



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Identifikace ochranných hrází a náspů liniových staveb
Student:	Bc. Jan Kostecký
Vedoucí:	Ing. Petr Pulc
Studijní program:	Informatika
Studijní obor:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	Do konce letního semestru 2020/21

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.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Karel Klouda, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 25. ledna 2020

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 17. září 2020

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2020 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 násypů liniových staveb*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Ochranné hráze a násypy liniových staveb výrazně ovlivňují rozsah záplav. Tato práce rozebírá postup jak tyto struktury identifikovat a vektorizovat do GIS formátu. Vstupem pro identifikaci jsou digitálních výškové modely a satelitní snímky Sentinel-2. K rozpoznání těchto struktur jsou užity konvoluční neuronové sítě.

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

Abstract

Leaves and embankments have great effect on the magnitude of flood. This work deals with identification and conversion to vector GIS format of these structures. The inputs for the algorithm are digital terrain models and Sentinel-2 satellite images. Convolutional neural networks are used for this task.

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	3
1.3 Sentinel	5
1.4 Formáty ukládání dat	5
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	7
1.5 Ochranné hráze a náspy liniových staveb	8
2 Algoritmy	9
2.1 Strojové učení	9
2.1.1 Přístupy k učení	9
2.1.2 Rozdělení učení podle cíle	9
2.1.2.1 Regrese	9
2.1.2.2 Klasifikace	10
2.1.2.3 Shlukování	10
2.2 Metriky měření úspěšnosti	10
2.2.1 Jaccardův koeficient	10
2.3 Účelová funkce	10
2.4 Umělé neuronové sítě	11
2.5 Konvoluční neuronové sítě	11
2.5.1 Plně konvoluční sítě	12
2.5.2 U-Net	12

2.6	Algoritmy počítačového vidění	14
2.6.1	Sémantická segmentace	14
2.6.2	Dilatace	14
2.6.3	Eroze	15
3	Implementace	17
3.1	Použité knihovny	17
3.2	Vstupní data	18
3.2.1	Satelitní snímky	18
3.2.2	Digitální výškové modely	18
3.2.3	Ochranné hráze a násypy liniových staveb	18
3.3	Předzpracování dat	20
3.3.1	Augmentace dat	20
3.4	Trénování modelu	20
3.4.1	Parametry modelu	20
3.4.2	Úspěšnost modelu	21
3.5	Transformace do vektorového formátu	21
	Závěr	25
	Literatura	27
	A Seznam použitých zkratk	29
	B Obsah příloženého CD	31

Seznam obrázků

1.1	DEM vs. DSM [1]	4
2.1	jednovrstvý perceptron [2]	12
2.2	Konvoluční neuronová síť [3].	12
2.3	Schéma plně konvoluční neuronové sítě [4].	13
2.4	Příklad U-net architektury převzatý z [5].	13
2.5	Příklad sémantické segmentace [6]	14
2.6	Vlevo původní stav, vpravo stav po aplikaci dilatace. [7]	15
2.7	Vlevo původní stav, vpravo stav po aplikaci eroze. [7]	15
3.1	RGB satelitní snímek	18
3.2	Normalizovaný DTM	19
3.3	Příklad anotovaných hrází a náspů liniových staveb	19

Úvod

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

Cílem této práce je identifikovat vyvýšené liniové stavby v digitálním výškovém modelu, jako další vstup pro zpřesnění použít obrazové satelitní snímky. Z takto identifikovaných staveb vytvořit co nejspojitéjší ale zároveň co nejvíce reálnou vektorovou reprezentaci a následně převést do vhodného GIS formátu.

Tato práce obsahuje v první kapitole přehled základních pojmů, kterým je potřeba porozumět pro pochopení problematiky. V druhé kapitole jsou definovány a rozebrány používané metody a algoritmy. Třetí kapitola pojednává o implementaci, která se skládá z předzpracování dat, trénování modelu a vektorizací výstupu modelu.

Základní pojmy

1.1 Digitální výškové modely

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

Výškový model může být reprezentován například rastrovou mřížkou, kde má každý bod mřížky přiřazenou informaci o jeho poloze (zeměpisné šířce a délce) a nadmořské výšce. Další možností jak reprezentovat výškový model je vektorová metoda zvaná TIN (triangulated irregular network), které ovšem není v této práci používána a nebude dále diskutována.

U digitálních výškových modelů se běžně užívají tři zkratky: DEM, DTM a DSM.

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).

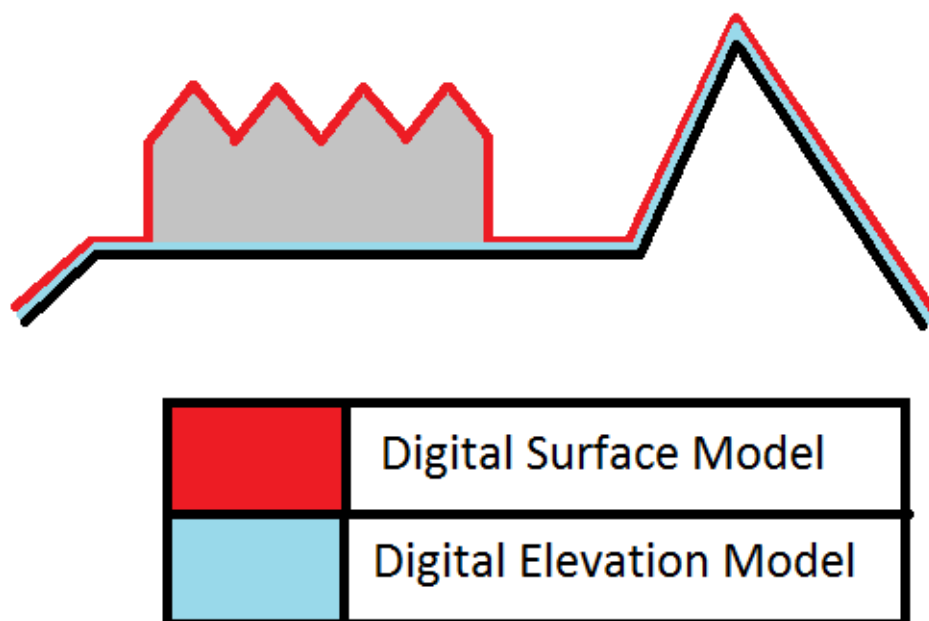
DTM obvykle popisuje nadmořskou výšku terénu bez člověkem uměle vytvořených staveb, ale i bez dalších přírodních objektů jako jsou například stromy. Někdy je ovšem DTM obohacen vektorovými informacemi o poloze řek, příkopů či hřebenů.

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

Rozdíl mezi DTM a DSM ilustruje obrázek 1.1

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.



Obrázek 1.1: DEM vs. DSM [1]

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, umí zobrazit pouze body v České a Slovenské republice a okolí, ale za to s větší přesností, než systémy globální.

Každý běžně používaný souřadnicový systém je jednoznačně identifikován pomocí SRID (spatial reference identifier).

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, které obstarávají sběr dat z různých zdrojů, jak ze satelitů jako v případě mise Sentinel, tak z pozemních stanic, letadel a dalších.

Mise Sentinel je rozdělena do několika částí a každá z nich obstarává 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 hladiny 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 60–10 m na pixel.

V nejjemnějším rozlišení 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 softwarovém nástrojích se 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 pomocí rovnic 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 a oproti jiným formátům umožňuje uložit více samostatných 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 (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 [8], což umožňuje oproti původnímu JPEG lepší zachování kvality grafiky i při stejné velikosti souboru. Formát umožňuje ale i bezztrátovou komprese. Dle [8] 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 používá syntaxi formátu JSON (JavaScript Object Notation). Nad rámec tohoto formátu definuje několik typů JSON objektů (např. Position, Point, MultiPoint, LineString, MultiLineString, Polygon, MultiPolygon), které jsou blíže specifikovány v [9].

Pro tuto práci je důležitý především typ objektu LineString (lomená čára). Objekt se definuje pomocí seznamu souřadnic bodů, které lomenou čáru tvoří. Příklad přesného formátu 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.

Povinné jsou tři soubory, které musí mít stejný název, ale mají různé přípony:

1. ZÁKLADNÍ POJMY

- .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 poloze každého objektu v .shp souboru
- .dbf – dodatečné atributy ve formátu dBase [10] pro objekty v .shp souboru

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

- .prj – informace o souřadnicovém systému a geodetickému datu [11], příklad v ukázce 1.3
- .cpg – kódování znaků v .dbf souboru
- .shp.xml – metadata v XML formátu

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 násypy liniových staveb

Obecně je liniová stavba definovaná jako stavba, kde podstatně převažuje délka nad výškou a šířkou. V kontextu této práce jsou zajímavé 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 jedné ze dvou (binární klasifikace) nebo více tříd (multinomiální klasifikace).

2.1.2.3 Shlukování

Shlukovací algoritmy patří do množiny algoritmů učení bez učitele. Shlukování rozděljuje vstupní data do několika podmnožin tak, aby si data v jedné podmnožině byla nějakým způsobem podobná a zároveň byla od dat v ostatních podmnožinách rozdílná. Tato podobnost je definována nějakou metrikou nebo mírou podobnosti. [12]

2.2 Metriky měření úspěšnosti

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 koeficient. Volba té správné metriky závisí na tom jaký mají modelovaná data charakter a je velmi důležitá pro správné vyhodnocení. V této práci je potřeba definovat Jaccardův koeficient neboli IoU (intersection over union).

2.2.1 Jaccardův koeficient

Jaccardův koeficient pro vektory reálných čísel je definován jako 2.1. Obecně pro množiny je používán vztah 2.2. Jak lze ze vzorce vyčíst, nabývá hodnoty z intervalu $< 0, 1 >$, kde 0 znamená, že množiny X a Y nemají žádný společný průnik a 1 znamená, že jsou množiny X a Y shodné.

$$J(x, y) = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 - \sum_{i=1}^n x_i y_i} \quad (2.1)$$

$$J(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|} \quad (2.2)$$

2.3 Účelová funkce

Součástí matematické optimalizace musí být funkce, pomocí které odhadujeme aktuální chybu modelu, takové funkci se říká účelová funkce (loss function). Ve strojovém učení je při procesu učení minimalizován chyba účelová funkce měřené na trénovací 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.

2.4 Umělé neuronové sítě

Umělé neuronové sítě (ANN – Artificial Neural Networks) jsou statickým modelem, který je inspirován biologickými neurony a jejich propojením. Základní stavební jednotka neuronové sítě je neuron, který na vstupu přijímá několik reálných čísel a na výstupu vrací jedno reálné číslo. [2]

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

Perceptron má na vstupu vektor reálných čísel (x_1, x_2, \dots, x_n) , dále je součástí perceptronu vektor vah (w_1, w_2, \dots, w_n) a intercept w_0 . Vnitřní potenciál perceptronu označený ξ spočteme vztahem 2.3.

Schéma perceptronu je znázorněno na obrázku 2.1. Perceptron má na vstupu vektor reálných čísel (x_1, x_2, \dots, x_n) , dále je součástí perceptronu vektor vah (w_1, w_2, \dots, w_n) a intercept w_0 . Vnitřní potenciál perceptronu označený ξ spočteme vztahem 2.3.

$$\xi = w_0 + \sum_{i=1}^n w_i x_i \quad (2.3)$$

Výstupem perceptronu je pak hodnota funkce $f(\xi)$. Této funkci se říká aktivizační funkce a v případě perceptronu je to funkce skoková jejíž předpis je určen vztahem 2.4.

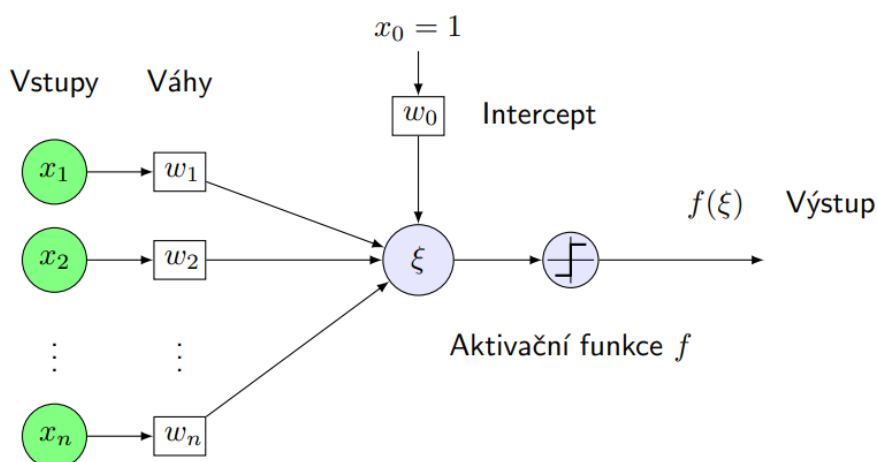
$$f(\xi) = \begin{cases} 1 & \text{když } \xi \geq 0 \\ 0 & \text{když } \xi < 0 \end{cases} \quad (2.4)$$

2.5 Konvoluční neuronové sítě

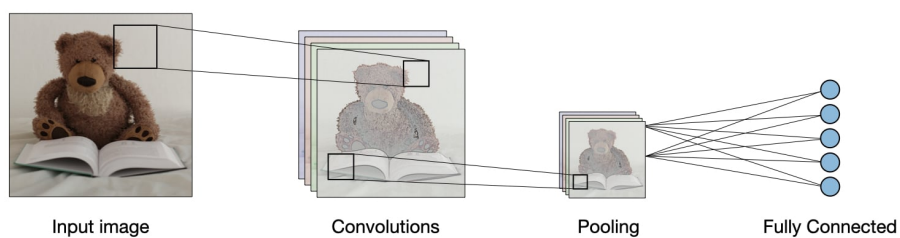
Hluboká neuronová síť (DNN – deep neural network) je ANN, která má mezi vstupní a výstupní vrstvou vrstvy další. Jedním ze speciálních případů hlubokých neuronových sítí jsou konvoluční neuronové sítě (CNN). CNN jsou používány především při analýze a zpracování obrazových dat v rastrovém formátu.

CNN má obvykle strukturu, která je na obrázku 2.2. Skládá z konvoluční vrstvy, pooling vrstvy a plně propojené vrstvy (fully connected).

[3]



Obrázek 2.1: jednovrstvý perceptron [2]



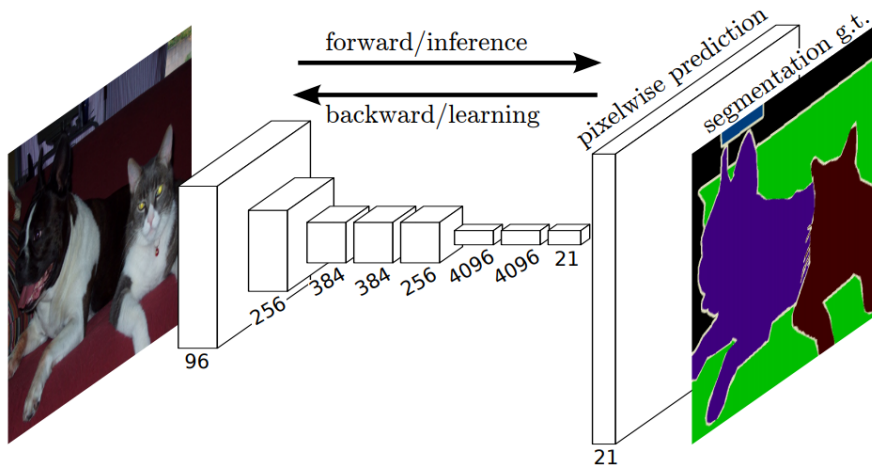
Obrázek 2.2: Konvoluční neuronová síť [3].

2.5.1 Plně konvoluční síť

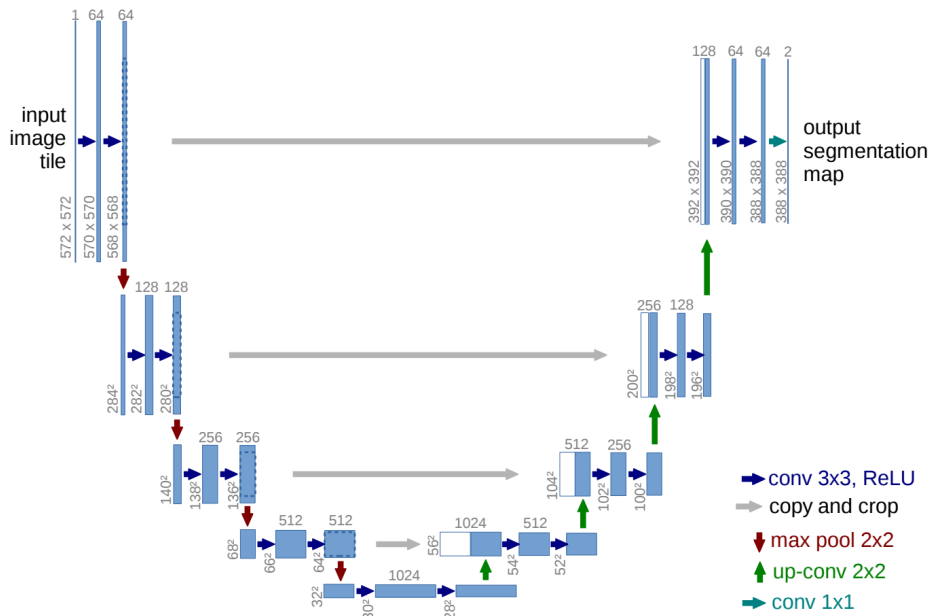
Plně konvoluční neuronové sítě obsahují oproti standardním CNN navíc upsampling [4], kdy se pomocí dekonvolucí získává obrazový výstup požadované velikosti.

2.5.2 U-Net

U-net, představena autory Olaf Ronneberger, Philipp Fischer a Thomas Brox v roce 2015 [5], 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 [5] je znázorněn na 2.4.



Obrázek 2.3: Schéma plně konvoluční neuronové sítě [4].



Obrázek 2.4: Příklad U-net architektury převzatý z [5].



Obrázek 2.5: Příklad sémantické segmentace [6]

2.6 Algoritmy počítačového vidění

Počítačové vidění je jedním z nových a rychle se vyvíjejících oborů s velkým praktickým využitím ve spoustě oblastí. Cílem je aby byl počítač schopný porozumět tomu, co obrazová data reprezentují, a získat z nich užitečné informace. Použití je opravdu široké, spousta lidí každý den odemyká svůj mobilní telefon pomocí funkce rozpoznávání obličeje, což je jen důmyslná aplikace metod počítačového vidění. 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.

2.6.1 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 2.5.

Pro sémantickou segmentaci je dle [4] jedním z nejlepších přístupů použití plně konvolučních neuronových sítí (FCN – fully convolutional networks).

2.6.2 Dilatace

Dilatace je jednou z operací matematické morfologie. Spočívá v aplikaci konvoluce konvoluční masky zvolené velikosti. V tomto případě konvoluční maska počítá maximální hodnotu pixelu a touto hodnotou je nahrazen pixel ve středu konvoluční masky. Výsledek takovéto operace lze vidět na obrázku 2.6. Bílé regiony se rozrostly, operace je tedy vhodná například na vyplnění nechtěných mezer.



Obrázek 2.6: Vlevo původní stav, vpravo stav po aplikaci dilatace. [7]



Obrázek 2.7: Vlevo původní stav, vpravo stav po aplikaci eroze. [7]

2.6.3 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 2.7

Implementace

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

3.1 Použité knihovny

Implementovat algoritmy strojového učení od základu v samotném jazyku Python by bylo velmi časově náročné, proto je použito několik užitečných knihoven popsaných dále.

- **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.
- **Keras** – API pro hluboké učení, používá Tensorflow.



Obrázek 3.1: RGB satelitní snímek

3.2 Vstupní data

Zadavatelem práce byla poskytnuta data z oblasti České republiky, Rakouska, Polska a Německa. Poskytnutá data se skládají z rastrového digitálního terénního modelu ve formátu GeoTIFF a vektorových hrází ve formátu Esri Shapefile. Dalšími jsou vstupními daty satelitní snímky Sentinel-2.

3.2.1 Satelitní snímky

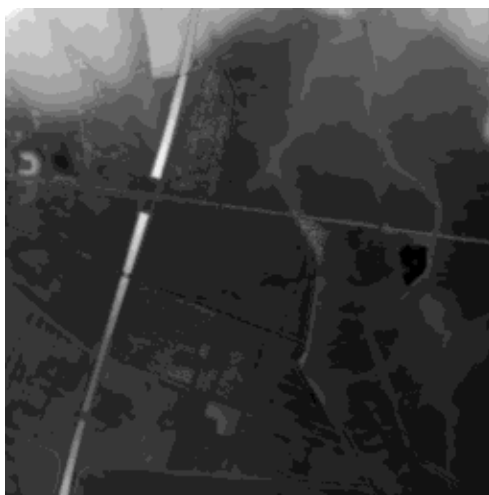
Satelitní snímky z mise Sentinel-2 poskytované evropskou kosmickou agenturou jsou dostupné z Copernicus Open Access Hub. Snímky jsou uloženy ve formátu Jpeg 2000 1.4.1.2. Použity byly snímky typu Level-2A v nejvyšším dostupném rozlišení, které činí 10 m. Snímky jsou organizovány v dlaždicích a je u nich použit souřadnicový systém UTM WSG84. Ukázkou části takového snímku je obrázek 3.1.

3.2.2 Digitální výškové modely

Digitální výškové modely byly poskytnuty zadavatelem práce a ne vždy jsou tyto modely volně dostupné. Rozlišení modelů je 10 m a jsou v různých systémech souřadnic podle toho, které území výškově popisují.

3.2.3 Ochranné hráze a násypy liniových staveb

Vektory ochranných hrází a násypů liniových staveb byly dodány zadavatelem ve formátu Esri Shapefile v různých systémech souřadnic. Tyto vektory jsou vlastně přesně to co chceme jako výstup, v kontextu strojového učení jsou to tedy anotovaná data.



Obrázek 3.2: Normalizovaný DTM



Obrázek 3.3: Příklad anotovaných hrází a náspů liniových staveb

3.3 Předzpracování dat

Předtím než mohou být data použita k natrénování modelu, je nutné provést jejich předzpracování.

První krokem je rasterizace vektorů liniových staveb. K tomu je využita funkce *RasterizeLayer* z knihovny *Gdal*. Je důležité dát pozor aby byla rasterizace provedena ve stejném prostorovém rozlišení 10 m, které mají DTM a satelitní snímky.

Je potřeba aby DTM, rasterizované liniové stavby i satelitní snímky byly zobrazené ve stejném souřadnicovém systému. Dalším krokem předzpracování je tedy převod mezi souřadnicovými systémy. Toho je dosaženo pomocí funkce *Warp* taktéž z knihovny *Gdal*.

V následujícím kroku jsou z DTM, satelitních snímků i liniových staveb v rastrovém formátu se stejným souřadnicovým systémem, vyříznuty obrázky ve velikosti 256x256, které si prostorově odpovídají. Takto oříznutý DTM, je navíc ještě normalizován na intervalu $< 0, 255 >$.

Ukázky výsledku takového předzpracování jsou na obrázcích 3.2, 3.1 a 3.3.

3.3.1 Augmentace dat

Dostupných dat, převážně pak anotovaných liniových staveb, ale i DTM, je omezené množství, které by nebylo pro natrénování modelu dostatečné. Vyřešit tento nedostatek lze augmentací dat pomocí rotací o úhel a překlopením, ale i deformací či přidáním šumu. V případě této práce je i náhodně volen počátek vyřezávaného obrázku velikosti 256x256.

3.4 Trénování modelu

Jako model byla pro tuto úlohu zvolena konvoluční neuronová síť U-net 2.5.2. K její implementaci bylo využito knihoven *Keras* a *Tensorflow*. Strukturu této sítě popisuje výstup funkce *model.summary()* 3.1.

Vstupem do U-net je *numpyarray* o rozměrech (256, 256, 4), je to vlastně satelitní snímek se třemi barevnými kanály (RGB) a k němu jsou jako 4. kanál přidány stupně šedi z DTM.

Výstupem je binární obrázek velikosti 256x256, kde hodnota 255 (bílá) znamená, že se na daném pixelu nachází liniová stavba. Ostatní pixely mají hodnotu 0.

3.4.1 Parametry modelu

Při trénování modelu je potřeba správně zvolit určité parametry, tak aby bylo trénování úspěšné.

- **Metrika** – Jako metrika je zvolen Jaccardův koeficient.

- **Účelová funkce** – Účelová funkce je ve tvaru 1 - Jaccardův koeficient.
- **Optimizer** – Optimizer je algoritmus pomocí kterého je optimalizována účelová funkce aby mohli být následně aktualizovány váhy neuronové sítě. V této práci byl zvolen *Nadam*.
- **Míra učení (learning rate)** – Určuje jak velké jsou kroky při hledání minima účelové funkce. V této práci byly voleny hodnoty v řádu 0.0001.
- **Velikost batche** – Velikost batche určuje po kolika vstupech dojde k aktualizaci vah. Jako defaultní hodnota na které při ladění tohoto parametru začít se doporučuje 32. Z důvodů výpočetní optimalizace se doporučuje volit velikost jako mocninu 2. V této práci byly otestovány hodnoty 2, 4, 8, 16 a 32 a nebylo pozorováno výraznějších rozdílů, lišila se pouze rychlost klesání hodnoty účelové funkce.
- **Počet epoch** – Jedna epocha odpovídá jednomu použití všech vstupních dat. Počet epoch může být tak velký dokud se úspěšnost modelu zlepšuje na trénovací množině a zároveň nedochází k přeučení, tedy úspěšnost roste i na validačních datech.

3.4.2 Úspěšnost modelu

Konvoluční síť U-net se strukturou 3.1 a parametry popsány výše dosáhla úspěšnosti, měřenou Jaccardovým koeficientem, **0.60** na trénovací množině a **0.32** na množině validační. Trénink byl ukončen potom, co po více jak 5 epochách za sebou neklesla hodnota účelové funkce tréninkové množině.

3.5 Transformace do vektorového formátu

Výstup predikovaný modelem je před převodem do vektorového formátu upravit tak, aby byly jednotlivé rozpoznané liniové struktury co nejspojitější. Za tímto účel jsou na výstup modelu aplikovány operace **dilatace** a následně **eroze**.

Identifikace jednotlivých liniových struktur je provedena užitím funkce *connectedComponents* z knihovny *OpenCV*. Tato metoda funguje obdobně jako hledání souvislých komponent z teorie grafů. Tímto je získán seznam pixelů, které tvoří jednotlivé liniové struktury. Souřadnice pixelu jsou převedeny do systému souřadnic geografického zobrazení a následně transformovány do formátu GeoJSON.

Ukázka 3.1: Souhrn použité U-net konvoluční sítě.

Layer (type)	Output Shape	Param #
input_20 (InputLayer)	[(None, 256, 256, 4)]	0
conv2d_361 (Conv2D)	(None, 256, 256, 16)	592

3. IMPLEMENTACE

batch_normalization_342 (BatchNormalization)	(None, 256, 256, 16)	64
dropout_171 (Dropout)	(None, 256, 256, 16)	0
conv2d_362 (Conv2D)	(None, 256, 256, 16)	2320
batch_normalization_343 (BatchNormalization)	(None, 256, 256, 16)	64
max_pooling2d_76 (MaxPooling2D)	(None, 128, 128, 16)	0
conv2d_363 (Conv2D)	(None, 128, 128, 32)	4640
batch_normalization_344 (BatchNormalization)	(None, 128, 128, 32)	128
dropout_172 (Dropout)	(None, 128, 128, 32)	0
conv2d_364 (Conv2D)	(None, 128, 128, 32)	9248
batch_normalization_345 (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d_77 (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_365 (Conv2D)	(None, 64, 64, 64)	18496
batch_normalization_346 (BatchNormalization)	(None, 64, 64, 64)	256
dropout_173 (Dropout)	(None, 64, 64, 64)	0
conv2d_366 (Conv2D)	(None, 64, 64, 64)	36928
batch_normalization_347 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_78 (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_367 (Conv2D)	(None, 32, 32, 128)	73856
batch_normalization_348 (BatchNormalization)	(None, 32, 32, 128)	512
dropout_174 (Dropout)	(None, 32, 32, 128)	0
conv2d_368 (Conv2D)	(None, 32, 32, 128)	147584
batch_normalization_349 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_79 (MaxPooling2D)	(None, 16, 16, 128)	0
conv2d_369 (Conv2D)	(None, 16, 16, 256)	295168
batch_normalization_350 (BatchNormalization)	(None, 16, 16, 256)	1024
dropout_175 (Dropout)	(None, 16, 16, 256)	0
conv2d_370 (Conv2D)	(None, 16, 16, 256)	590080
batch_normalization_351 (BatchNormalization)	(None, 16, 16, 256)	1024
conv2d_transpose_76 (Conv2DTranspose)	(None, 32, 32, 128)	131200
concatenate_76 (Concatenate)	(None, 32, 32, 256)	0
conv2d_371 (Conv2D)	(None, 32, 32, 128)	295040
batch_normalization_352 (BatchNormalization)	(None, 32, 32, 128)	512
dropout_176 (Dropout)	(None, 32, 32, 128)	0
conv2d_372 (Conv2D)	(None, 32, 32, 128)	147584
batch_normalization_353 (BatchNormalization)	(None, 32, 32, 128)	512
conv2d_transpose_77 (Conv2DTranspose)	(None, 64, 64, 64)	32832

3.5. Transformace do vektorového formátu

concatenate_77 (Concatenate)	(None, 64, 64, 128)	0

conv2d_373 (Conv2D)	(None, 64, 64, 64)	73792
batch_normalization_354 (BatchNormalization)	(None, 64, 64, 64)	256

dropout_177 (Dropout)	(None, 64, 64, 64)	0

conv2d_374 (Conv2D)	(None, 64, 64, 64)	36928
batch_normalization_355 (BatchNormalization)	(None, 64, 64, 64)	256

conv2d_transpose_78 (Conv2DTranspose)	(None, 128, 128, 32)	8224

concatenate_78 (Concatenate)	(None, 128, 128, 64)	0

conv2d_375 (Conv2D)	(None, 128, 128, 32)	18464
batch_normalization_356 (BatchNormalization)	(None, 128, 128, 32)	128

dropout_178 (Dropout)	(None, 128, 128, 32)	0

conv2d_376 (Conv2D)	(None, 128, 128, 32)	9248
batch_normalization_357 (BatchNormalization)	(None, 128, 128, 32)	128

conv2d_transpose_79 (Conv2DTranspose)	(None, 256, 256, 16)	2064

concatenate_79 (Concatenate)	(None, 256, 256, 32)	0

conv2d_377 (Conv2D)	(None, 256, 256, 16)	4624
batch_normalization_358 (BatchNormalization)	(None, 256, 256, 16)	64

dropout_179 (Dropout)	(None, 256, 256, 16)	0

conv2d_378 (Conv2D)	(None, 256, 256, 16)	2320
batch_normalization_359 (BatchNormalization)	(None, 256, 256, 16)	64

conv2d_379 (Conv2D)	(None, 256, 256, 1)	17

Total params: 1,947,137
 Trainable params: 1,944,193
 Non-trainable params: 2,944

Závěr

Cílem této práce bylo implementovat nástroj, který z digitálních výškových modelů a satelitních snímků Sentinel-2 extrahuje ochranné hráze a náspy liniových staveb a následně je převede do vhodného vektorového GIS formátu tak, aby identifikované hráze a náspy měli co nejspojitéjší vektorovou reprezentaci, ale zároveň taková reprezentace byla co nejbližší reálnému stavu.

Za tímto účelem bylo použito metod strojového učení, konkrétně sémantické segmentace pomocí konvoluční neuronové sítě. Na výstup konvoluční neuronové sítě byly aplikovány morfologické operace pro odstranění mezer a šumu. Na závěr byly identifikovány souvislé komponenty a převedeny do formátu GeoJSON.

Jako další vylepšení v této práci implementovaného nástroje je možné se za prvé pokusit o zlepšení úspěšnosti modelu CNN, která pro reálné použití není dostatečná a to například zvětšením tréninkové množiny, či vhodnějším předzpracováním dat. Rezerva je zřejmě i v kvalitě vstupu, kdy by pro výrazně vyšší úspěšnost bylo potřeba lepšího rozlišení jak DTM, tak satelitních snímků. Za druhé je třeba vylepšit zpracování výstupu modelu, kde metoda prezentovaná v této práci není příliš úspěšná. Dilatace a eroze nejsou schopny odstranit všechny chyby modelu a tím pádem není metoda hledání souvislých komponent tou nejhodnější. Další problém metody souvislých komponent jsou křivky, které se kříží. Tyto jsou identifikovány jako jedna křivka což způsobí problémy při převodu na vektory.

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] 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>
- [3] 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>
- [4] Evan Shelhamer, T. D., Jonathan Long: Fully convolutional networks for semantic segmentation. 2015, [cit. 2020-09-15].
- [5] 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>
- [6] 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.
- [7] 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
- [8] JPEG: *Overview of JPEG 2000 [online]*. [cit. 2020-09-10]. Dostupné z: <https://jpeg.org/jpeg2000/>
- [9] Internet Engineering Task Force: *The GeoJSON Format [online]*. [cit. 2020-09-10]. Dostupné z: <https://tools.ietf.org/html/rfc7946>

LITERATURA

- [10] 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>
- [11] 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
- [12] Kostecký, J.: Shlukování anomálií. Bakalářská práce. České vysoké učení technické, Fakulta informačních technologií, 2015, [cit. 2020-09-11].

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

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