiné by tak bylo možné potvrdit nebo vyvrátit tvrzení, že kvalita predikce výrazně závisí na tom, zda jsou vlastnosti terénu na kterém jsou hráze predikovány stejné jako ty, na kterých byl model trénován.

Ukázka 4.1: Souhrn použité U-net konvoluční sítě.

| Layer (type) | Output Shape | Param # |
|--|---------------------|---------|
| input_1 (InputLayer) | [(None, 64, 64, 1)] | 0 |
| conv2d (Conv2D) | (None, 64, 64, 16) | 160 |
| batch_normalization (BatchNormalization) | (None, 64, 64, 16) | 64 |
| dropout (Dropout) | (None, 64, 64, 16) | 0 |
| conv2d_1 (Conv2D) | (None, 64, 64, 16) | 2320 |
| batch_normalization_1 (BatchNormalization) | (None, 64, 64, 16) | 64 |
| max_pooling2d (MaxPooling2D) | (None, 32, 32, 16) | 0 |
| conv2d_2 (Conv2D) | (None, 32, 32, 32) | 4640 |
| batch_normalization_2 (BatchNormalization) | (None, 32, 32, 32) | 128 |
| dropout_1 (Dropout) | (None, 32, 32, 32) | 0 |
| conv2d_3 (Conv2D) | (None, 32, 32, 32) | 9248 |
| batch_normalization_3 (BatchNormalization) | (None, 32, 32, 32) | 128 |
| max_pooling2d_1 (MaxPooling2D) | (None, 16, 16, 32) | 0 |
| conv2d_4 (Conv2D) | (None, 16, 16, 64) | 18496 |
| batch_normalization_4 (BatchNormalization) | (None, 16, 16, 64) | 256 |
| dropout_2 (Dropout) | (None, 16, 16, 64) | 0 |

4. IMPLEMENTACE

| | | |
|---|---------------------|--------|
| conv2d_5 (Conv2D) | (None, 16, 16, 64) | 36928 |
| batch_normalization_5 (BatchNormalization) | (None, 16, 16, 64) | 256 |
| max_pooling2d_2 (MaxPooling2D) | (None, 8, 8, 64) | 0 |
| conv2d_6 (Conv2D) | (None, 8, 8, 128) | 73856 |
| batch_normalization_6 (BatchNormalization) | (None, 8, 8, 128) | 512 |
| dropout_3 (Dropout) | (None, 8, 8, 128) | 0 |
| conv2d_7 (Conv2D) | (None, 8, 8, 128) | 147584 |
| batch_normalization_7 (BatchNormalization) | (None, 8, 8, 128) | 512 |
| max_pooling2d_3 (MaxPooling2D) | (None, 4, 4, 128) | 0 |
| conv2d_8 (Conv2D) | (None, 4, 4, 256) | 295168 |
| batch_normalization_8 (BatchNormalization) | (None, 4, 4, 256) | 1024 |
| dropout_4 (Dropout) | (None, 4, 4, 256) | 0 |
| conv2d_9 (Conv2D) | (None, 4, 4, 256) | 590080 |
| batch_normalization_9 (BatchNormalization) | (None, 4, 4, 256) | 1024 |
| conv2d_transpose (Conv2DTranspose) | (None, 8, 8, 128) | 131200 |
| concatenate (Concatenate) | (None, 8, 8, 256) | 0 |
| conv2d_10 (Conv2D) | (None, 8, 8, 128) | 295040 |
| batch_normalization_10 (BatchNormalization) | (None, 8, 8, 128) | 512 |
| dropout_5 (Dropout) | (None, 8, 8, 128) | 0 |
| conv2d_11 (Conv2D) | (None, 8, 8, 128) | 147584 |
| batch_normalization_11 (BatchNormalization) | (None, 8, 8, 128) | 512 |
| conv2d_transpose_1 (Conv2DTranspose) | (None, 16, 16, 64) | 32832 |
| concatenate_1 (Concatenate) | (None, 16, 16, 128) | 0 |
| conv2d_12 (Conv2D) | (None, 16, 16, 64) | 73792 |
| batch_normalization_12 (BatchNormalization) | (None, 16, 16, 64) | 256 |
| dropout_6 (Dropout) | (None, 16, 16, 64) | 0 |
| conv2d_13 (Conv2D) | (None, 16, 16, 64) | 36928 |
| batch_normalization_13 (BatchNormalization) | (None, 16, 16, 64) | 256 |
| conv2d_transpose_2 (Conv2DTranspose) | (None, 32, 32, 32) | 8224 |
| concatenate_2 (Concatenate) | (None, 32, 32, 64) | 0 |
| conv2d_14 (Conv2D) | (None, 32, 32, 32) | 18464 |
| batch_normalization_14 (BatchNormalization) | (None, 32, 32, 32) | 128 |
| dropout_7 (Dropout) | (None, 32, 32, 32) | 0 |
| conv2d_15 (Conv2D) | (None, 32, 32, 32) | 9248 |
| batch_normalization_15 (BatchNormalization) | (None, 32, 32, 32) | 128 |
| conv2d_transpose_3 (Conv2DTranspose) | (None, 64, 64, 16) | 2064 |

4.10. Návrhy na možné budoucí vylepšení

| | | |
|---|--------------------|------|
| concatenate_3 (Concatenate) | (None, 64, 64, 32) | 0 |
| conv2d_16 (Conv2D) | (None, 64, 64, 16) | 4624 |
| batch_normalization_16 (BatchNormalization) | (None, 64, 64, 16) | 64 |
| dropout_8 (Dropout) | (None, 64, 64, 16) | 0 |
| conv2d_17 (Conv2D) | (None, 64, 64, 16) | 2320 |
| batch_normalization_17 (BatchNormalization) | (None, 64, 64, 16) | 64 |
| conv2d_18 (Conv2D) | (None, 64, 64, 1) | 17 |

Total params: 1,946,705
Trainable params: 1,943,761
Non-trainable params: 2,944

Závěr

Cílem této práce bylo prozkoumat metody identifikace ochranných hrází a násypů liniových staveb v digitálním výškovém modelu. Dále ověřit, zda je pro zpřesnění identifikace vhodné použít, jako další vstup, obrazové satelitní snímky. Následně takto identifikované ochranné hráze a násypy liniových staveb převést do vhodného vektorového GIS formátu.

K identifikaci ochranných hrází a násypů liniových staveb je využito metod strojového učení. Jako vhodná metoda byla vybrána sémantická segmentace, která je realizována užitím konvolučních neuronových sítí. Jako konvoluční neuronová síť s nejlepší úspěšností pro tuto úlohu se podle měření metrikou Intersection Over Union ukázala být síť typu U-net.

Užitečnost obrazových satelitních snímků pro celkovou úspěšnost identifikace nebyla jednoznačně prokázána, neboť modely s použitím digitálních výškových modelů a zároveň satelitních snímků nedosahovaly úspěšnosti lepší, než modely s použitím pouze digitálních výškových modelů. Byly ovšem ukázány příklady, v kterých byly satelitní snímky přínosné, nebylo tedy na místě tvrzení o přínosu satelitních snímků zamítnout a byly prezentovány návrhy na další postupy, jak tato zjištění využít dále.

Dalším důležitým pozorováním bylo, že úspěšnost identifikace se výrazně liší podle toho, pro jakou geografickou polohou je identifikace prováděna. Z toho může vyplývat, že pro reálné použití je výhodné mít několik různých modelů, každý pro různé typy terénu, samozřejmě v závislosti na dostupných trénovacích datech. Toto tvrzení je ovšem potřeba potvrdit otestováním identifikace na větším množství a více rozdílných datech, než bylo provedeno v této práci.

Literatura

- [1] Epsilon7 : An image comparing Digital Elevation Models and Digital Surface Models. 2012, [cit 2020-08-20]. Dostupné z: <http://wiki.gis.com/wiki/index.php/File:DEMvsDSM.png>
- [2] Zhou, B.; Zhao, H.; Puig, X.; aj.: Scene Parsing through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [3] Karel Klouda, D. V., Juan Pablo Maldonado Lopez: BI-VZD přednáška 11. 2019, [cit 2020-09-13]. Dostupné z: <https://courses.fit.cvut.cz/BI-VZD/lectures/files/BI-VZD-11-cs-handout.pdf>
- [4] Afshine Amidi, S. A.: Deep Learning cheatsheets for Stanford's CS 230. [online], 2019, [cit 2020-09-15]. Dostupné z: <https://github.com/afshinea/stanford-cs-230-deep-learning>
- [5] Evan Shelhamer, T. D., Jonathan Long: Fully convolutional networks for semantic segmentation. 2015, [cit. 2020-09-15].
- [6] Olaf Ronneberger, P. F.; Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. [online], květen 2015, [cit. 2020-09-01]. Dostupné z: <https://arxiv.org/pdf/1505.04597.pdf>
- [7] Chaurasia, A.; Culurciello, E.: LinkNet: Exploiting encoder representations for efficient semantic segmentation. *2017 IEEE Visual Communications and Image Processing (VCIP)*, Dec 2017, doi: 10.1109/vcip.2017.8305148. Dostupné z: <http://dx.doi.org/10.1109/VCIP.2017.8305148>
- [8] Zhao, H.; Shi, J.; Qi, X.; aj.: Pyramid Scene Parsing Network. 2017, 1612.01105.

- [9] opencv dev team: OpenCV documentation. [cit 2020-09-17]. Dostupné z: https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html
- [10] JPEG: *Overview of JPEG 2000 [online]*. [cit. 2020-09-10]. Dostupné z: <https://jpeg.org/jpeg2000/>
- [11] Internet Engineering Task Force: *The GeoJSON Format [online]*. [cit. 2020-09-10]. Dostupné z: <https://tools.ietf.org/html/rfc7946>
- [12] The Sustainability of Digital Formats: *dBASE Table File Format (DBF) [online]*. [cit. 2020-09-10]. Dostupné z: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000325.shtml>
- [13] Knippers, R.: Spatial referencing, An overview. [cit. 2015-05-05]. Dostupné z: https://unstats.un.org/unsd/geoinfo/UNEGN/docs/_data_ICAcourses/_HtmlModules/_Documents/D06/documents/D06-03_KnippersPPTeaching.pdf
- [14] Kostecký, J.: Shlukování anomálií. Bakalářská práce. České vysoké učení technické, Fakulta informačních technologií, 2015, [cit. 2020-09-11].
- [15] Wu, M.; Zhang, C.; Liu, J.; aj.: Towards Accurate High Resolution Satellite Image Semantic Segmentation. *IEEE Access*, ročník 7, 2019: s. 55609–55619, doi:10.1109/ACCESS.2019.2913442.
- [16] He, K.; Zhang, X.; Ren, S.; aj.: Deep Residual Learning for Image Recognition. 2015, 1512.03385.
- [17] Wing, O. E.; Bates, P. D.; Neal, J. C.; aj.: A new automated method for improved flood defense representation in large-scale hydraulic models. *Water Resources Research*, ročník 55, č. 12, 2019: s. 11007–11034.
- [18] Choung, Y.: Mapping Levees Using LiDAR Data and Multispectral Orthoimages in the Nakdong River Basins, South Korea. *Remote Sensing*, ročník 6, č. 9, 2014: s. 8696–8717, ISSN 2072-4292, doi:10.3390/rs6098696. Dostupné z: <https://www.mdpi.com/2072-4292/6/9/8696>
- [19] Sofia, G.; Fontana, G. D.; Tarolli, P.: High-resolution topography and anthropogenic feature extraction: testing geomorphometric parameters in floodplains. *Hydrological Processes*, ročník 28, č. 4, 2014: s. 2046–2061, doi:<https://doi.org/10.1002/hyp.9727>, <https://onlinelibrary.wiley.com/doi/pdf/10.1002/hyp.9727>. Dostupné z: <https://onlinelibrary.wiley.com/doi/abs/10.1002/hyp.9727>
- [20] RNDr. Ladislav Plánka, C.: Kartografie I. [cit. 2021-05-05]. Dostupné z: <https://www.hgf.vsb.cz/export/sites/hgf/544/.content/galerie->

souboru/planka/kartografie_I/KARTOGRAFIE_I_06_KARTOMETRIE_STUDENTI.pdf

- [21] Wood, D. J.; Brown, C. R.; Doyle, L.; aj.: Identification of River Defences from Digital Terrain Models using Deep Learning. In *FLOODrisk 2020-4th European Conference on Flood Risk Management*, Budapest University of Technology and Economics, 2021.
- [22] Yakubovskiy, P.: Segmentation Models. https://github.com/qubvel/segmentation_models, 2019.
- [23] Simonyan, K.; Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015, 1409.1556.

Seznam použitých zkratk

ANN Artificial neural network

CNN Convolutional neural network

DEM Digital elevation model

DSM Digital surface model

DTM Digital terrain model

GIS Geographic information system

IoU Intersection over union

SRS Spatial reference system

Obsah přiloženého CD

| | |
|--------------------|---|
| readme.txt | stručný popis obsahu CD |
| src | |
| impl | zdrojové kódy implementace |
| thesis | zdrojová forma práce ve formátu \LaTeX |
| kosteja3.pdf | text práce ve formátu PDF |