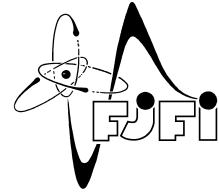




ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Algoritmy pro zpracování obrazových dat ze satelitů Sentinel

Algorithms for image data processing from Sentinel satellites

Bakalářská práce

Autor: **Emanuel Frátrik**
Vedúcí práce: **RNDr. Michal Šorel, Ph.D.**
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Emanuel Frátrik
Studijní program:	Aplikace přírodních věd
Studijní obor:	Matematické inženýrství
Studijní zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Algoritmy pro zpracování obrazových dat ze satelitů Sentinel
Název práce (anglicky):	Algorithms for image data processing from the Sentinel satellites

Pokyny pro vypracování:

- 1) Seznamte se se specifikací družic Sentinel-1 a Sentinel-2 a přístupem k jejich datům.
- 2) Nastudujte vhodné metody zpracování obrazu a strojového učení (konvoluční neuronové sítě, náhodné lesy, gradient boosting) podle pokynů vedoucího.
- 3) Zvolte vhodné třídy objektů pro rozpoznávání s ohledem na dostupnost mapových dat 'ground truth' v OpenStreetMap.
- 4) Navrhněte a implementujte rozpoznávací algoritmus založený na metodách strojového učení pomocí knihovny keras/tensorflow. Porovnejte přístup založený na příznamech vypočtených na předem daných regionech a plně konvolučních neuronových sítích.
- 5) Vyhodnoťte úspěšnost navržených algoritmů.

Doporučená literatura:

- 1) W. K. Pratt, Digital Image Processing (3rd ed.). John Wiley, New York, 2001.
- 2) I. Goodfellow, Y. Bengio, A. Courville, Deep Learning. MIT Press, 2016.
- 3) J. B. Campbell, R. H. Wynne, Introduction to Remote Sensing (5th ed.). Guilford Press, 2011.

Jméno a pracoviště vedoucího bakalářské práce:

RNDr. Michal Šorel, Ph.D.

Ústav teorie informace a automatizace AV ČR, Pod Vodárenskou věží 4, 182 08 Praha 8

Jméno a pracoviště konzultanta:

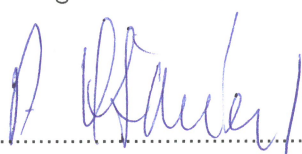
Datum zadání bakalářské práce: 31.10.2020

Datum odevzdání bakalářské práce: 7.7.2021

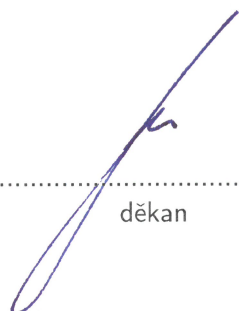
Doba platnosti zadání je dva roky od data zadání.

V Praze dne 30.10.2020


.....
garant oboru


.....
vedoucí katedry




.....
děkan

Pod'akovanie:

Chcel by som tu pod'akovať predovšetkým svojmu školiteľovi RNDr. Michalovi Šorelovi, Ph.D. za ochotu, ústretovosť, odborné zázemie a množstvo cenných rád pri vedení mojej bakalárskej práce.

Čestné prehlásenie:

Prehlasujem, že som túto prácu vypracoval samostatne a uviedol som všetku použitú literatúru.

V Prahe dňa 7. Júla 2020

Emanuel Frátrik

Názov práce:

Algoritmy pre spracovanie obrazových dát zo satelitov Sentinel

Autor: Emanuel Frátrik

Odbor: Matematické inženýrství

Zameranie: Aplikované matematicko-stochastické metódy

Druh práce: Bakalárska práca

Vedúci práce: RNDr. Michal Šorel, Ph.D., Ústav teórie informácie a automatizácie AV ČR Pod Vodárenskou věží 4 182 08 Praha 8

Abstrakt: Automatické mapovanie krajinného pokryvu a využitia územia je jednou z najdôležitejších úloh vzdialeného prieskumu Zeme. Problémom v tejto oblasti je nedostatok kvalitných ground truth dát, ktorých tvorba býva často náročná minimálne z časového hľadiska. V tejto práci ukážeme, že satelitné snímky generované misiou Sentinel-2 a mapové dáta z verejnej databázy OpenStreetMap tvoria prístupný a cenný zdroj dát na automatizáciu takejto tvorby datasetov krajinného pokryvu. Následne sú tieto datasety použité na riešenie klasifikácie a sémantickej segmentácie pokrytia krajiny v satelitných snímkach za pomoci algoritmov založených na metódach strojového učenia. Dosiahnuté výsledky sú nakoniec porovnávané a diskutované vzhľadom ku kvalite vytvorených modelov ale aj vzhľadom ku použitém algoritmom, počtu spektrálnych pásiem a nevyváženosti použitých datasetov.

Keľúčové slová: dataset, klasifikácia, krajinný pokryv, multispektrálne satelitné snímky, neurónová sieť, OpenStreetMap, Sentinel-2, sémantická segmentácia, strojové učenie, využitie územia

Title:

Algorithms for image data processing from the Sentinel satellites

Author: Emanuel Frátrik

Abstract: Automatic mapping of land cover and land use is one of the most important tasks of remote sensing. The problem in this area is the lack of ground truth data, the creation of which is often demanding at least in terms of time. In this work, we show that satellite imagery generated by the Sentinel-2 mission and map data from the public OpenStreetMap database form an accessible and valuable source of data to automate such land cover dataset creation. Subsequently, these datasets are used to solve the classification and semantic segmentation of landscape coverage in satellite images using algorithms based on machine learning methods. The achieved results are finally compared and discussed with respect to the quality of the created models but also with respect to the algorithms used, the number of spectral bands and the imbalance of the datasets used.

Key words: dataset, classification, land cover, multispectral satellite imagery, neural network, OpenStreetMap, Sentinel-2, semantic segmentation, machine learning, land use

Obsah

Použité skratky	11
Úvod	13
1 Klasifikácia pokrytia krajiny	15
2 Zdroje dát	17
2.1 Program Copernicus a Sentinel-2	17
2.2 Špecifikácia misie Sentinel-2	17
2.2.1 Obrazové dáta	19
2.3 OpenStreetMap	21
2.3.1 Dátový model OSM	23
2.3.2 Prístup k dátam OSM	24
3 Strojové učenie	25
3.1 Teória strojového učenia	25
3.1.1 Fázy učenia	26
3.2 Skupinové učenie (ensemble learning)	29
3.3 Ďalšie algoritmy	31
3.4 Neurónové siete	32
3.4.1 Konvolučné neurónové siete	34
4 Spracovanie dát a príprava datasetov	39
4.1 Dataset pre klasifikáciu	41
4.1.1 Použité triedy povrchov a objektov	42
4.2 Dataset pre sémantickú segmentáciu	44
4.2.1 Vytvorené sémantické triedy povrchov	46
4.2.2 Spektrálne charakteristiky snímok	46
5 Klasifikácia a sémantická segmentácia povrchov	53
5.1 Klasifikácia	53
5.1.1 Metodika	53
5.1.2 Experimenty a porovnanie	55
5.2 Sémantická segmentácia	63
5.2.1 Metodika	63
5.2.2 Experimenty a výsledky	65
Záver	81

Literatúra	89
Prílohy	91
Použité OSM značky na tvorbu datasetu pre klasifikačnú úlohu	91
Použité OSM značky na tvorbu datasetu pre segmentačnú úlohu	92
Hyperparametre určené pre algoritmy na klasifikačnú úlohu	93
Permutačné dôležitosti prediktorov algoritmu k-nn	95

Použité skratky

Skratka	Popis
NDVI	Normalized difference vegetation index
NDWI	Normalized difference water index
NDBI	Normalized difference built up index
SWIR	Krátkovlnné infračervené žiarenie (Shortwave Infrared Radiation)
NIR	Blízke infračervené žiarenie (Near Infrared Radiation)
OSM	OpenStreetMap
UTM	Universal Transverse Mercator
MGRS	Military Grid Reference System
WGS84	Svetový geodetický systém 1984 (World Geodetic System 1984)
k-NN	k-nearest neighbours
TOA	Top of atmosphere
BOA	Bottom of atmosphere
mIoU	mean Intersection over Union

Úvod

Zlepšenie prístupnosti satelitných a geo-priestorových dát, predovšetkým multispektrálnych obrázkov vo vysokom časovom a priestorovom rozlíšení nám v súčasnosti umožňuje efektívnejšie skúmať povrch Zeme a jeho zmeny, sledovať stav vegetácie a plodín na poľnohospodárskych poliach, určovať ich typ, zisťovať kvalitu vodných plôch a stanovovať presné hodnoty geofyzikálnych parametrov [58, 44, 59, 73]. Satelitné a geo-priestorové dáta sa ukazujú ako neoceniteľné aj pri sledovaní katastrof, riadenia krízových situácií ako sú povodne a požiare ale aj pri strategickom plánovaní [64, 19]. Veľké objemy takýchto dát vyžadujú automatické spracovanie a preto sú prirodzene vhodné na aplikácie štatistického modelovania, pokročilých metód počítačového videnia a strojového učenia.

V tejto práci sa venujeme aplikácii strojového učenia na automatické mapovanie a klasifikáciu krajinného pokryvu a využitia povrchu v satelitných snímkach. Takéto automatické mapovanie povrchu krajiny je dôležité pre už spomenuté sledovanie rozvoja životného prostredia ako aj jeho zmien. K tejto úlohe sa dá pristúpiť rôznymi spôsobmi ale najtypickejším prístupom je tzv. učenie s učiteľom (*supervised learning*), ktorý si osvojujeme aj v tejto práci. V prípade učenia s učiteľom je nutné mať prístup ku kvalitnému *ground truth* datasetu, obsahujúcemu dostatok vzoriek (*samples*) často pohybujúcich sa rádovo v stovkách tisícov až miliónoch. Práve požiadavka na rozsiahly dataset s *ground truth* je často limitujúcim faktorom v automatizácii mapovania povrchu v satelitných snímkach pretože jeho získanie môže byť časovo náročný proces vyžadujúci odborné znalosti. My sa tento problém pokúsime vyriešiť použitím mapových dát z verejne dostupnej databázy OpenStreetMap (OSM) a navrhnuť tak automatickú tvorbu datasetov teda bez nutnosti ručného označovania (*labeling-u*).

Ako zdroj multispektrálnych obrazových dát v našej práci slúži misia Sentinel-2 poskytujúca okrem iného obrázky s až 13 spektrálnymi pásmami čo z nej robí unikát v porovnaní s podobnými misiami akými sú napríklad Landsat a SPOT. Charakteristika misie Sentinel-2 ako aj popis databázy OSM bude spracovaný v kapitole 2.

Predtým ale ešte v kapitole 1 stručne priblížime problematiku klasifikácie krajinného pokryvu ako aj už vykonané práce a výskum v tejto oblasti.

Kapitola 3 prinesie stručný prehľad metód strojového učenia a algoritmov použitých v našej práci na klasifikáciu objektov akými sú *gradient boosting*, *random forest* a *k-nearest neighbours* ako aj pokročilých plne konvolučných neurových sietí použitých na klasifikáciu na úrovni pixelov teda sémantickú segmentáciu. V tejto kapitole podrobnejšie opíšeme architektúru neúronovej siete U-Net, ktorá bola vybratá nakoľko dosahuje tzv. *state of the art* výsledky v prípade sémantickej segmentácie povrchov a krajinného pokrytia [46].

Následne sa budeme v kapitole 4 venovať metodike tvorby datasetov postupne najprv pre klasifikačnú úlohu a následne tvorbe datasetu pre sémantickú segmentáciu.

V kapitole 5 popíšeme metódy použité na klasifikáciu a sémantickú segmentáciu objektov a povrchov v snímkach zo Sentinelu-2 s použitím vopred vypočítaných spektrálnych prediktorov

ale aj použitím moderných metód hlbokého učenia pomocu plne konvolučných neurónových sietí. Vyskúšame použitie časovej rady ako aj rôznych prediktorov a zhodnotíme ich vplyv na klasifikáciu a sémantickú segmentáciu povrchov. Taktiež tu zhrnieme výkony jednotlivých použitých algoritmov a ich nedostatky s ohľadom na riešené úlohy.

Kapitola 1

Klasifikácia pokrytia krajiny

Mapy pokrytia krajiny sú dôležitým nástrojom na zisťovanie charakteristík povrchu, plánovania rozvoja, detekciu zmien a pod. [30, 35]. Formálne sa rozlišuje mapovanie pokrytia krajiny a mapovanie využitia krajiny. V našej práci ale tieto dva termíny budeme používať zámenne.

Tvorba máp pokrytia povrchu často vyžaduje odborné znalosti ako aj znalosti daného územia a získavanie potrebných informácií je často veľmi zdĺhavý proces [49]. Prirodzene teda vzniká potreba automatizácie a zefektívnenia takejto tvorby. Ďalší prírôkový prírôk Zeme sa ukázal ako neoceniteľný zdroj dát umožňujúci automatizáciu klasifikácie pokrytia krajiny a sledovania jej zmien na lokálnej ale aj globálnej škále [35, 58]. Existuje veľa možností k prístupu dát z diaľkového prírôku a to napríklad Landsat Thematic Mapper a Enhanced Thematic Mapper poskytujúce multispektrálne snímky v strednom rozlíšení [85], Hyperion a AVIRIS poskytujúce hyperspektrálne snímky [31] ďalej QuickBird, IKONOS a Spot-5 [69] poskytujúce snímky vo vysokom priestorovom rozlíšení a v neposlednom rade Sentinel-2 poskytujúci multispektrálne snímky v strednom priestorovom ale vysokom časovom rozlíšení [54].

Medzi najjednoduchšie metódy klasifikácie povrchov v satelitných snímkach patria metódy používajúce prahovanie (*thresholding*) a výpočet spektrálnych indexov. Takéto indexy boli vyvinuté na kvantifikáciu biofyzikálnych charakteristík. Medzi často používané indexy patrí napr. *normalized difference vegetation index* NDVI [28], *normalized difference water index* NDWI [39] a jeho modifikácia MNDWI [83], *normalized difference built up index* NDBI [86] a nakoniec spomeňme ešte index OPTRAM [73] na kvantitatívny odhad vlhkosti pôdy. Výhodou použitia spektrálnych indexov je, že nevyžadujú tréningové vzorky dát. Medzi hlavné nevýhody ich použitia patrí to, že slúžia na zvýraznenie špecifickej vlastnosti povrchu. Okrem iného takéto indexy sú obvykle založené iba na využití vypočítaného lineárneho vzťahu, čo je často nedostačujúce na odlíšenie niektorých tried povrchov.

Ďalšie metódy klasifikácie snažiac sa čeliť limitácii využitia prahovania a spektrálnych indexov zahŕňajú strojové učenie a to konkrétne s učiteľom ale aj bez učiteľa. Výhodou učenia bez učiteľa je nepotrebnosť prístupu ku *ground truth* v dobe učenia. V oboch prípadoch sa môže dané učenie vykonávať buď na úrovni pixelov snímky (sémantická segmentácia) alebo na úrovni objektov. Oba prístupy prinášajú isté výhody ako aj nevýhody a voľba konkrétneho prístupu často závisí výhradne na dostupnom datasete. Podľa práce Plaza et al. [67] je prístup vzhľadom ku klasifikácii na úrovni objektov lepší na aplikáciu v snímkach s vysokým priestorovým rozlíšením vzhľadom k potrebe priestorovej informácie.

Faridatul a Wu narhujú vo svojej práci [34] tri nové spektrálne indexy na zlepšenie mapovania mestského pokrytia povrchu pričom ich navrhnutá metóda prekonáva dovtedy používané metódy založené na použití algoritmu *support vector machine*. Autori v [21] porovnávajú použitie tech-

niky založenej na prahovaní a použití indexov ako NDWI oproti použitiu algoritmov strojového učenia na mapovanie vodných plôch s rôznymi spektrálnymi a priestorovými charakteristikami. Záverom bolo, že použitie metód strojového učenia prinášalo zlepšenia v presnosti modelov vo väčšine prípadov. Ako hlavnú nevýhodu použitia strojového učenia považujú potrebu prístupnosti ku kvalitnému a rozsiahlemu datasetu. Autori taktiež považujú misiu Sentinel-1 a Sentinel-2 ako vhodný zdroj dát pre aplikáciu monitorovania vodných plôch. Li et al. [55] navrhli automatickú metódu klasifikácie štyroch povrchov na základe prahovania pomocou spektrálnych indexov a následnom použití vyselektovaných vzoriek dát na učenie algoritmu *support vector machine*. Zhang et al. [87] porovnávali použitie spektrálnych pásiem zo Sentinelu-2 a špecifických spektrálnych indexov na klasifikáciu štyroch tried povrchov pričom ukázali, že použitie konkrétnych pásiem (Red, NIR, SWIR) dosahuje lepšie klasifikačné výsledky ako prístup s indexami, pričom použitie všetkých 13 spektrálnych pásiem priniesol ešte ďalšie zlepšenie. Autori v práci [62] používajú algoritmus *random forest* na sémantickú segmentáciu pokrytia krajiny s využitím multispektrálnych snímok zo Sentinelu-2. Malinkovski et al. [59] vytvorili automatizovanú metódu na klasifikáciu pokrytia krajiny taktiež za pomoci algoritmu *random forest* s využitím datasetu Corine Land Cover na trénovanie.

Nemálo vyvíjaných metód používa na tvorbu výsledných klasifikačných máp pokrytia časové série snímok na zabezpečenie čo najvyššej presnosti modelov ako aj presnosti výsledných máp. Paris et al. navrhujú vo svojej práci [63] metódu založenú na učení bez učiteľa a s využitím časovej rady multispektrálnych snímok. Výsledná metóda priniesla zvýšenie celkovej metriky *accuracy* o 18% a 11% vzhľadom k referenčným datasetom Corine Land Cover 2012 a GlobalLand30.

Ako bolo spomenuté, využitie učenia s učiteľom prináša výzvu vzhľadom k prístupu datasetu s obsahujúceho *ground truth*. Projekty ako OSM predstavujú jednoduchý nástroj na extrakciu potrebných informácií o type povrchu resp. *ground truth* a sú považované za užitočný zdroj dát pri klasifikácii pokrytia a využitia krajiny, jeho zmien ale aj aktualizácie už existujúcich datasetov [33, 52].

Avšak použitie OSM dát umožňuje automatizáciu tvorby datasetov s *ground truth* za cenu možnej zníženej pozičnej ale aj značkovacej a tematickej presnosti nakoľko klasifikácia povrchov často závisí od subjektívneho vnímania prispievateľa. Taktiež úplnosť OSM dát často nedosahuje homogénne hodnoty v rozličných regiónoch pričom je pravdepodobnejšie, že v hustejšie obývaných oblastiach budú dáta kvalitnejšie označované [41, 43, 60]. Autori v [81] porovnávali OSM dáta s dvoma referenčnými datasetmi, menovite Corine Land Cover a s oficiálnou Portugalskou mapou pokrytia a odhadli tematickú presnosť OSM dát medzi 77.3% až 91.9%.

V neposlednom rade sa v súčasnej dobe používajú na riešenie úloh v počítačovom videní a rozpoznávaní obrazu s veľkým úspechom neurónové siete. Ich úspech spočíva v schopnosti detekcie relevantných vzorcov a štruktúr v obrazových dátach a preto je prirodzené, že sa aplikujú aj na spracovanie a klasifikáciu pokrytia v multispektrálnych snímkach [80]. Konkrétne v prípade sémantickej segmentácie povrchov prinášajú plne konvolučné neurónové siete niekoľko násobné zlepšenie oproti klasickým metódam [76]. V práci [74] využívajú autori OSM ako zdroj dát na tvorbu mapy pokrytia krajiny pričom oblasti s nedostatočným označením dopĺňajú pomocou metód hlbokého učenia. V práci [79] autori používajú architektúru plne konvolučnej siete U-Net na klasifikáciu povrchov v satelitných snímkach na úrovni objektov ale aj na úrovni pixelov. Navrhnutý klasifikačný model dosiahol vysoké celkové *F1 score* a to až 74.9% pri klasifikácii do 43 tried povrchov. Segmentačný model vykazoval vysokú hodnotu *mIoU* vzhľadom k triedam *forests*, *inland waters* a *arable land*. Niektorí autori preferovali využitie *transfer learning*-u a augmentácie snímok vzhľadom k limitovanej veľkosti trénovaciemu datasetu [22, 68].

Kapitola 2

Zdroje dát

V tejto časti priblížime čitateľovi zdroje dát neskôr využité na tvorbu datasetov. V prvom rade ide o misiu Sentinel-2 poskytujúcu multispektrálne obrazové dáta a v druhom rade OpenstreetMap (OSM), ktorý slúži ako zdroj mapových *ground truth* dát.

2.1 Program Copernicus a Sentinel

Program Copernicus má za úlohu sledovať našu planétu a jej prostredie pričom výsledné dáta a informačné služby sú pre všetkých používateľov bezplatné a verejne dostupné.

Pre osobitné potreby programu slúži rodina satelitov zvaných *Sentinel*. Konkrétne ide o 4 samostatné satelity resp. konštelácie satelitov, a to Sentinel-1, -2, -3, -6, zatiaľ čo satelity Sentinel-4 a -5 sú nástroje na palube meteorologických satelitov organizácie EUMETSAT. Medzi službami poskytovanými programom Copernicus sú monitorovanie atmosféry, morí, pevniny, zmien klímy, bezpečnosti a núdzových situácií.

V našej práci sa zameriavame na krajinný pokryv a preto je pre nás najvýznamnejšia služba poskytujúca dáta z pozorovania pevniny. Jedným z produktov služby pre monitorovanie pevniny programu Copernicus je tzv. CORINE Land Cover inventár krajinnej pokrývky v ktorom sa nachádza klasifikácia krajinnej pokrývky do 44 tried v troch hierarchických skupinách [3].

Každá zo siedmich misií Sentinel má svoju špecializáciu a zabezpečuje konkrétnu potrebu programu Copernicus. Hlavnou úlohou misie Sentinel-1 je monitorovanie krajiny a oceánu pričom toto je zabezpečené radarovým snímaním nezávislým na počasí a fáze dňa. Sentinel 3 poskytuje optické, radarové a výškové údaje pre námorné a pozemné služby. Misie Sentinel-4, -5 a -5P majú za hlavnú úlohu monitorovať stav a predovšetkým zloženie atmosféry pričom treba podotknúť, že misie Sentinel-4 a -5 ešte nezapočali. Podobne aj spustenie misie Sentinel-6, ktorej úlohou bude presné meranie morskej hladiny pre účely oceánografie a výskumu klímy sa plánuje.

2.2 Špecifikácia misie Sentinel-2

Ako už bolo spomenuté Sentinel-2 je jednou z misií programu Copernicus a poskytuje multispektrálne snímky s vysokým časovým a priestorovým rozlíšením. Tématicky naväzuje na misie Landsat a SPOT [16, 1]. Jeho hlavnou úlohou je systematické monitorovanie pevniny zahŕňajúce monitorovanie vegetácie a jej stavu napríklad mapovaním obsahu chlorofylu v listoch alebo tvorbou spektrálnych indexov ako napríklad vegetačných. Ďalej sa predpokladá jeho využitie v oblasti geomarketingu, dokumentácii stavu územia, podpory krízového riadenia, monitorovania lesných

Pásmo	Centrálne λ nm	Šírka - $\Delta\lambda$ nm	Rozlíšenie - m	Poznámka
B1	443	20	60	Detekcia rozptylu aerosólu
B2	490	65	10	Modrá
B3	560	35	10	Zelená
B4	665	30	10	Červená
B5	705	15	20	Vegetačný červený okraj
B6	740	15	20	Vegetačný červený okraj
B7	783	20	20	Vegetačný červený okraj
B8	842	115	10	Blízka infračervená (NIR)
B8A	865	20	20	Úzka blízka infračervená
B9	945	20	60	NIR - Absorpcia vodnej pary
B10	1375	30	60	SWIR - Detekcia tenkého cirrusu
B11	1610	90	20	Krátko-vlnná infračervená (SWIR)
B12	2190	180	20	SWIR

Tabuľka 2.1: Charakteristika spektrálnych pásiem vzhľadom k spektrálnemu a priestorovému rozlíšeniu získavanými inštrumentom MSI na palube družíc Sentinel-2. Zdroj: [13]

požiarov a dlhodobého sucha [14]. V neposlednom rade sa snímky zo Sentinelu-2 tešia použitiu v poľnohospodárstve alebo pri zisťovaní vodného stresu ako aj pri pozorovaní vnútrozemských vodných plôch a pobrežných oblastí.

Túto misiu tvorí konštelácia dvoch satelitov Sentinel-2A a Sentinel-2B na helio-synchronnej orbite. Dvojica satelitov je voči sebe posunutá o 180° . Sentinel-2A bol vypustený 23. júna 2015 raketou Vega z európskeho kozmodrómu vo Francúzskej Guyane a Sentinel-2B nasledoval z rovnakého miesta 7. marca 2017. Výška obežnej dráhy na ktorej satelity operujú je 786 km.

Samotné časové rozlíšenie teda čas, za ktorý družica preletí nad rovnakým územím je nad rovníkom 10 dní pre jeden satelit z konštelácie a teda iba 5 dní pre oba satelity v konštelácii. Dokonca v stredných zemepisných šírkach sa doba skraca na 2-3 dni pri bez-obláčnej atmosfére. V prípade priestorového rozlíšenia teda vzdialenosti na povrchu reprezentovanú stranou jedného pixela hovoríme o hodnotách 10, 20 a 60 metrov závislú na konkrétnom spektrálnom pásme. V praxi 10 m rozlíšenie umožní na obrázku rozoznať väčšie objekty akými sú napríklad les, obytná oblasť alebo jazero ale pri rozlišovaní jednotlivých stromov, úzkych ciest a menších domov nám nepomôže. S touto limitáciou budeme musieť počítať a na tvorbu datasetu vyberať len väčšie oblasti a objekty. V prípade radiometrického rozlíšenia, inštrument v Sentineli-2 zbiera dáta v rozlíšení 12 bitov. Tieto merania sú nakoniec postupne prekonvertované do odrazivosti (*reflectance*) a uložené ako 16-bitové rastrové snímky.

Šírka záberu snímania oboch satelitov resp. ich senzorov je 290 km. Snímanie zemského povrchu prebieha od 56° južnej šírky až po 84° severnej šírky. Optický senzor MSI na palube satelitov Sentinel-2 zachytáva až 13 spektrálnych pásiem od viditeľného až po stredné infračervené pásmo.

Práve unikátne parametre družíc Sentinel-2 a zaznamenávanie v multispektrálnej doméne zahŕňajúc až 3 pásma v tzv. vegetačnom červenom okraji (*vegetation red edge*) tvoria z misie Sentinel-2 ideálny zdroj dát na monitorovanie vegetácie a jej stavu. Stručný prehľad charakteristík spektrálnych pásiem zachytávaných multispektrálnym inštrumentom MSI na palubách družíc Sentinel-2 sú zhrnuté v tabuľke 2.1.

2.2.1 Obrazové dáta

Výsledky snímania multispektrálnym inštrumentom na palube družíc Sentinel-2 sú dostupné v produktoch s rôznou úrovňou spracovania. Medzi produkty, ktoré nie sú dostupné pre používateľov patria produkty typu Level-0, Level-1A. Voľne dostupné sú iba produkty typu Level-1B, Level-1C a Level-2A. Analýza produktu Level-1B vyžaduje aby mal používateľ znalosti ortorektifikačných metód.

Produkty Level-0, Level-1A a Level-1B sú tvorené z tzv. granúl čo sú akési najmenšie časti snímku. Granule sú dlhé 25 km pozdĺž záberu senzora a 23 km naprieč záberom senzora.

Granule tvoriace produkty Level-1C a Level-2A sa nazývajú dlaždice a líšia sa predovšetkým v type atmosferickej korekcie. V oboch prípadoch ide o štvorcové ortorektifikované rastre v projekcii UTM/WGS84 zaberajúce 110 x 110 km², ktoré vznikli tak, že sa každá bunka mriežky, vytvorenej použitím UTM zón a rovnobežkových zón pochádzajúcich z MGRS (*Military Grid Reference System*) delí ešte na podbunky rozmerov takmer 110 x 110 km². Každá podbunka resp. dlaždica má na okraji prekryv s vedľajšou dlaždicou zhruba 10 km z každej strany [7] a celá pokrýva presne vymedzenú časť územia čo poskytuje výhody hlavne pri opakovanom spracovaní a potrebe monitorovať presne určené územie. Pre názornú predstavu dlaždíc slúži obrázok 2.1.

Všetky produkty poskytované misiou Sentinel-2 sa skladajú z dátových položiek a metadát v rastrovom a vektorovom formáte a sú kompaktné uložené v súborovom kontajnery formátu SAFE.

Jednotlivé datové položky a metadáta zahŕňajú:

- obrazové dáta (dlaždice) v JPEG2000 formáte každého z 13 spektrálnych pásiem
- náhľad snímku vo formáte JPEG2000
- doplnkové dáta z družicovej telemetrie
- pomocné informácie popisujúce použité parametre
- indikátory kvality (masky oblačnosti, defektných pixelov a iné) popisujúce produkt a jeho radiometrické, geometrické a snímkové vlastnosti

Charakteristika prístupných produktov Level-1B, Level-1C a Level-2A je zhrnutá v tabuľke 2.2.

2.2.1.1 Atmosferické korekcie

Jedným z najdôležitejších krokov v predspracovaní satelitných snímok pre ďalšiu analýzu je zmiernenie vplyvu atmosféry. Zmiernenie vplyvu atmosféry sa v prípade Sentinelu-2 dosahuje konverziou TOA (*top of atmosphere*) odrazivosti (*reflectance*) v produkte Level-1C do odrazivosti BOA (*bottom of atmosphere*) v produkte Level-2A. BOA odrazivosť popisuje presnejšie spektrálne vlastnosti pozemných objektov a sledovaných povrchov pretože reprezentuje odrazivosť ako by bola pozorovaná v blízkosti daného objektu a povrchu. Je teda vhodnejšia na generovanie spektrálnych indexov alebo neskoršiu aplikáciu metód strojového učenia. Rozdiel medzi produktami v TOA a BOA odrazivosti sú ukázané na obrázku 2.2.

Podľa dokumentácie je konverzia z TOA na BOA odrazivosť založená na metóde navrhnutej v [70] a využíva LIBRADTRAN radiačný transfer model. Túto konverziu môže vykonať aj samotný užívateľ z produktu Level-1C v programe Sentinel-2 Toolbox [17] alebo pomocou procesora zvaného Sen2Cor [12].

Produkt	Popis	Distribúcia	Objem dát	Záber
Level-1B	TOA žiarivosť (<i>radiance</i>) v geometrii senzora	Systematická generácia a online distribúcia	27MB	25 x 23 km ² /granula
Level-1C	TOA odrazivosť (<i>reflectance</i>) v kartografickej geometrii (UTM/WGS84)	Systematická generácia a online distribúcia	800MB*	110 x 110 km ² /dlaždica*
Level-2A	BOA odrazivosť (<i>reflectance</i>) v kartografickej geometrii (UTM/WGS84)	Systematická generácia (od decembra 2018) dovedy pomocou Sentinel-2 toolboxu	1200MB*	110 x 110 km ² /dlaždica*

Tabuľka 2.2: Charakteristika úrovní spracovania jednotlivých produktov. * Dokumentácia udáva odlišné hodnoty pričom hodnoty v tabuľke boli určené priamo z daného produktu.

Detekcia oblačnosti a tieňov väčších oblakov v satelitných snímkach je dôležitým krokom v ich predspracovaní. Takáto oblačnosť môže úplne znemožňovať ďalšie analýzy v prípade hustej oblačnosti alebo v prípade tenkých priehľadných oblakov ako cirrusi a tieňov oblakov môže výrazne ovplyvňovať odhady bio-fyzikálnych a geo-fyzikálnych parametrov [82]. V prípade snímok zo Sentinelu-2 pozostáva detekcia oblačnosti z využitia viackrokového prahovania na základe znalosti charakteristických odrazivostí oblakov v 60 m rozlíšení v prípade spracovania Level-1C. V prípade spracovania Level-2A sa používa opäť prahovanie spolu s aplikáciou neurónových sietí [8].

Masky oblačnosti sú v produkte Level-1C zastúpené len vo vektorovom formáte (.gml) [6]. Produkt typu Level-2A obsahuje okrem vektorovej masky oblačnosti aj pravdepodobnostné masky oblačnosti a masky snehového zastúpenia v rastrovom formáte (.jp2). Okrem detekcie oblačnosti použitej priamo pri tvorení produktov je možné použiť aj detektor oblačnosti využívajúci algoritmus zvaný *gradient boosting* v implementácii LightGBM. Tento detektor oblačnosti bol vytvorený výskumným tímom Sentinel-hub [18].

Ako už bolo spomenuté produkt Level-2A obsahuje taktiež raster s klasifikáciou scény v snímke v 12 triedach zahŕňajúcich okrem iného oblačnosť so strednou a vysokou pravdepodobnosťou výskytu ďalej tenké cirrusi, oblasť s vegetáciou a bez, vodné plochy a tieň oblakov [8].

2.2.1.2 Spektrálne indexy

Spektrálne indexy predstavujú jeden z najjednoduchších spôsobov zvýrazňovania a kvantifikáciu istých vlastností povrchov a objektov v satelitných snímkach. Metóda výpočtu spektrálnych indexov je založená na pozorovaniach optických a fyzikálnych vlastností objektov a ich špecifických spektrálnych signatúrach. Takéto indexy vznikajú ako aplikácia nejakej funkcie, najčastejšie afinnej transformácie na rôzne spektrálne pásma rastrov pričom táto aplikácia prebieha na úrovni pixelov. Vlastnosťami a atribútmi, ktoré sú často predmetom záujmu a teda je vhodné ich zvýrazniť a kvantifikovať patria už spomínaná zdravosť vegetácie, prítomnosť vodnej plochy, snehu, geologického ložiska alebo človekom vystavanej oblasti ale môže ísť napríklad aj o odhad vlh-

kosti v pôde a podobne [48, 28, 39, 83, 73]. Medzi typicky používané spektrálne indexy patria napríklad normalizované diferenčné indexy s hodnotami v intervale [-1, 1].

Ukážka spektrálnych indexov NDVI a NDWI nad bez-oblačnou snímkou (Level-2A) z misie Sentinel-2 nad územím Prahy 1.8.2020 vygenerovaných pomocou webovej aplikácie EOBrowser [5] možno vidieť na obrázku 2.3

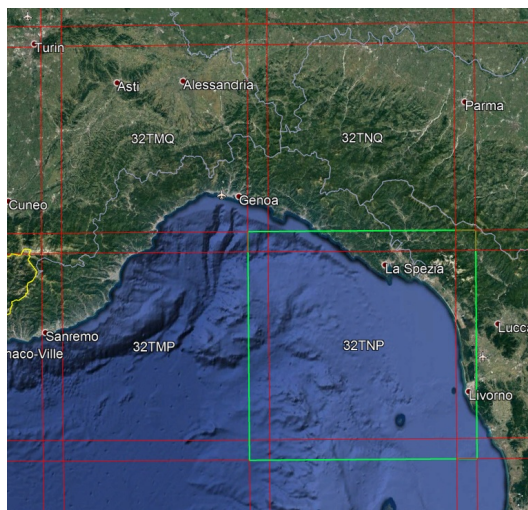
2.2.1.3 Prístup k snímkam

Dáta Sentinelu-2 sú voľne prístupné širokej verejnosti a dá sa k nim dostať rôznymi spôsobmi zahrňajúcimi interaktívne webové rozhrania ale aj aplikačné programovateľné rozhrania (API) pre interakciu s programovacími jazykmi. Medzi používané patrí napr.:

- Copernicus Open Acces Hub: <https://scihub.copernicus.eu/dhus/#/home>
- Český datový sklad prevádzkovaný CESNET-om: <https://collgs.czechspaceportal.cz/>
- API prístupované cez: <https://apihub.copernicus.eu/apihub/>

Viac informácií o oficiálnom prístupe k dátam nájde čitateľ v [2].

Nedávno vzniknutý a veľmi užívateľský prívetivý je aj webový program zvaný Sentinel-Hub [15], ktorý okrem iného poskytuje tzv. EOBrowser [5], výkonný nástroj na prehliadanie, vizualizáciu a analýzu satelitných snímok z rôznych misií medzi, ktorými nechýbajú snímky v spracovaní Level-1C a Level-2A zo Sentinelu-2.



Obr. 2.1: Na obrázku je vidieť systém dlaždíc a taktiež prekryv jednotlivých dlaždíc (10 km) z každej strany. Zvýraznená dlaždica pokrýva presne určené územie v okolí Talianska. Názov dlaždice 32TNP napovedá, že sa nachádza v zóne 32T systému *Military Grid Reference System* (MGRS), pričom NP ďalej upresňuje polohu tejto dlaždice v zóne 32T. Zdroj: [7]

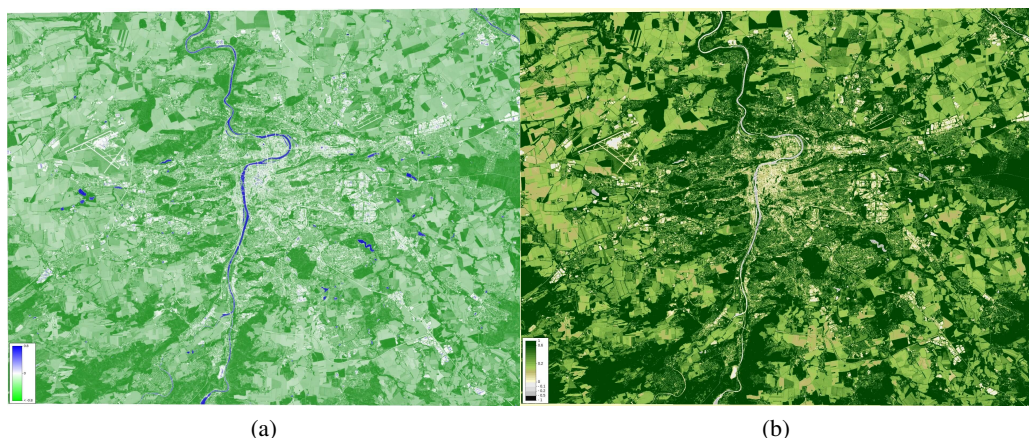
2.3 OpenStreetMap

OpenStreetMap (OSM) je voľne prístupná, editovateľná mapa celého sveta, ktorá je prevažne tvorená používateľmi a je licencovaná pod Open Database Licence. OSM poskytuje databázu

geo-priestorových dát vo vektorovom formáte definujúcich bodové, lineárne a mnohoúhľovníkové štruktúry a objekty spolu s popisom vlastností danými tzv. značkami (*tags*).



Obr. 2.2: Vizuálne porovnanie odrazivosti (*reflectance*) TOA **(a)** a BOA **(b)** v pravých farbách (*true color*). Zdroj: [5]



Obr. 2.3: Príklad spektrálnych indexov NDWI **(a)** na zvýraznenie vodných plôch a NDVI **(b)** na kvalitatívne rozlíšenie zdravosti vegetácie. V prípade NDWI indexu platí, že čím modrejšia farba tým je hodnota bližšia 1 poukazujúca na prítomnosť vodnej plochy. Podobne v prípade NDVI indexu platí, že čím zelenšia farba tým je hodnota bližšia 1 a reprezentuje zdravšiu resp. hustejšiu vegetáciu. Bližšie informácie o indexoch môže čitateľ dohľadať v [48, 83]. Zdroj: [5]

Práve voľná prístupnosť OSM z nej robí obľúbený a často používaný zdroj dát v štúdiách a vedeckých prácach [11]. Jednou z nevýhod projektov ako je OSM je možnosť vzniku nepresností napríklad v označení atribútov mapových objektov prirodzene plynúcich z podstaty jej tvorby za pomoci dobrovoľníkov. Miera nepresnosti OSM je popisovaná a sledovaná metrikami ako je pozičná presnosť a presnosť v značovaní. Ďalšou dôležitou metrikou OSM je jej úplnosť. Úplnosť OSM dát často dosahuje nižšie hodnoty v menej obývaných a zaľudnených oblastiach v porovnaní napr. s mestskými oblasťami [41, 43, 60]. V súvislosti s odhadom týchto metrick bolo vypracovaných nemalo prác. Napríklad Demetriou vo svojej práci zaoberajúcej sa odhadom týchto troch

metriek pre cestné siete na Cypre ukázal, že pozičná presnosť a úplnosť v pomenovaní ciest je významne vyššia pri cestách vyššej triedy v porovnaní s cestami nižšej triedy [29]. Mohammad a Mahmoud vo svojej práci považujú za hlavný problém kvality OSM dát heterogénnosť OSM datasetu z hľadiska úplnosti v porovnaní s referenčnou mapou [36].

2.3.1 Dátový model OSM

Na definíciu a reprezentáciu bodových, lineárnych a mnohoúhelníkových objektov a štruktúr spolu s ich atribútmi a vlastnosťami vo fyzickom svete slúži v OSM databáze trojica elementárnych dátových typov súhrnne zvaných elementy. Elementy sú základné komponenty tzv. koncepčného dátového modelu fyzického sveta.

Atribúty, vlastnosti a spôsoby použitia reprezentovaných objektov sú popísané tzv. značkami (*tag*), ktoré sú pridelené elementom. Tieto značky sú dvojice v tvare <kl'úč=hodnota>, napr. *landuse=forest*. Každý element môže mať jednu alebo viac značiek. Pridelovanie značiek nemá presne definovanú metódu a existujú len akési odporúčania alebo zaužívané postupy v značkovani používané komunitou. Samotný používateľ ale môže vytvoriť vlastné hodnoty značiek popisujúce špecifické atribúty popisovaných objektov. Zaužívané popisovanie značkami môže čitateľ dohľadať v [10]. Práve kvôli nie exaktne definovanej metodike a rôznorodým skúsenostiam prispievateľov vznikajú nepresnosti v značení vlastností objektov.

V dátovom modeli OSM sa uvažujú práve tieto tri elementy:

- **Uzol (Node)**

Používa sa na označenie objektov, ktoré je možno reprezentovať ako body. Zvyčajne sú používané na tvorbu elementu *cesta*. Každý uzol musí obsahovať súradnice popisujúce jeho polohu ako zemepisnú šírku a dĺžku a tiež musí mať unikátny *node id* (64-bitový integer), ktoré ho odlišuje od ostatných uzlov v databáze. Ďalším nepovinným prvkom definujúcim uzol je nadmorská výška. Ak je uzol použitý na reprezentáciu samostatného objektu napr. stromu tak by mal mať pridelenú zodpovedajúcu značku definujúci účel, vlastnosti a pod.

- **Cesta (Way)**

Je na druhú stranu používaná na značenie lineárnych štruktúr ako sú cesty, diaľnice, rieky tvoriace tzv. *otvorené cesty* alebo hranice objektov tvoriacich tzv. *uzavreté cesty* resp. *mnohouholníky (polygon)* alebo *plochy (area)*. Môžu byť tvorené ako usporiadaná kolekcia z 2 až 2000 uzlov. Opäť je unikátnosť medzi všetkými *cestami* zabezpečená používaním celých čísel (*way id*).

- **Vzt'ah (Relation)**

Definuje abstraktnejšie štruktúry a slúži na spájanie uzlov a *ciest* ako aj ďalších *vzt'ahov* do väčších celkov. Príkladom takéhoto celku je napríklad *vzt'ah* zvaný *multipolygon*, ktorý slúži na zaznamenávanie oblastí s dierami. Ako príklad oblasti s dierami môže slúžiť jazero vnútri ktorého sú ostrovy. Multipolygon sa používa taktiež na značenie prvkov resp. objektov pozostávajúcich z viac ako 2000 uzlov. Podobne ako v prípade uzlov a ciest majú aj *vzt'ahy* pridelené unikátne celé číslo (*id*) odlišujúce od seba všetky existujúce *vzt'ahy* v OSM databáze.

Detaily ohľadom elementov a s nimi súvisiacich atribútov používaných v OSM dátovom modeli môže čitateľ dohľadať v [9].

2.3.2 Prístup k dátam OSM

Všetky dáta OSM sú uložené v databáze, no na ich získanie slúži veľké množstvo API, webových rozhraní ako aj samotná stránka *openstreetmap.org*. Všetky geo-priestorové dáta sú exportované najčastejšie v XML (.osm) alebo GeoJSON (.json) formáte a obsahujú všetky elementy nachádzajúce sa v požadovanej oblasti záujmu.

Medzi populárne prostriedky získania dát z OSM patrí:

- **Planet.osm:** <https://planet.osm.org/>
- **Overpass API:** <https://overpass-api.de/index.html>
- **Overpass turbo:** <https://overpass-turbo.eu/>

Iné možnosti prístupu k OSM dátam sú popísané v [4].

Kapitola 3

Strojové učenie

Veľkému rozmachu strojového učenia v súčasnej dobe vd'áčíme hlavne zjednodušenej prístupnosti k informáciám a dátam predovšetkým pomocou internetu a rýchlej výpočtovej technike. Strojové učenie často ponúka efektívne riešenie problémov, ktoré nie sú ľahko alebo vôbec riešiteľné explicitným zostavením inštrukcii resp. naprogramovaním stroja na vykonávanie konkrétnej úlohy. Medzi takéto úlohy patrí veľké množstvo úloh v oblasti počítačového videnia ako je napríklad rozponávanie tvári, segmentácia obrazu v autonómnych vozidlách či už ako v našej práci riešime, klasifikácia povrchov a objektov v satelitných snímkach. Okrem aplikácii v počítačovom videní sa metódy strojového učenia efektívne a s úspechom aplikujú aj v množstve iných odborov a aplikáciach ako napríklad odporúčacích systémoch, pri spracovaní biomedicínskych alebo fyzikálnych dát. Všeobecne platí, že strojové učenie nájde uplatnenie všade tam kde je potrebné vo veľkom množstve dát nájsť vzorce a tieto použiť buď na automatizovanú predikciu alebo v nejakom rozhodovacom procese.

3.1 Teória strojového učenia

Formálne je strojové učenie založené na koncepte zvanom štatistická teória učenia, ktorá formalizuje konštrukciu odhadov funkcií, ktoré pri rastúcom počte dát konvergujú ku skutočnej odhadovanej funkcii. Menej formálne povedané je strojové učenie riešenie optimalizačnej úlohy kde sa navyše vyžaduje schopnosť generalizácie modelu vzhľadom k vopred nevideným dátam. Typicky sú algoritmy strojového učenia rozdelené do troch kategórií na základe prítomnosti *ground truth* v množine dát slúžiacej na učenie:

- **Učenie s učiteľom** (*supervised learning*)
 - Ide o veľmi v praxi častý prístup k učeniu. Základom je dostupnosť množiny dát resp. datasetu $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$, $n \in \mathbb{N}$ kde každá dvojica $(\mathbf{x}_i, \mathbf{y}_i)$, $i \in \hat{n}$ sa nazýva vzorka (*sample*) alebo aj označované dáta (*labeled data*) a je tvorená vektorom \mathbf{x}_i zvaným prediktor (*feature vector*) a vektorom \mathbf{y}_i tiež zvaným *ground truth* alebo cieľová premenná (*target variable*). Kvôli jednoduchosti budeme ďalej uvažovať \mathbf{y}_i ako skalár. Názov učenie s učiteľom pochádza práve z dostupnosti *ground truth* \mathbf{y}_i , ktoré slúžia ako učiteľ definujúc tak správny výstup. Množina \mathcal{D} teda reprezentuje n realizácií dvojice náhodných vektorov (\mathbf{X}, \mathbf{Y}) . Na základe týchto realizácií sa snažíme nájsť odhad \hat{f} (hypotézu) zobrazenia $f(\mathbf{X}) = \mathbf{Y}$ pričom chceme aby platilo

$$\hat{f} = \operatorname{argmin}_{f \in \mathbb{F}} L_n(f) \quad (3.1)$$

kde L_n je nejaká vhodne zvolená empirická riziková resp. účelová funkcia tvaru

$$L_n(f) = \frac{1}{n} \sum_{i=1}^n L(\hat{f}(x_i), y_i) \quad (3.2)$$

a \mathbb{F} je množina možných funkcií f zvaná aj priestor hypotéz. V praxi teda môže ísť často o zložitý optimalizačný problém. Medzi typy úloh, ktoré bývajú riešené takýmto prístupom patrí napríklad klasifikácia kedy obor hodnôt náhodnej veličiny Y je konečná množina tried. Daná odhadová funkcia \hat{f} pre nejakú klasifikačnú úlohu priradí každému vstupu x_i práve jednu hodnotu z konečnej množiny tried. Špeciálnym typom klasifikácie je tzv. sémantická segmentácia obrazu, kde je cieľom aplikovať klasifikáciu na úrovni pixelov obrázka teda priradiť triedu z konečnej množiny každému pixelu. Ďalšou často riešenou úlohou je tzv. regresia kedy obor hodnôt náhodného veličiny Y je nejaká podmnožina reálnych čísel \mathbb{R} .

- **Učenie bez učiteľa** (*unsupervised learning*)

- V tomto prípade máme prístup len k datasetu tvaru $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}, n \in \mathbb{N}$, ktorý teda neobsahuje *ground truth*. Učenie bez učiteľa sa aplikuje tam kde je potrebné nájsť vzorce resp. štruktúru v dátach. Úlohou, ktorá sa rieši často pomocou učenia bez učiteľa je klustrovanie (*clustering*). V podstate ide o klasifikáciu dát $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}, n \in \mathbb{N}$ do klustrov resp. skupín na základe vzorcov vypočítaných v dátach. Ďalším príkladom použitia učenia bez učiteľa je odhad a priori pravdepodobnostnej distribúcie dát alebo detekcia anomálií v dátach.

- **Hybridné učenie** (*semi-supervised learning*)

- Ide o prirodzené zlúčenie oboch typov učenia popísaných vyššie. V tomto prípade máme prístup k množine vstupných dát x_i ale len z časti prístup k dátam y_i . Dataset je teda tvaru $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_m, (\mathbf{x}_{m+1}, y_{m+1}), \dots, (\mathbf{x}_n, y_n)\}, m < n \in \mathbb{N}$. Typickým príkladom môže byť klasifikácia objektov v satelitných snímkach kde je práve zaobstaranie *ground truth* často veľmi náročná úloha.

Ďalšou paradigmou použitou v strojovom učení je tzv. *reinforcement learning*. Algoritmy resp. agenti založený na tomto prístupe interagujú s prostredím tak, že za každú vykonanú akciu získajú pozitívnu alebo negatívnu odmenu [40]. Daný agent sa musí na základe takéhoto cyklu vykonaných akcií a odmien naučiť operovať v danom prostredí.

3.1.1 Fázy učenia

Ešte pred nástupom aplikácie algoritmu strojového učenia musíme mať prístup k datasetu. V praxi sa daný dataset rozdelí na dve alebo tri časti zvané tréningový, validačný a testovací dataset. Ich význam je popísaný v ďalších odstavcoch.

3.1.1.1 Tréningovanie

Tréningom máme na mysli riešenie optimalizačnej úlohy definovanej v rovnici 3.1 kedy sa snažíme minimalizovať účelovú funkciu na základe dostupných dát teda v prípade učenia s učiteľom ide o dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}, n \in \mathbb{N}$. Pri riešení optimalizačnej úlohy nevyužíva celá množina dát ale len jej časť. Zbytok dát je takpovediac odložený stranou a slúži na určenie

vlastností modelu ako je generalizácia, na výber najlepšieho spomedzi dostupných modelov a tiež na určenie dodatočných hyperparametrov modelu.

3.1.1.2 Validácia

Táto fáza ma za úlohu určiť generalizačné vlastnosti modelu, teda schopnosť adaptovať sa na vopred nevidené dáta. Konkrétne v tejto fáze máme k dispozícii validačný dataset, ktorý vznikol ako časť tréningového datasetu. Okrem určenia vlastností modelu slúži fáza validácie na následné nájdenie optimálnych parametrov spojených s daným modelom zvaných aj hyperparametre, ktoré neboli určené tréningom resp. riešením optimalizačnej úlohy.

Pri tréningu algoritmu sa typicky odhaduje jeho tréningová chyba, resp. hodnota rizikovej funkcie získaného odhadu, ktorá slúži na určenie jeho optimality spomedzi ostatných odhadov. Okrem minimálnej tréningovej chyby sa od algoritmu požaduje aj čo možno najnižšia tzv. generalizačná chyba meraná na vopred nevidených dátach.

Charakteristiky, ktoré robia model optimálnym sú:

1. Schopnosť dosiahnuť čo možno najnižšiu tréningovú chybu
2. Schopnosť čo možno najviac zmenšiť rozdiel medzi tréningovou a generalizačnou chybou

S týmito dvoma vlastnosťami modelu súvisí tzv. pretrénovanie (*overfitting*) a podtrénovanie (*underfitting*). Podtrénovaný model vzniká vtedy ak algoritmus nie je schopný dostatočne dobre znížiť tréningovú chybu na tréningovom datasete. Na druhú stranu k pretrénovaniu dochádza v prípade keď je rozdiel medzi tréningovou a generalizačnou chybou príliš veľký.

Pretrénovaniu sa zabráňuje tzv. regularizáciou čo je akákoľvek modifikácia učiaceho sa algoritmu, ktorá má za cieľ zmenšiť generalizačnú ale nie tréningovú chybu. Môže ísť teda o zmenšenie priestoru hypotéz \mathbb{F} napríklad reštrikciou počtu parametrov, ktorý dané hypotézy parametrizuje čo bude mať za následok menej komplexný model. Ďalšou používanou regularizačnou metódou je pridanie regularizačného členu. Typicky ide o L2 alebo L1 normu váh \mathbf{w} parametrizujúcich danú hypotézu $\hat{f} = \hat{f}(\mathbf{w})$ do predpisu 3.2 čím vznikne

$$L_n^{reg}(f) = \frac{1}{n} \sum_{i=1}^n L(\hat{f}(x_i, \mathbf{w}), y_i) + \lambda \|\mathbf{w}\|_2^2 \quad (3.3)$$

V danom predpise 3.3 sa parameter $\lambda \geq 0$ nazýva *weight decay* pretože určuje v akej miere bude regularizačný člen prispievať do výslednej účelovej funkcie. Algoritmus bude v takomto nastavení preferovať hypotézy s nižšími veľkosťami váh \mathbf{w} čo vytvára daný regularizačný efekt.

Hyperparametre zahŕňajú všetky parametre použitého algoritmu, ktoré sa algoritmus procesom učenia nenaučí. Určenie optimálnych hyperparametrov zvané aj ladenie hyperparametrov je jednou s najdôležitejších úloh pri návrhu úspešného modelu. Je ich treba často určiť empiricky, na základe predchádzajúcich skúseností alebo vyskúšať rôzne kombinácie hyperparametrov a vybrať tie, ktoré prinášajú najlepšie výsledky vzhľadom k určeným metrikám. Medzi najjednoduchšie a najviac používané patrí tzv. *grid-search* metóda, ktorá je aplikáciou metódy hrubej sily. *Grid-search* v podstate skúša každú určenú sadu hyperparametrov z prehľadávanej oblasti priestoru hyperparametrov. Iným prístupom k prehľadávaniu priestoru hyperparametrov je *randomized search* teda akési znáhodnené prehľadávanie s vopred určenými intervalmi resp. hodnotami hyperparametrov a to tak, že sa skúšajú náhodné sady hyperparametrov. Existujú aj iné, sofistikovanejšie metódy založené na genetických a evolučných algoritmoch.

Tieto metódy na ladenie hyperparametrov sa následne kombinujú s tzv. *cross-validáciou*. V praxi sa používa tzv. *k-fold cross-validácia*, ktorá prebieha tak, že sa z daného datasetu $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}, n \in \mathbb{N}$ vyberie k neprekrývajúcich sa podmnožín pričom sa algoritmus trénuje na zjednotení $k - 1$ podmnožín a zostávajúca podmnožina sa použije na odhad generalizačnej chyby. Táto procedúra sa opakuje k -krát postupne na všetkých k podmnožinách pričom ako výsledná generalizačná chyba je brany priemer jednotlivých určených chýb. Toto slúži na presnejší odhad generalizačnej schopnosti algoritmu a teda aj na určenie optimálnych hyperparametrov spojených s modelom alebo výber najlepšieho spomedzi skúmaných modelov.

3.1.1.3 Testovanie

Táto fáza je poslednou a predpokladá prístup k testovaciemu datasetu. Teda časti dát, ktoré neboli použité ani na trénovanie algoritmu ale ani pri ladení hyperparametrov modelu pri validácii. Testovacia fáza pomáha určiť najlepší spomedzi skúmaných modelov $\hat{f}_1, \hat{f}_2, \dots, \hat{f}_k; k \in \mathbb{N}$ na základe odhadu generalizačných schopností modelov.

3.1.1.4 Metriky

V predchádzajúcich odstavcoch sme hovorili o trénovacej a generalizačnej chybe, ktoré sú samé o sebe metriky určujúce mieru neúspešnosti modelu pri riešení konkrétnej úlohy. Na ich kvantitatívne určenie ich ale musíme zdefinovať. Typicky teda definujeme mieru úspešnosti m s oborom hodnôt $[0, 1]$ a potom sa daná chyba *error* bude počítat' na základe vzťahu $error = 1 - m$. V nasledujúcich odstavcoch priblížime metriky používané pri porovnávaní klasifikačných modelov. Názvy metrick ponechávame bez slovenského prekladu pretože to považujeme za kontraproduktívne.

Confusion matrix reprezentuje kompaktný záznam výsledkov klasifikácie na základe kumulácie informácie o počte úspešných a neúspešných výsledkov predikcie modelu. Takýto záznam sa hodí pre jednoduchšiu vizuálnu inšpekciu kvality modelu. Samotná konfúzna matica teda nie je metrika a v prípade binárnej klasifikácie nadobúda tvar

	1	0	Skutočné
1	TP	FN	
0	FP	TN	
Predpovedané			

Číslo 1 indikuje pozitívnu triedu a 0 negatívnu. TP znamená počet *true positive* výsledkov, podobne TN je počet *true negative* výsledkov, FP počet *false positive* a FN *false negative* výsledkov. Rozšírenie pre klasifikáciu s viacerými triedami je priamočiare a preto ho nebudeme uvádzať.

Accuracy udáva podiel počtu správnych predpovedí klasifikátora ku počtu všetkých jeho predpovedí. V prípade binárnej klasifikácie je definovaná ako

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (3.4)$$

Výhodou tejto metriky je jej interpretovateľnosť no na druhú stranu prináša neuspokojivý odhad kvality klasifikátora pri nevyváženom datasete kedy nadhodnocuje kvalitu klasifikátora. Metriku *accuracy* je teda vhodné použiť ak sú distribúcie tried porovnateľné.

Recall (sensitivity) udáva percentuálne koľko relevantných vzoriek bolo správne klasifikovaných daným klasifikátorom inak povedané meria schopnosť klasifikátora detekovať relevantnú triedu. Jeho nízka hodnota indikuje veľký počet falošne negatívnych predpovedí. Daná metrika sa počíta na základe vzťahu:

$$recall = \frac{TP}{TP + FN} \quad (3.5)$$

Precision je metrika, ktorá udáva mieru dôveryhodnosti predpovede daným klasifikátorom. Nízka hodnota tejto metriky poukazuje na prevahu falošne pozitívnych predpovedí a je definovaná ako:

$$precision = \frac{TP}{TP + FP} \quad (3.6)$$

F1 score (dice coefficient) sa definuje ako harmonický priemer metrík *recall* a *precision*:

$$F_1 = 2 \frac{precision * recall}{precision + recall} = 2 \frac{TP}{2TP + FN + FP} \quad (3.7)$$

F1 score teda agreguje informáciu, ktorú získame z metrík *precision* a *recall* a poslúži tak ako lepšia miera kvality klasifikátora pri nevyváženom datasete.

Intersection over Union (jaccard index) tiež známy ako koeficient podobnosti meria podobnosť medzi *ground truth* a predpoveďou klasifikátora. Táto metrika je definovaná nasledovne:

$$IoU = \frac{TP}{TP + FN + FP} \quad (3.8)$$

Použitie tejto metriky býva časté v úlohách ako detekcia objektov a sémantická segmentácia pretože dáva lepší vzhľad o kvalite klasifikátora oproti napríklad metrike *accuracy*. Ďalej v práci budeme používať priemernú hodnotu tejto metriky značenú *mIoU* a počítanú ako

$$mIoU = \frac{1}{|C|} \sum_{i \in C} IoU_i \quad (3.9)$$

kde C značí množinu tried.

3.2 Skupinové učenie (ensemble learning)

Skupinové učenie (*ensemble learning*) je založené na myšlienke učenia niekoľkých modelov oddelene a následnom použití ich kombinovanej znalosti na vykonanie predikcie. Táto kombinovaná znalosť môže byť získaná ako majoritné hlasovanie, priemerovanie alebo spojenie viacerých modelov za seba (*stacking*).

Jednou skupinou algoritmov využívajúcich skupinové učenie sú tie, založené na rozhodovacích stromoch (*decision trees*). Takýto rozhodovací strom je neparametrický algoritmus formálne reprezentovaný grafom ale v podstate ide o sériu rozhodnutí s definovaným prahom (*threshold*) v každom vrchole (*node*). Podotkneme, že účelová funkcia takéhoto stromu nie je diferencovateľná kvôli diskkrétnej štruktúre stromu a preto sa ako štandardná procedúra učenia používajú metódy založené na *greedy* algoritmoch, ktoré iteratívne budujú daný strom vrchol po vrchole. Detaily tohto postupu vynecháme a zvedavého čitateľa odkážeme na [61].

Výhodou použitia rozhodovacích stromov je ich jednoduchosť a interpretovateľnosť no na druhú stranu majú sklon k pretrénovaniu a teda nedisponujú dostatočnou generalizačnou schopnosťou. V terminológii strojového učenia resp. štatistiky hovoríme o vysokom rozptyle (*variance*) a malom posune (*bias*) odhadu získanom rozhodovacím stromom. Konkrétne metódy skupinového učenia založené na rozhodovacích stromoch sú popísané v nasledujúcich odstavcoch.

Bagging je prvou možnosťou ako agregovať schopnosti jednotlivých rozhodovacích stromov a tým znížiť rozptyl v ich predikcii [51]. Táto metóda prebieha tak, že sa v prvom rade vytvorí množina datasetov náhodným vzorkovaním s opakovaním z celého tréningového datasetu metódou zvanou *bootstrapping*. Prevzorkované datasety môžu mať rovnaký alebo menší počet vzoriek oproti základnému tréningovému datasetu. Ďalej sa na každom datasete s tejto množiny natrénuje rozhodovací strom čím získame odhady \hat{f}_i . Nakoniec sa jednoduchým priemerovaním (v prípade regresie) získa odhad s menším rozptylom a teda daný algoritmus disponuje lepšou generalizačnou schopnosťou. Odhad získaný *bagging*-om je tvaru

$$\hat{f}_{bag} = \frac{1}{k} \sum_{i=1}^k \hat{f}_i \quad (3.10)$$

V prípade klasifikačného problému sa namiesto priemerovania zväčša používa majoritné hlasovanie. Samotný názov *bagging* pochádza zo slov *bootstrap aggregating*.

Náhodný les (Random forest) vylepšuje myšlienku *bagging*-u tak, že sa ešte navyše aplikuje reštrikcia na počet prediktorov a ich náhodný výber pri tvorení každého z rozhodovacích stromov pri tzv. *split*-och. Pri každej tvorbe optimálneho rozhodnutia (*split*-e) sa môže použiť len p z m náhodne vybraných prediktorov pričom $p \leq m$. Táto modifikácia v prípade náhodného lesa slúži na zníženie korelácie predikcii jednotlivých rozhodovacích stromov a teda prináša významnejší pokles rozptylu v porovnaní so samotnou metódou *bagging* [51]. *Bagging* sa dá teda chápať ako špeciálny prípad náhodného lesa kedy sa na tvorbu každého rozhodovacieho stromu použije vždy $p=m$ prediktorov.

Boosting je sám o sebe všeobecná metóda a môže byť aplikovaný na rozličné algoritmy zohrávajúce úlohu tzv. *weak learner*-a, ktorým môže byť napr. aj neurónová sieť. V našom prípade sa ale obmedzujeme na rozhodovacie stromy. Na rozdiel od náhodných lesov resp. *bagging*-u, je *boosting* založený na postupnej sekvenčnej tvorbe modelu. Myšlienka za *boosting*-om je nasledujúca. V prvom kroku sa natrénuje rozhodovací strom na tréningových dátach a následne sa pridá do finálneho modelu prenasobený faktorom λ (*shrinkage*) definujúcim akési spomalenie učenia. Následne sa vypočítajú reziduá $r_1(x) = y - \lambda \hat{f}_1(x)$ daného stromu \hat{f}_1 . Ďalší krok pozostáva z tréningovania nového stromu na reziduách $r_1(x)$ namiesto *ground truth* y čím získame nový odhad \hat{f}_2 vzhľadom ku ktorému aktualizujeme reziduá $r_2(x) = r_1(x) - \lambda \hat{f}_2(x)$, ktoré sa opäť použijú na tréningovanie nového stromu. Takýmto iteratívnym spôsobom sa postupuje ďalej kým sa nenatrénuje požadované množstvo rozhodovacích stromov, ktoré sú nakoniec pridané do finálneho modelu tvaru:

$$\hat{f}_{boost} = \sum_{i=1}^k \lambda \hat{f}_i \quad (3.11)$$

kde \hat{f}_i je i -ty rozhodovací strom a λ je spomínaný *shrinkage* parameter. Každý strom je teda tvorený tak, že má k dispozícii informáciu o tréningovaní resp. chybovosti predchádzajúcich rozhod-

vacích stromov na základe reziduii čo umožňuje iteratívne vylepšovanie (*boost*-ovanie) finálneho modelu \hat{f}_{boost} [51]. Podotknime, že *boosting* iteratívnym vylepšovaním finálneho modelu znižuje posun (*bias*) tohto modelu narozdiel od *bagging*-u a náhodného lesa, ktoré sú primárne určené na zníženie rozptylu [61].

Existujú rôzne varianty tohto algoritmu ako napríklad AdaBoost [37], ktorého hlavnou myšlienkou je adaptácia na chyby v predpovediach z predchádzajúcej iterácie a to tak, že sa priradia váhy jednotlivým vzorkám v datasete aby ťažko klasifikovateľné vzorky získali väčšiu váhu. Zároveň sa v modeli 3.11 aktualizujú váhy $\alpha_i(\lambda)$, ktoré nahradzujú člen λ , na základe chybovosti jednotlivých modelov \hat{f}_i . Čím presnejší je model \hat{f}_i pri predpovediach tým väčšia váha $\alpha_i(\lambda)$ mu je priradená. Táto adaptácia váh vzoriek dát ale aj jednotlivých odhadov je vykonávaná v každej iterácii z čoho pochádza názov *adaptive boosting*.

Ďalšou variantou *boosting*-u je *gradient boosting*, ktorý využíva generický prístup pomocou účelovej funkcie a jej optimalizáciu pomocou metód založených na výpočte jej gradientu. Konkrétne, reziduá $r(x)$ popísané vyššie sú počítané ako gradient tejto účelovej funkcie. Bližší popis algoritmu môže čitateľ dohľadať v [38]. Existuje niekoľko implementácií *gradient boosting*-u medzi, ktorými je napríklad LightGMB ale aj XGBoost [25], ktorú využívame aj v našej práci.

3.3 Ďalšie algoritmy

K-nearest neighbours je jedným z najjednoduchších neparаметrických algoritmov strojového učenia. V kontexte tohto algoritmu nemožno hovoriť o ničom ako trénovanie resp. optimalizácia a nedefinuje sa teda ani žiadna účelová funkcia. Základným princípom v prípade klasifikácie je pre každý vstup \mathbf{x}_{test} nájsť k najbližších susedov resp. bodov v priestore vstupných vektorov \mathbf{x}_i z trénovacieho datasetu. Miera blízkosti resp. vzdialenosť sa meria na základe vhodne zvolenej metriky ako napríklad euklidovská vzdialenosť a pod. Príslušná trieda y_{test} sa určí na základe väčšinového hlasovania spomedzi týchto k najbližších susedov. Takáto metóda teda vyžaduje prístup k celému trénovaciemu datasetu pretože ide o tzv. *learning-by-memorization* algoritmus a najväčšie prehl'ádavanie daného trénovacieho datasetu vzhľadom k najbližším susedom môže byť výpočetne neefektívne a preto sa v praxi využívajú efektívnejšie prístupy založené na stromových štruktúrach [61].

Logistická regresia slúži na riešenie binárnych klasifikačných úloh ale dá sa rozšíriť na riešenie klasifikačných úloh s viacerými triedami. Formálne patrí medzi skupinu generalizovaných lineárnych modelov GLM. Ide teda o parametrický model tvaru: $\mathbb{E}Y = f(\mathbf{w}^T \mathbf{x})$, $Y \sim F(\mu)$, $\mu = \mathbb{E}Y$. Ak predpokladáme binárnu klasifikáciu s dvoma triedami $y \in \{0, 1\}$ tak môže byť logistická regresia zapísaná v tvare:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \mathbf{x}); p(y = 0|\mathbf{x}, \mathbf{w}) = 1 - \sigma(\mathbf{w}^T \mathbf{x}) \quad (3.12)$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (3.13)$$

Nelineárna funkcia σ slúži ako aktivačná funkcia, resp. sa v terminológii GLM hovorí skôr o jej inverzii zvanej logit, ktorá sa označuje ako linkovacia funkcia. V prípade binárnej klasifikácie platí $Y \sim \text{Be}(p)$. Hodnoty váh \mathbf{w} sa v praxi hľadajú cez maximalizáciu vierohodnostnej funkcie resp. log-vierohodnostnej funkcie napríklad pomocou metódy *gradient descent*.

Support vector machine (SVM) je algoritmus založený na myšlienke lineárnej separácie vzoriek datasetu pomocou nadroviny [27]. Najbližšie body k tejto nadrovine sú tzv. *support* vektory. Algoritmus sa snaží nájsť danú nadrovinu tak aby maximalizoval vzdialenosť tejto oddeľujúcej nadroviny od *support* vektorov. Kľúčovou myšlienkou v prípade SVM je možnosť vyjadriť lineárnu funkciu používanú SVM ako skalárny súčin medzi vzorkami nasledovne:

$$\mathbf{w}^T \mathbf{x} + b = b + \sum_{i=1}^m \alpha_i (\mathbf{x}^T \mathbf{x}_i) \quad (3.14)$$

kde \mathbf{x}_i je i -ty prediktor a α definuje vektor koeficientov. Toto nám ďalej umožňuje nahradiť skalárny súčin za funkciu

$$k(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x})^T \phi(\mathbf{x}_i) \quad (3.15)$$

zvanú *kernel* pričom funkcia ϕ zobrazuje pôvodný *feature* vektor do viac-dimenzionálneho priestoru aby bolo možné lineárne separovať aj také dáta, ktoré v nie sú v pôvodnom priestore lineárne separovateľné. Získame teda

$$f(\mathbf{x}) = b + \sum_{i=1}^m \alpha_i k(\mathbf{x}, \mathbf{x}_i) \quad (3.16)$$

pričom daná funkcia je lineárna v priestore do ktorého zobrazuje ale nelineárna v pôvodnom priestore ako funkcia \mathbf{x} [40]. V literatúre sa takýto prístup nazýva *kernel trick* pričom jeho využitie prináša výhody ako možnosť trénovať modely, ktoré sú nelineárnou funkciou x ale aj vzhľadom k tomu, že odpadá nutnosť implementácie zobrazenia k ako explicitnej aplikácie skalárneho súčinu $\phi(\mathbf{x})^T \phi(\mathbf{x}_i)$ ale namiesto toho môže byť daný skalárny súčin ekvivalentný nelineárnej ale výpočtne menej náročnej operácii vzhľadom k \mathbf{x} [40].

3.4 Neurónové siete

Neurónová sieť sa dá chápať ako parametrický model zostavený z množstva parametrických funkcií hrajúcich rolu akýchsi výpočetných jednotiek zvaných neuróny. Takéto neuróny sa často definujú ako nasledujúca funkcia

$$y(\mathbf{x}) = \phi(\mathbf{w}^T \mathbf{x} + b) \quad (3.17)$$

Pričom ϕ je istá nelineárna funkcia nazývaná aktivačná funkcia. Práve nelineárny charakter aktivačnej funkcie umožňuje využitie skupiny neurónov na učenie komplexných nelineárnych funkcií. V praxi sa používajú rôzne aktivačné funkcie ako napríklad ReLU definovanej ako $ReLU(x) = \max(0, x)$ ďalej $\tanh(x)$, alebo sigmoid 3.13. Parametre \mathbf{w} a b sa nazývajú váhy a posun (bias). Tieto váhy \mathbf{w} a posun b sú odhadované procesom učenia neurónovej siete. V rámci neuronových sietí sú neuróny zoskupované do vrstiev, čo umožňuje ich efektívnejšiu reprezentáciu.

Najzákladnejším typom neurónovej siete, ktorý poznáme je tzv. *feed-forward* sieť kedy sú dané neuróny zostavené tak, že výpočet alebo tok informácií prebieha len v jednom smere. Príkladom takéhoto typu siete môže byť viacvrstvový *perceptron* (*multilayer-perceptron*) [40] pozostávajúci len z tzv. plne spojených vrstiev (*fully-connected layers*) kde je každý neurón z predchádzajúcej vrstvy spojený s každým neurónom nasledujúcej vrstvy. V takejto i -tej plne spojenej vrstve

možno zapísať výstup k -teho neurónu v i -tej vrstve ako:

$$z_k^i = \phi^i\left(\sum_{j=1}^{K_{i-1}} w_{jk}^i z_j^{i-1} + b_k^i\right) \quad (3.18)$$

V danom vzťahu označuje horný index číslo vrstvy. Vzťah 3.18 v skratke hovorí, že k -ty neurón v i -tej vrstve je definovaný ako aplikácia aktivačnej funkcie ϕ^i na lineárnu kombináciu všetkých K_{i-1} neurónov v $i-1$ vrstve. Tejto lineárnej kombinácii sa hovorí aj pre-aktivácia.

Všeobecne sieť pozostáva zo vstupnej vrstvy, skrytých vrstiev, ktorých môže byť teoreticky neobmedzene mnoho a z výstupnej vrstvy.

Proces učenia neurónovej siete pozostáva z riešenia problému 3.1. Nakoľko je model parametrický tak sa dá optimalizácia preformulovať na hľadanie optimálnych hodnôt váh w a posunov b jednotlivých neurónov tvoriacich sieť. Samotné učenie pozostáva z dvoch krokov:

- *Forward pass*
 - V tejto fáze dochádza k určeniu odhadu resp. výstupu \hat{y} na základe vstupu \mathbf{x} neurónovou sieťou a následná evaluácia odhadu vzhľadom k vhodnej účelovej funkcii. Takto sa získa informácia o chybovosti modelu. Typicky používanou účelovou funkciou býva tzv. *cross-entropy* daná vzťahom:

$$CE(y_i, \hat{f}) = - \sum_{m \in C} y_i^m \log(\hat{f}^m(\mathbf{x}_i)) \quad (3.19)$$

kde C definuje množinu tried, y_i^m určuje skutočnú pravdepodobnosť m -tej triedy v i -tej vzorke dát a $\hat{f}^m(\mathbf{x}_i)$ je odhad tejto pravdepodobnosti daný modelom.

- *Backward pass*
 - Táto fáza má na starosti propagáciu informácie o chybe modelu stanovenej na základe hodnoty účelovej funkcie a príslušnú aktualizáciu parametrov siete θ zahŕňajúcich všetky váhy a biasy za účelom zmenšenia tejto chyby. Táto aktualizácia sa typicky vykonáva na základe negatívneho príspevku gradientu účelovej funkcie vzhľadom k parametrom θ z čoho plynie aj názov takejto metódy a síce *gradient descent* [40]. Tento algoritmus nadobúda tvar:

$$\theta^{i+1} = \theta^i - \eta \nabla_{\theta} L(\theta^i) \quad (3.20)$$

Parameter η sa nazýva *learning rate* a ide o hyperparameter siete ktorý treba vhodne nastaviť.

Existujú rôzne varianty algoritmu *gradient descent* ako napr. *stochastic gradient descent* kedy dochádza k výpočtu účelovej funkcie (*forward pass*) a aktualizáciám parametrov (*backward pass*) na základe menších množín dát, náhodne navzorkovaných z tréningového datasetu, zvaných *mini-batch*-e. Veľkosť tejto podmnožiny sa nazýva *batch_size* a ide o hyperparameter siete. Jedna iterácia celou množinou dát sa obvykle nazýva epocha.

Ďalšie vylepšenia vzhľadom ku *gradient descent*-u zahŕňajú použitie konceptu hybnosti (*momentum*) pri aktualizácii parametrov siete alebo metódy založené na automatickej adaptácii hyperparametra *learning rate* pre každý parameter siete ako je vykonané v algoritmoch AdaGrad, Ada-Delta, RMSProp alebo Adam [53]. Možnosť použiť algoritmus ako *gradient descent* vyžaduje efektívny výpočet gradientov čo je v prípade neurónových sietí zabezpečené použitím algoritmu *back-propagation* [72]. Bližšie informácie o týchto algoritmoch sú popísané v [40].

Regularizácia bola popísaná všeobecne pre algoritmy strojového učenia v sekcii 3.1.1.2. V kontexte neurónových sietí môže zahŕňať regularizácia výber menej komplexnej siete obsahujúcej menej trénovateľných parametrov, ďalej použitie regularizačného člena v účelovej funkcii podľa vzťahu 3.3. Iným spôsobom ako sa vyhnúť pretrénovaniu siete môže byť zväčšenie trénovacieho datasetu. Takéto zväčšenie môže byť zabezpečené umelým pozmenením dát (augmentácia). V prípade obrazových dát môže ísť o rôzne transformácie ako rotácia obrázka, priblíženie ale aj zmena jas a iné. Jednou z najjednoduchších a často používaných regularizačných metód je skoré zastavenie (*early stopping*) procesu učenia vtedy keď sieť začne vykazovať zväčšovanie generalizačnej chyby [40].

Ďalšou možnosťou regularizácie špecifickou pre neurónové siete je použitie tzv. *dropout*-u. Táto technika funguje tak, že sa počas trénovania siete náhodne s pravdepodobnosťou p vynuluje príspevok resp. všetky vychádzajúce spojenia neurónu v danej vrstve siete. Bolo ukázané, že takéto náhodné vynulovanie prispieva významne k zníženiu efektu pretrénovania [45]. V prípade konvolučných sietí rozobratých v ďalšej sekcii sa používa priestorový *dropout*, ktorý vynuluje s pravdepodobnosťou p celú tzv. *feature map*-u generovanú konvolučnou vrstvou. Takáto modifikácia vychádza z možnosti, že príslušné pixely môžu vykazovať priestorovú koreláciu a teda jednoduchý *dropout* nebude mať regularizačný efekt [78]. Často používanou metódou na zlepšenie učenia je tzv. *batch* normalizácia [50], ktorá vnáša do modelu šum čo má za následok regularizáciu siete.

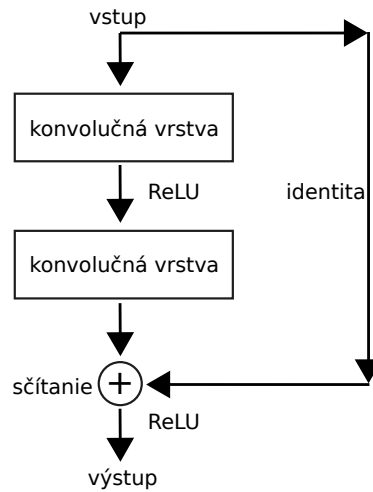
3.4.1 Konvolučné neurónové siete

Konvolučná neurónová sieť je model, obsahujúci aspoň jednu konvolučnú vrstvu, ktorá zohráva úlohu detekcie lokálnych vzorov a štruktúr všeobecne v nejakom signále, ktorým môže byť napríklad obrázok. Konvolučná neurónová sieť je teda okrem iného vhodná pri riešení problémov v počítačovom videní a spracovaní a rozpoznávaní obrazu pretože dokáže využiť priestorovú informáciu zakódovanú v rozložení pixelov obrázka. V prípade použitia konvolúcie na obrazové dáta s jedným spektrálnym kanálom používame dvojrozmernú, diskretnú konvolúciu tvaru:

$$(V * J)(i, j) = \sum_{m=1}^m \sum_{n=1}^n V(i-m, j-n)J(m, n) \quad (3.21)$$

kde V zohráva úlohu vstupu a J je tzv. jadro (*kernel*). Hodnoty jadra J sa v konvolučnej vrstve používajú ako váhy teda zohrávajú úlohu optimalizovateľných parametrov siete. Je dôležité podotknúť, že význam konvolučných vrstiev je predovšetkým v tom, že sú schopné procesom učenia určiť vhodné prediktory (*feature extraction*) a tak užívateľ a pozbavujú nutnosti samostatne definovať vhodné prediktory na riešenie konkrétnej úlohy.

Hyperparametrami definujúcimi vlastnosti konvolučnej vrstvy sú predovšetkým veľkosť jadra (*kernel size*), ktoré ovplyvňuje efektívnu oblasť vstupu na ktorú je aplikovaná konvolúcia. Inak povedané neuróny z $i-1$ vrstvy nie sú napojené na každý neurón i -tej konvolučnej vrstvy ako to bolo v plne prepojených vrstvách ale namiesto toho je každý neurón spojený len s časťou príslušných neurónov určenou práve veľkosťou daného konvolučného jadra. Toto obmedzenie spojení neurónov sa nazýva *local receptive field* a umožňuje detekciu väčších detailov a lokálnych vzorov. Ďalej je významným hyperparametrom počet samotných konvolučných jadier pričom každé jadro je zodpovedné za konkrétnu *feature map*u. Okrem toho pri definovaní konvolučnej vrstvy je potrebné určiť tzv. *stride* čo je krok aplikovaný konvolučným jadrom na vstupný obrázok a nakoniec tzv. *padding*, ktorý zabraňuje zmenšovaniu rozmerov vstupného obrázka.



Obr. 3.1: Znázornenie reziduálneho bloku využívaného v architektúre siete ResNet. V praxi sa namiesto identity používa napríklad 1×1 konvolučná vrstva zabezpečujúca zhodnosť počtu konvolučných jadier vzhľadom k tomu, že nasleduje operácia sčítania po prvkoch. Časť označená ako konvolučná vrstva často zahŕňa aj *pooling* vrstvy a *batch* normalizáciu.

Konvolučné vrstvy sú často používané spolu s tzv. *pooling* vrstvami zabezpečujúcimi zmenšovanie priestorových rozmerov obrázku resp. vytvorených *feature* máp a tak efektívnejšiu lokalizáciu relevantných štruktúr a obrazcov v obrázku ale aj na zníženie výpočetej náročnosti. Používané sú napríklad *max pooling* vrstvy alebo *average pooling* vrstvy [61].

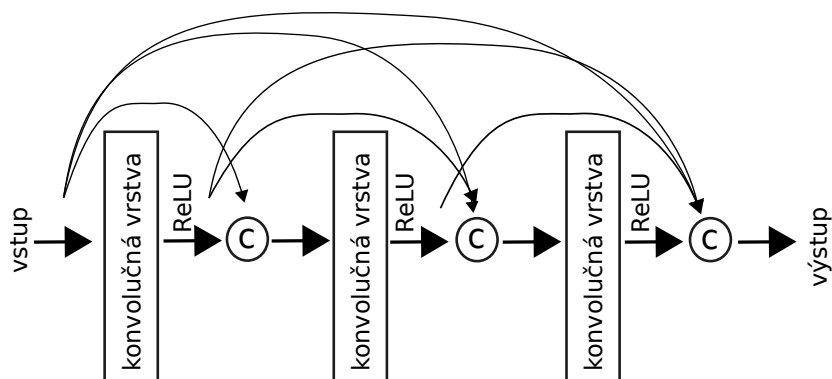
ResNet je *feed-forward* konvolučná neurónová sieť, ktorá sa bola navrhnutá za účelom zlepšenia vlastností hlbokých modelov, obsahujúcich veľké množstvo vrstiev, ktoré vykazujú efekt zvaný *vanishing gradient* [40], ktorý znemožňuje efektívnu propagáciu informácie o chybe siete predovšetkým do predných vrstiev. Základným stavebným kameňom siete ResNet je tzv. reziduálny blok formálne zapísany ako:

$$x^l = F^l(x^{l-1}) + x^{l-1} \quad (3.22)$$

kde x^l reprezentuje výstup l -tej vrstvy a F^l označuje klasické nelineárne zobrazenie aplikované na výstup x^{l-1} $l - 1$ vrstvy. Takáto modifikácia spočíva v pridaní identity $+x^{l-1}$, zvannej *skip connection*, umožňuje kvázi preskočiť problematické úseky v priestore parametrov a tak uľahčiť propagáciu informácie o chybe modelu pomocou gradientov do predných vrstiev siete. Autori tejto siete vo svojej práci [42] empiricky ukázali, že pridanie reziduálnych blokov skutočne pomáha riešiť *vanishing gradient* problém a umožňuje tak použitie hlbokých sietí modelujúcich komplexnejšie funkcie. [61]. Schématický zobrazenie reziduálneho bloku je na obrázku 3.1.

DenseNet model siete ďalej rozširuje myšlienku *skip connections* a to tak, že obsahuje tzv. hustý blok (*dense block*) vrstiev kde je *skip connection* použité z každej predchádzajúcej vrstvy do každej nasledujúcej vrstvy v danom bloku. Namiesto sčítania *feature* máp ako v sieti ResNet sa v modeli DenseNet aplikuje ukladanie za seba (*concatenation*) čo je formálne definované ako:

$$x^l = F^l(x^{l-1}, x^l, \dots, x^2, x^1) \quad (3.23)$$



Obr. 3.2: Znáznorenie *dense* bloku s tromi konvolučnými vrstvami. Takéto vrstvy často obsahujú aj *pooling* vrstvy a *batch* normalizáciu. Na rozdiel od reziduálneho bloku tento blok prepája každú predchádzajúcu konvolučnú vrstvu s každou nasledujúcou v rámci daného *dense* bloku a toto spojenie je vykonané operáciou *concatenation* čo má za následok zväčšovanie počtu konvolučných jadier po danej operácii.

Takéto bloky často obsahujú menšie množstvo parametrov v porovnaní s tradičnou konvolučnou sieťou nakoľko takéto husté prepojenie zabezpečuje nepotrebnosť učenia nadbytočných *feature* máp [47]. Schématické znázornenie *dense* bloku je na obrázku 3.2.

Xception predstavuje „extrémnu“ verziu modelu Inception. Tento model, na rozdiel od klasického prístupu kedy každá konvolučná vrstva extrahuje informáciu z predchádzajúcej za účelom detekcie štruktúr v obrázkoch, aplikuje viacero paralelných transformácií ako konvolúcia a *pooling* a pod. za účelom neskôr tieto extrahované *feature* mapy spojiť a nechať model vybrať relevantnú informáciu. Model Xception pridáva niekoľko vylepšení v podobe použitia tzv. *depthwise separable convolution* [61] ale aj zmeny poradia niektorých transformácií oproti pôvodnému modelu Inception. Bližšie detaily môžu byť opäť dohľadane v originálnej práci F. Cholleta [26].

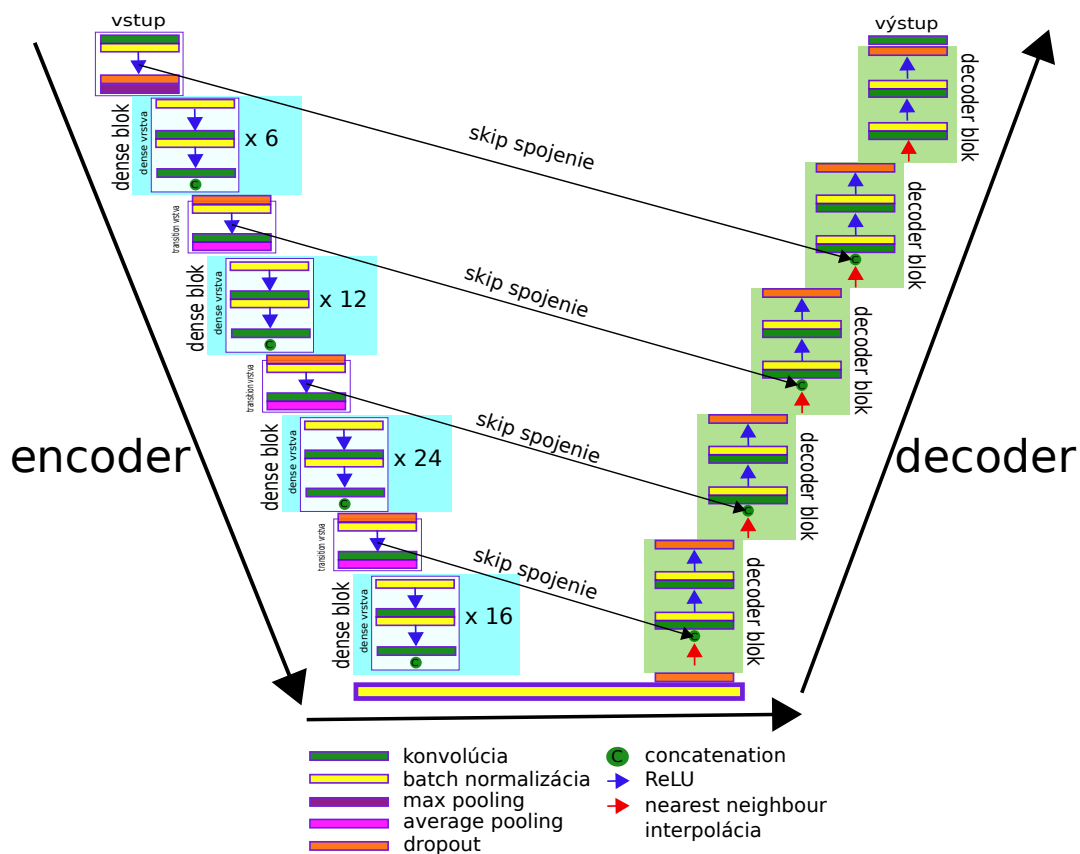
3.4.1.1 Plne konvolučné siete

Typicky sú konvolučné vrstvy v architektúrach neurónových sietí na klasifikáciu obrázkov nasledované plne spojenými vrstvami zabezpečujúcimi finálnu klasifikáciu. V prípade sémantickej segmentácie je ale takáto strata priestorovej informácie o pozícii vzorov (*features*) v pôvodnom obrázku nežiaduca. Okrem toho je potrebné v prípade sémantickej segmentácie získať na výstupe segmentačnú mapu rovnakých rozmerov ako mal pôvodný obrázok. Naivný prístup k riešeniu sémantickej segmentácie spočíva v aplikácii po sebe idúcich konvolučných vrstiev bez znižovania priestorových rozmerov *feature* máp pričom takto získame finálnu predikciu rovnakých rozmerov ako mal vstupný obrázok. Takáto metóda by ale bola výpočetne náročná nakoľko by každá *feature* mapa mala rovnaké priestorové rozlíšenie ako pôvodný obrázok. Efektívnejší prístup spočíva z použitia dvoch fáz tzv. *down-sampling* a *up-sampling path*. *Down-sampling path*, nazývaný aj *encoder*, pozostáva spravidla z aplikácie konvolučných vrstiev a *pooling* vrstiev s cieľom detekovať relevantné vzorce a štruktúry a zároveň znižovať rozmery vstupu čo má za následok menej výpočetne náročné tréningovanie siete. *Up-sampling path*, zvaný aj *decoder*, ďalej zabezpečuje spätnú lokalizáciu štruktúr a získanie pôvodného rozlíšenia obrázka aplikáciou bilinéarnej resp. *nearest neighbour* interpolácie nasledovanej konvolučnými vrstvami poprípade použitím tzv. transponovaných konvolúcií [61] alebo *unpooling*-u. Ďalším vylepšením vhl'adom k získaniu jemnejšej seg-

mentačnej mapy je použitie *skip connections* z *down-sampling* do *up-sampling* časti siete. Príkladom takejto plne konvolučnej siete je architektúra FCN [76]. Autori v danej práci nahradzujú plne prepojené vrstvy na klasifikáciu za ich ekvivalentnú reprezentáciu pomocou 1×1 konvolúcie. Základná architektúra FCN-32s využíva len jeden *up-sampling* krok pomocou bilineárnej interpolácie. Pričom ďalšie modifikácie FCN-16s a FCN-8s pridávajú niekoľko ďalších *up-sampling* krokov a taktiež využívajú *skip connections* na získanie jemnejšej segmentačnej masky čo sa týka hraníc objektov. Ďalšími príkladmi plne konvolučných neurónových sietí je napríklad architektúra SegNet [20] alebo architektúra DeepLab [24].

U-Net je architektúra symmetrickej plne konvolučnej neurónovej siete založená na už spomínanej architektúre FCN pôvodne vyvinutá na sémantickú segmentáciu bio-medicínskych obrázkov [71]. Autori v modeli U-Net používajú rovnaký počet *up-sampling* krokov ako *down-sampling* krokov pričom *up-sampling* kroky využívajú na zväčšenie priestorového rozlíšenia transponovanú konvolúciu s trénovateľnými parametrami namiesto fixnej bilineárnej interpolácie ako tomu bolo v prípade FCN. V modeli U-Net sú taktiež využité *skip connections* pričom na dané *feature* mapy je aplikovaná operácia *concatenation* namiesto sčítavania ako tomu bolo v modeli FCN.

V našej práci budeme neskôr experimentovať práve s modelom U-Net nakoľko ide o používaný model a ako bolo ukázané v [46] ide o vhodný model na sémantickú segmentáciu v satelitných snímkach. V danom modeli použijeme namiesto transponovanej konvolúcie fixnú interpoláciu typu *nearest neighbour* nasledovanú konvolúciami. Ďalšiu modifikáciu prinesieme experimentami s priestorovými *dropout* vrstvami. Schématické znázornenie neskôr použitej architektúry U-Net s *encoder*-om DenseNet 121 a aj s prípadne použitými priestorovými *dropout* vrstvami je zobrazené na obrázku 3.3. Parameter p definujúci pravdepodobnosť *dropout*-u má hodnotu 0.1 v prvej takejto vrstve od vstupu pričom hodnota tohto parametra stúpa o hodnotu 0.1 až kým nedosiahne 0.5 v *dropout* vrstve na začiatku *decoder*-a kde opäť začne klesať od 0.5 až na hodnotu 0.1 v poslednej *dropout* vrstve. Takýto postupný rast a pokles hodnoty parametra p súvisí priamo úmerne s počtom *feature* máp v danej časti siete.



Obr. 3.3: Na danom obrázku môžeme vidieť schématické znázornenie plne konvolučnej neurónovej siete U-Net využívanej v našej práci. *Encoder*-om je v tomto prípade sieť DenseNet 121 pozostávajúca zo štyroch *dense* blokov nasledovaných tzv. *transition* blokmi. Každý *dense* blok sa ešte skladá z niekoľkých *dense* vrstiev pričom ich počet je určený označením „ x {počet *dense* vrstiev}“. Tieto *dense* vrstvy v rámci daného *dense* bloku obsahujú husté prepojenia medzi sebou navzájom podobne ako je to na obrázku 3.2. Zmenšovanie priestorovej rozlíšenia vždy na polovicu predchádzajúceho rozmeru je zabezpečené v *encoder*-y prvou konvolučnou vrstvou a štyrmi *pooling* vrstvami. *Decoder* pozostáva z piatich tzv. *decoder* blokov kde sa okrem iného vykonáva *up-sampling* s faktorom 2 pomocou interpolácie typu *nearest neighbour*. Architektúra U-Net taktiež využíva *skip* prepojenie medzi *encoder*-om a *decoder*-om čím zabezpečuje lepší tok informácie o pozícii jednotlivých vzorcov a štruktúr v pôvodnom obrázku čím získame jemnejšiu segmentačnú masku predovšetkým vzhľadom k hraniciam objektov.

Kapitola 4

Spracovanie dát a príprava datasetov

Najdôležitejšou súčasťou úspešnej aplikácie metód strojového učenia s učiteľom je kvalitný a čo možno rozsiahly dataset. Preto v tejto kapitole opíšeme metodiku spracovania obrazových a geo-priestorových dát zo Sentinelu-2 a OSM do datasetov pre klasifikačnú a následne segmentačnú úlohu.

Metodika tvorby oboch typov datasetov bola spočiatku podobná a dá sa zhrnúť do nasledujúcich troch krokov:

1. Získanie dlaždíc Sentinelu-2 Level-2A (BOA) z danej oblasti záujmu v danom časovom rozmedzí

Kroku 1 predchádzala implementácia jednoduchého modulu na sťahovanie a spracovanie obrazových dát z misie Sentinel 2 pomocou poskytovaného API v jazyku Python 3.8. Na spracovanie dát v rastrovom formáte sme v práci prevažne využívali modul rasterio [75].

Ako oblasť záujmu na tvorbu datasetov bolo vybrané územie v zóne 33U v *Military Grid Reference System* (MGRS) resp. 33N v UTM zahŕňajúcu celú Česku republiku, južnú časť Poľska, východnú časť Nemecka, severnú časť Rakúska, Maďarska a západnú časť Slovenska. Toto územie je reprezentované 25 dlaždicami zo Sentinelu-2 a nachádza sa zhruba v oblasti definovanej štvoricou bodov:

- 52.3°N, 12.0°E
- 52.3°N, 19.5°E
- 47.7°N, 12.3°E
- 47.7°N, 19.1°E

Každá snímka bola získaná v štyroch časových okamihoch teda ako 4 dlaždice tvoriace 4-člennú časovú radu zaberajúca 3-mesačné obdobie letných mesiacov presnejšie 1.6.2020 - 31.8.2020. Treba ale podotknúť, že jednotlivé časové rady sa môžu pre rôzne snímky líšiť v čase kedy boli nasnímané jednotlivé dlaždice tvoriace túto radu.

Dokopy bolo teda celkovo využitých 100 dlaždíc. Letné mesiace boli vybrané kvôli väčšiemu množstvu bezoblačných dní. Relatívne krátke časové rozmedzie zaručí len malú variáciu v spektrálnych vlastnostiach jednotlivých snímok v priebehu času. Bližší popis analýzy zmien spektrálnych charakteristík v snímkach v priebehu času bude zhrnutý v sekcii 4.2.2 o spektrálnych charakteristikách snímok použitých na tvorbu datasetu pre segmentáciu. Na spracovanie, analýzu a tvorbu datasetov bolo vybrané spracovanie Level-2A predovšetkým

kvôli atmosferickej korekcii v odrazivosti BOA popísanej v 1. kapitole ale aj kvôli prístupu ku rastu obsahujúcej klasifikáciu scény.

V oboch prípadoch datasetov bolo použitých len 9 z 13 spektrálnych pásiem v 20 m rozlíšení nachádzajúcich sa v produkte Level-2A. Spektrálne pásma B02, B03, B04 boli prevzorkované z pôvodného 10 m rozlíšenia na 20 m rozlíšenie, ostatné použité pásma boli v 20 m rozlíšení už pôvodne. Okrem týchto 9 spektrálnych pásiem bolo pre každú dlaždicu vygenerovaných 5 spektrálnych indexov v 20 m rozlíšení slúžiacich ako prídavné spektrálne pásma. Jednotlivé použité spektrálne pásma boli tieto:

- B02
- B03
- B04
- B05
- B06
- B07
- B8A
- B11
- B12
- NDVI [48]
- NDWI [83]
- NDMI [39]
- NDBI [86]
- NBRI [?]

2. Získanie vektorových dát z OSM a ich kompaktné uloženie do *shapefile*

V 2. kroku prípravy datasetov bolo treba zaobstarať potrebné vektorové dáta z OSM, slúžiace ako *ground truth* pre oblasť záujmu popísanú vyššie. Na zadenovanie *ground truth* sme vytvorili niekoľko tried popisujúcich daný objekt alebo povrch. Každá trieda je reprezentovaná príslušnými značkami alebo množinou značiek z OSM. Na samotné stiahnutie dát z OSM, teda elementov s príslušnými značkami, v našej práci používame python-ovský modul OSMnx [23] postavený na Overpass API, ktorý slúži predovšetkým na efektívnu analýzu cestných sietí ale ponúka aj jednoduché aplikačné rozhranie k získavaniu požadovaných elementov z OSM na základe užívateľom zadenovanej oblasti záujmu a špecifických hodnôt značiek.

Objekty a povrchy patriace do jednotlivých tried museli výhradne tvoriť polygónové alebo multipolygónové (množina polygónov) štruktúry a nie body a čiary, ktoré majú nulovú plochu a teda pre ne rastrová reprezentácia nemá zmysel. Keďže ale cesty pre vozidlá sú v OSM takmer výhradne reprezentované elementom *cesta* (*way*) nebolo ich možné zahrnúť do samostatnej triedy definujúcej cestu pre vozidlá. Viac o jednotlivých triedach sa čitateľ dozvie v sekciách 4.1 a 4.2 o spracovaní dát na tvorbu datasetov pre klasifikáciu a sémantickú segmentáciu.

Ako bolo spomenuté na prípravu datasetov boli použité spektrálne pásma v 20 m rozlíšení čo vytváralo limitáciu na zadenovanie vhodných tried objektov a povrchov. Zabezpečenie tejto požiadavky bolo dosiahnuté tak, že každý objekt musel mať *bounding box*, teda obdĺžnik ohraničujúci polygón tvoriaci objekt, s oboma stranami väčšími ako 25 m.

Najpodstatnejšou limitáciou pri tvorbe datasetov sa ukázalo sťahovanie dát z OSM kvôli odkladom serveru a častému odpojeniu skriptu od servera spôsobenému limitáciou servera na počet požiadaviek za deň. Keďže bola oblasť záujmu relatívne rozsiahla tak bola rozdelená na menšie časti pri vytváraní požiadavok na server, ale ani toto nepomohlo k častým odpojeniam.

Získané objekty boli následne kompaktné uložené spolu s prísluchajúcim popisom triedy (*label*) v geometrii polygon vo formáte *shapefile* pre neskoršie jednoduché spracovanie.

3. Ďalšie spracovanie závisle na type výsledného datasetu (klasifikačná a segmentačná úloha)

Posledný krok spracovania bol založený na agregácii dát získaných v predchádzajúcich dvoch krokoch.

Všetky vzniknuté obrázky resp. segmentačné masky sú od tohto kroku reprezentované ako viacrozmerné numpy pole (*numpy ndarray*) pretože už ďalej nie je potreba mať informáciu o polohách pixelov vzhľadom k nejakému koordinačnému systému.

Bližšie popísaný 3. krok tvorby jednotlivých datasetov, spolu s ich charakteristikou bude uvedený v nasledujúcich sekciách venovaných datasetu pre klasifikáciu a sémantickú segmentáciu.

4.1 Dataset pre klasifikáciu

V poslednom kroku sa dataset pre klasifikáciu tvoril tak, že sa z každej snímky v rastrovom formáte (dlaždice) vyrezávali menšie podobrázky obdĺžnikového tvaru (*bounding box*) ohraničujúce polygóny. Tieto polygóny reprezentovali práve nejaký objekt alebo povrch pričom boli získané spolu s informáciou o *ground truth* z OSM a nakoniec uložené lokálne v shapefile. Všetky pixely vo vyrezanom obrázku obdĺžnikového tvaru, ktoré nepatrili danému objektu, teda charakterizujúce pozadie, boli označené maskou a ďalej neboli použité napr. na tvorbu spektrálnych prediktorov.

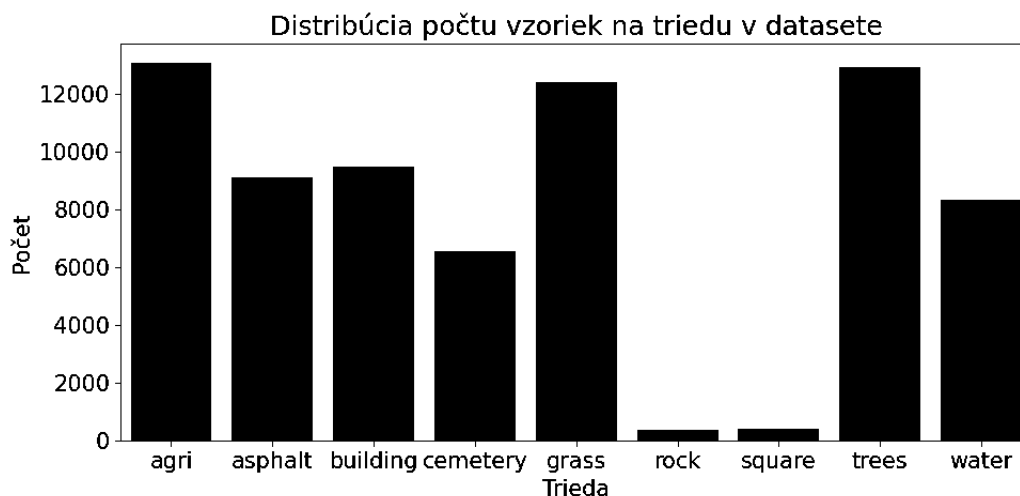
Vyrezávaniu predchádzalo generovanie spektrálnych indexov, ktoré boli použité ako dodatočné spektrálne pásma. Spektrálne indexy majú rozsah hodnôt $[-1, 1]$. Navyiac sa pri generovaní týchto indexov označili všetky *nodata* pixely maskou -2 . Každý takto dodatočne vygenerovaný spektrálny index bol nakoniec ešte afinne transformovaný z intervalu $[-1, 1]$ do intervalu $[0, 10000]$ kde nula slúži na reprezentáciu vylúčených (*nodata*) pixelov. Prakticky to znamená, že všetky reprezentatívne pixely majú hodnoty v intervale $[3333, 10000]$. Takto vybraný interval zabezpečil aby hodnoty pixelov všetkých spektrálnych pásiem mali zhruba rovnakú škálu pričom sa nestratí žiadna informácia obsiahnutá v indexoch. To v prvom rade slúžilo na neskoršie jednoduchšie predspracovanie popísané v časti 5.1 ale aj aby sa zamedzilo klasifikátorom možnému priradeniu rozličnej dôležitosti vytvoreným prediktorom. Iný typ škálovania indexov pre porovnanie nebol vykonaný kvôli časovej náročnosti tvorby datasetu.

Výrez jedného polygónu reprezentujúceho nejaký objekt alebo povrch prebieha postupne pre každý zo 4 časových záznamov a pre každé spektrálne pásmo a je implementovaný pomocou už spomínaného modulu rasterio. Každý vzniknutý obrázok má rozdielne rozmery a je reprezentovaný ako 4-rozmerné numpy pole tvaru $(T \times B \times W \times H)$, pričom T je počet časových záznamov ($T=4$), B je počet spektrálnych pásiem spolu so spektrálnymi indexami ($B=14$), W a H je šírka a výška daného obrázka.

Po vyrezaní obrázka nasledovalo maskovanie všetkých pixelov obsahujúcich oblačnosť, defektné pixely alebo sneh v danom obrázku. Tieto pixely boli lokalizované pomocou klasifikácie scény obsiahnutej v produkte Level-2A každej dlaždice.

Výsledný dataset obsahuje 103 737 obrázkov serializovaných vo formáte *.pickle*. Spolu s každým obrázkom reprezentujúcim jednu triedu povrchu sú uložené vo formáte *.pickle* aj maska oblačnosti a dodatočné údaje o *ground truth* obrázka, body tvoriace polygón obrázka, príslušný ohraničujúci obdĺžnik ale aj informácie o počte vylúčených pixelov na danom obrázku.

Vytvorený dataset vykazuje istú mieru nevyváženosti čo sa týka počtu reprezentujúcich obrázkov na triedu. Táto nevyváženosť po vylúčení chybných obrázkov je reprezentovaná na grafe 4.1.



Obr. 4.1: Nevyváženosť datasetu pre klasifikáciu po vylúčení chybných obrázkov. Kvôli nízkemu zasúpeniu tried *rock* a *square* boli tieto nakoniec vylúčené z neskoršej klasifikácie.

Analýzou rozmerov obrázkov v datasete sme dospeli k záveru, že priemerná šírka obrázka je 14 pixelov (280 m) so štandardnou odchýlkou 32 pixelov (640 m) a priemerná výška obrázka je 15 pixelov (300 m) so štandardnou odchýlkou 32 pixelov (640 m). Najväčšia dosiahnutá šírka je 2063 pixelov (41 km) a výška 2142 pixelov (42 km) prislúchajúca obrázku reprezentujúcemu triedu *water*. Treba podotknúť, že samotný rozmer obrázka nič nehovorí o tvare polygónu, ktorý reprezentuje konkrétny objekt pretože napríklad trieda *water* obsahuje množstvo vzoriek obrázkov na ktorých je vyobrazené len úzke riečne koryto. Práve triedy *agri*, *trees* a *grass*, ktoré majú v datasete najväčšie zastúpenie v porovnaní s ostatnými triedami vykazujú aj najväčšie rozptyly rozmerov. Časť obrázkov má dokonca rozmery rovné jednému pixelu. Takto malé obrázky boli v ďalšej príprave na učenie vylúčené. Pre predstavu rozličnosti v rozmeroch obrázkov v datasete pred vylúčením chybných obrázkov slúži histogram 4.2 s veľkosťou binu reprezentujúcim 3 pixely (60 m). Z tohto grafu môžeme usúdiť, že nadpolovičná väčšina obrázkov v datasete má rozmer do 6 pixelov (120 m) resp. štvrtina len rozmer do 3 pixelov (60 m). Ide vskutku o veľmi malé obrázky, často ľudským okom nerozpoznateľné, o aký objekt alebo povrch ide. Príklad troch zástupcov tried povrchov resp. objektov v datasete je na obrázku 4.5.

4.1.1 Použité triedy povrchov a objektov

Ako už bolo spomenuté na tvorbu datasetu pre klasifikačnú úlohu bolo vybraných 9 tried a povrchov s rešpektom k limitácii 20 m rozlíšenia na základe menšieho prieskumu používaných OSM značiek.

Tieto triedy povrchov spolu s ich významom sú uvedené v tabuľke 4.1.

Presný popis jednotlivých použitých značiek reprezentujúcich 9 tried povrchov/objektov nájde čitateľ v prílohe 5.2.2.3.

Vizuálnou analýzou obrázkov v datasete sa zistilo, že niektoré triedy ako trieda *agri* a *grass* môžu charakterizovať rovnaký povrch a ich rozdiel spočíva iba v značke priradenej používateľom OSM. Nie je teda jasné o akú triedu by malo ísť.

Predovšetkým kvôli problému s nedostatkom reprezentatívnych vzoriek obrázkov sa výsledné analýzy a aplikácie strojového učenia vykonávali len na 7 triedach po vylúčení tried *rock* a *square*.

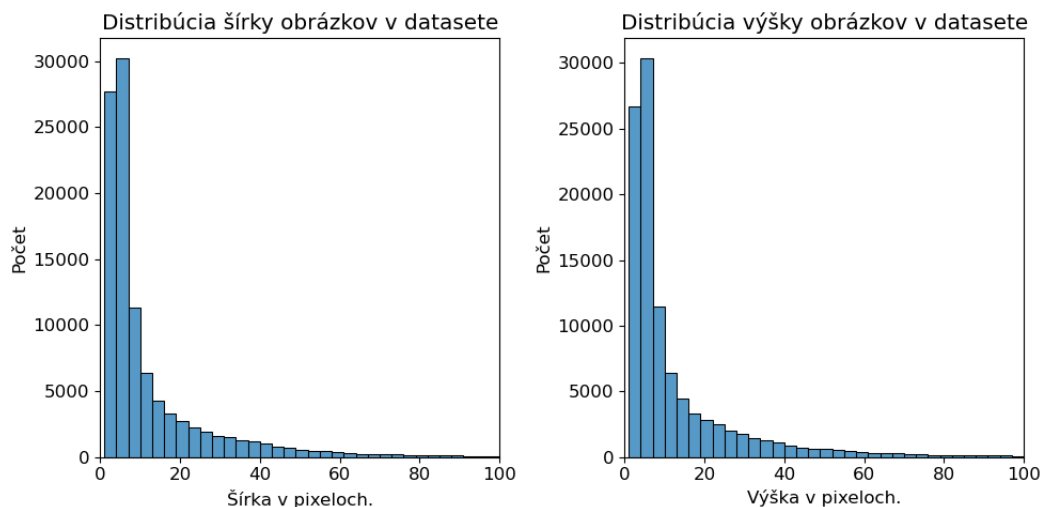
Označenie	Význam
trees	Povrch obsahujúci stromy rôzneho druhu
grass	Povrch reprezentujúci trávnatu oblasť .Môže ísť o pasienky, lúku ale aj napr. časť trávinatej plochy v parku
building	Trieda slúžiaca na popis jednotlivých budov ako rodinný dom, univerzita, apartmány, supermarket a iné
asphalt	Povrch obsahujúci asphalt ako parkovacie miesto, časť cesty a iné
cemetery	Pohrebisko/Cintorín
water	Vodné plochy ako jazero, časť rieky, nádrž
agri	Pole slúžiace na poľnohospodárske účely
rock	Skala
square	Námestie

Tabuľka 4.1: Charakteristika tried povrchov pre klasifikačnú úlohu

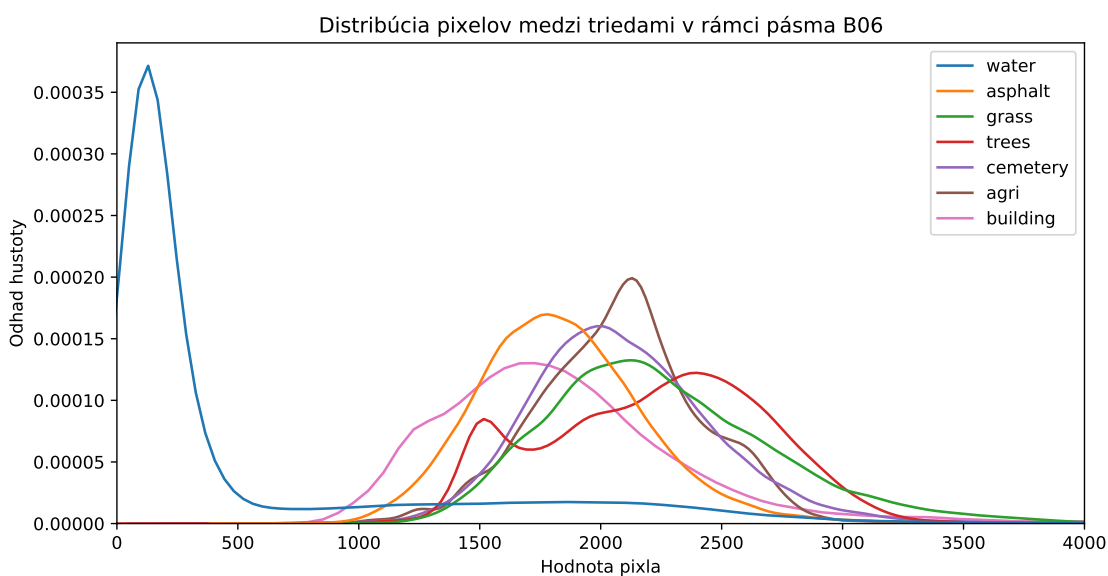
Mieru podobnosti resp. rozdielnosti medzi triedami môžeme vypočítavať porovnaním rozloženia hodnôt pixelov reprezentujúce odrazivosť v danom spektrálnom pásme. Porovnanie rozložení pixelov medzi triedami vzhľadom k pásmu B06 sú zobrazené na obrázku 4.3. Na tomto odhade hustoty pravdepodobnosti hodnôt pixelov možno vidieť, že trieda *water* má veľmi špecifický tvar a viditeľne sa líši od ostatných tried. Takýto záver vyplýva aj z vizuálnej analýzy distribúcií vzhľadom k iným spektrálnym pásmam ako aj indexom. Takto špecifické rozloženie hodnôt prevažne s nízkou odrazivosťou v pásmach naznačuje, že trieda *water* bude najľahšie odlišiteľná od ostatných tried. V pásmach B02, B03 a B04 vykazuje trieda *trees* podobné rozloženie ako má trieda *water* ale v ostatných pásmach predovšetkým v B06, B07 (červený okraj) ako aj v indexe NDVI sa rozloženia opäť vizuálne významne líšia.

Veľmi podobné rozloženia vykazujú aj trieda *building* a *asphalt* vo všetkých spektrálnych pásmach ako aj indexoch čo prisudzujeme možnosti, že strecha budovy môže byť tvorená asfaltom alebo povrchom s veľmi podobnou spektrálnou charakteristikou. Opäť tento výsledok naznačuje, že tieto dve triedy budú pre klasifikátor ťažké na rozlíšenie. Podobne predpokladáme, že problém v klasifikácii a častú zámennosť bude spôsobovať aj trojica tried *agri*, *grass* a *cemetery* i keď asi nie v takej miere ako trieda *asphalt* a *building*.

Podobne môžeme odhadnúť významnosť konkrétnych spektrálnych pásiem použitých na neskoršiu klasifikáciu konkrétnej triedy povrchu. Príklad takéhoto rozloženia hodnôt pixelov v rámci triedy *asphalt* porovnávané medzi jednotlivými spektrálnymi pásmami bez indexov sú zobrazené na obrázku 4.4. Na danom obrázku môžeme vidieť, že pre triedu *asphalt* sú rozloženia pásiem B07 a B11 veľmi podobné čo poukazuje na možnú korelovanosť daných dvoch pásiem. Taktiež je z grafu zjavná podobnosť rozptylov distribúcií jednotlivých pásiem.

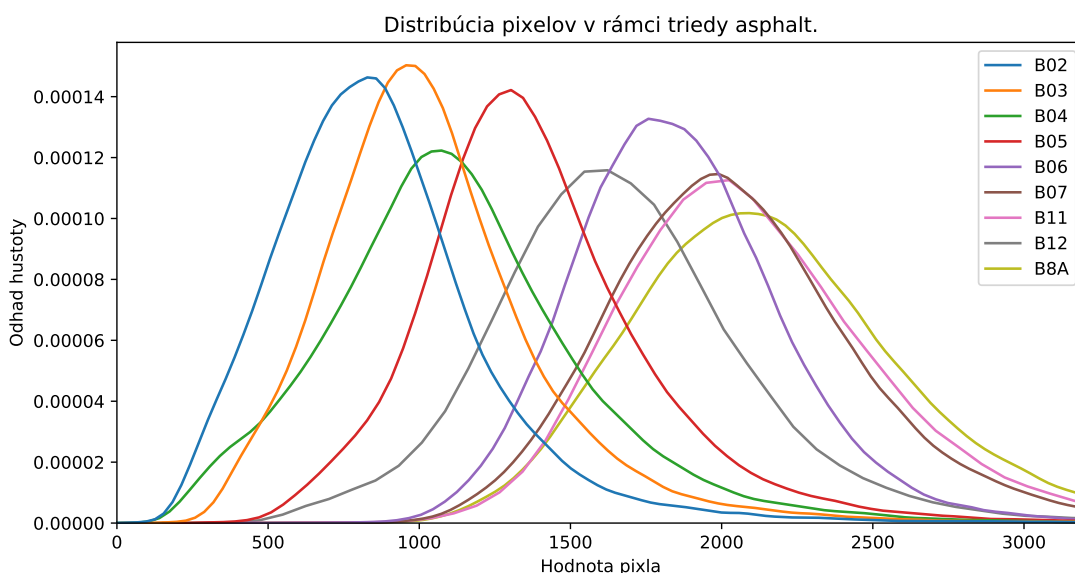


Obr. 4.2: Distribúcia rozmerov obrázkov v datasete pre klasifikáciu.

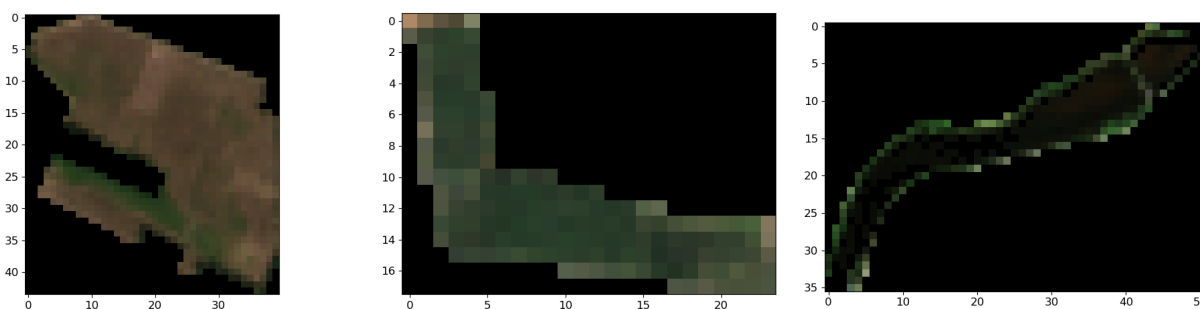
Obr. 4.3: Porovnanie odhadu rozložení hodnôt odrazivosti medzi triedami vzhľadom k spektrálnemu pásmu *B06*. Zjavná odlišnosť v rozložení triedy *water* napovedá, že by táto trieda mohla byť najľahšie odlišiteľná od ostatných skúmaných tried povrchov/objektov.

4.2 Dataset pre sémantickú segmentáciu

Obrázky určené pre klasifikáciu na úrovni pixelov resp. sémantickú segmentáciu nevyžadujú aby sa na jednom obrázku nachádzal práve jeden typ povrchu a preto nebolo nutné vyrezávať objekty na základe ohraničujúcich obdĺžnikov ako bolo učené pri tvorbe datasetu pre klasifikačnú úlohu. Nutnosťou v tomto prípade bolo vytvoriť *ground truth* segmentačnú masku všet-



Obr. 4.4: Porovnanie odhadu rozložení hodnôt pixelov v rámci triedy *asphalt* medzi jednotlivými spektrálnymi pásmami (nezahŕňajúc spektrálne indexy). Podobnosť v distribúciach niektorých pásiem naznačuje možnú koreláciu medzi danými pásmami.



Obr. 4.5: Príklad zástupcov tried v prirodzenej farbe (zľava) *grass*, *trees*, *water*. Čierne pixely reprezentujú vylúčené okolie. Nízke hodnoty odrazivosti vody vo väčšine spektrálnych pásiem spôsobujú tmavý nádych pixelov takmer splývajúcich s vylúčenými pixelmi na pravom obrázku reprezentujúcim triedu *water*, pričom zelené okraje patria vegetácii a stromom na brehu.

kých snímok. Nakoľko sme mali informáciu o *ground truth* jednotlivých objektov z OSM uloženú lokálne v shapefile tak bola *ground truth* segmentačná maska tvorená postupne. Pomocou modulu *rasterio* sme vždy vytvorili akúsi podmasku dlaždice naraz vo všetkých štyroch časových okamihoch a všetkých spektrálnych pásmach zahŕňajúc spektrálne indexy popísané v časti 2.2.1.2. Jednou z podmasiek ako bude uvedené ďalej je aj samotná oblačnosť, ktorú sa pokúsime taktiež klasifikovať pričom bola získaná z klasifikácie scény nachádzajúcej sa v produkte Level-2A. Podmasky boli nakoniec vykreslené iteratívne do jedného rastra reprezentujúceho výslednú *ground truth* masku obsahujúcu kódy jednotlivých povrchov.

Pretože ale mala každá dlaždica ako aj vzniknutá segmentačná maska v 20 m rozlíšení rozmer 5490 x 5490 pixelov tak sme sa ich rozhodli narezat' na 2025 menších neprekrývajúcich sa obrázkov rozmerov 122 x 122 pixelov. Okrem toho bol každý obrázok už od začiatku rozdelený na 4 samostatné obrázky rovnakého územia v 4 rôznych časových okamihoch. Obrázky boli nakoniec ešte prefiltrované a vylúčili sme tie, ktoré obsahovali zhruba viac ako 15 % pixelov triedy označenej ako *Nodata* vznikajúcich z neuplného pokrytia niektorých dlaždíc obrazovou informáciou.

Výsledkom týchto úprav je množina 145 703 obrázkov uložených vo formáte *.pickle* s pracovným názvom *OSMLandCoverDataset*. Serializované vo formáte (*.pickle*) sú multispektrálny obrázok daného územia reprezentovaný numpy poľom rozmerov (B x W x H) kde (B=4) je počet príslušných spektrálnych pásiem s indexami a W=122 a H=122 je šírka a výška obrázka v pixeloch. Okrem toho sa tam nachádza príslušná segmentačná maska tiež reprezentovaná numpy poľom rozmerov (B=1 x W=122 x H=122), dodatočná informácia o čase kedy bola snímka zaznamenaná a informácia o použitých spektrálnych pásmach spolu s indexami.

Vzniknutý *OSMLandCoverDataset* vykazuje výraznejšiu mieru nevyváženosti čo sa týka počtu pixelov na triedu povrchu oproti datasetu pre klasifikačnú úlohu z predchádzajúcej sekcie. Táto nevyváženosť po vylúčení obrázkov s viac ako 50% zastúpením pixelov triedy *Unknown* je zobrazená na obrázku 4.6.

4.2.1 Vytvorené sémantické triedy povrchov

Pri voľbe OSM značiek definujúcich triedy povrchov pre sémantickú segmentáciu sme používali OSM značky určené na popis krajinného pokryvu a využitia vytvorené v práci [74]. Niektoré triedy povrchov sme rozšírili o OSM značky použité už pri tvorbe datasetu pre klasifikáciu. Okrem tried povrchov boli na tvorbu *ground truth* masiek zahrnuté aj dodatočné tri triedy reprezentujúce postupne, oblasti bez informácie o *ground truth* (*Unknown*), oblasti bez obrazovej informácie (*Nodata*), a triedu pre označenie oblačnosti (*Clouds*). Označenie, použitý kód a význam jednotlivých tried povrchov sú zhrnuté v tabuľke 4.2.

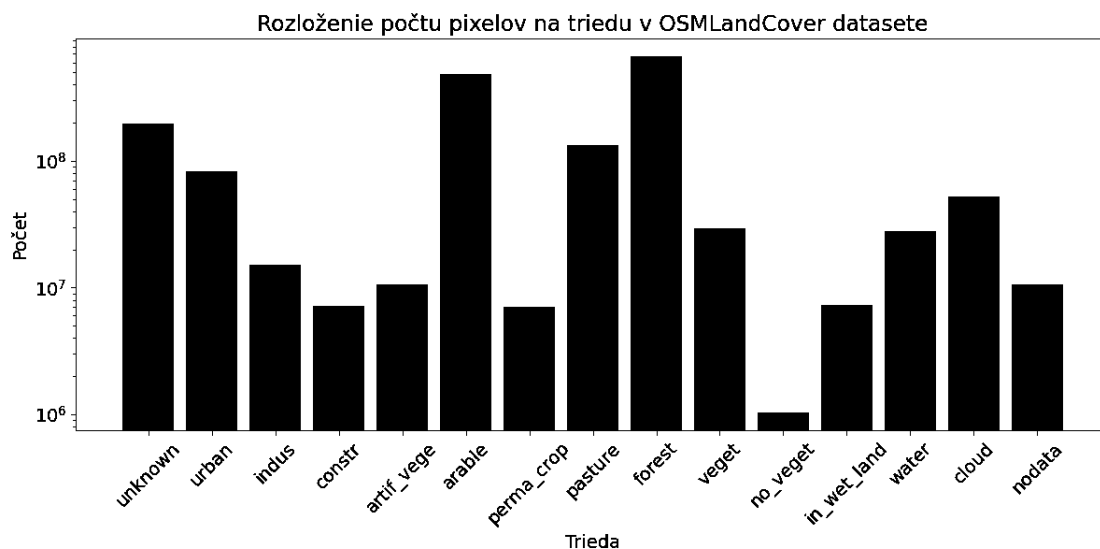
Opäť platí, že presný popis jednotlivých OSM značiek použitých na tvorbu 12 z 15 tried povrchov nájde čitateľ v prílohe 5.2.2.3.

4.2.2 Spektrálne charakteristiky snímok

Po vytvorení *OSMLandCover* datasetu nasledovala predovšetkým vizuálna inšpekcia obrázkov ako aj analýza spektrálnych profilov jednotlivých tried. Ukázalo sa, že použitie triedy *Unknown* prinesie v segmentačnej úlohe zásadný problém nakoľko je prevažne tvorená typom povrchov, ktoré už sú zahrnuté v iných skúmaných triedach. Z OSM sme ale nezískali informáciu o type daného povrchu označeného ako trieda *Unknown* pretože sa ešte nenašiel nikto kto by túto informáciu do OSM zahrnul. Bližšie informácie o prístupe k spracovaniu triedy *Unknown* budú uvedené v sekcii 5.2. Ďalším významným problémom je podobnosť spektrálnych signatúr niektorých tried vyplývajúcim z ich zadefinovania ale aj nepresnosti v značkovaní OSM objektov. Na

Kód	Označenie	Význam
0	Unknown	Reprezentuje pixely bez informácie o <i>ground truth</i> . Najčastejšie dané územie neobsahovalo žiaden popis v OSM alebo daný povrch nie je zahrnutý v analýze.
1	Urban	Obytná oblasť. Môže zahŕňať aj vegetáciu v okolí domov, záhrady a pod.
2	Industrial & commercial	Zahŕňa priemyselné oblasti so súvisiacou infraštruktúrou (železnica, prístav), budovy slúžiace na komerčné účely (sklady) ale aj povrchy obsahujúce asfalt, betón a iné.
3	Mine, dump & construction	Označuje staveniská, skládky alebo doly.
4	Artificial non-agricultural vegetated areas	Zahŕňa všetky umelo vytvorené územia s trávnatým povrchom ako park, štadión, cintorín, záhrada, golfové ihrisko a iné.
5	Arable land	Územie slúžiace na poľnohospodárske účely predovšetkým na pestovanie plodín.
6	Permanent crops	Špecificky označuje oblasť používanú na pestovanie viniča alebo ovocné sady.
7	Pastures	Územie označujúce lúku alebo pastviny.
8	Forest	Lesný porast.
9	Vegetation	Slúži na popis trávnatých oblastí alebo oblastí s nízkou vegetáciou ako sú kry.
10	Open spaces with little or no vegetation	Reprezentuje kamenisté, štrkové a pieskové povrchy s málo alebo žiadnou vegetáciou.
11	Inland wet lands	Mokrade a močiare.
12	Water bodies	Vodné plochy ako jazerá, rieky, nádrže.
13	Clouds	Dodatočná trieda pre pixely obsahujúce oblačnosť.
14	Nodata	Reprezentuje pixely bez informácie o hodnotách spektrálnych pásiem.

Tabuľka 4.2: Charakteristika tried povrchov pre segmentačnú úlohu určujúca označenie povrchu jeho význam ako aj kód použitý v segmentačnej maske.

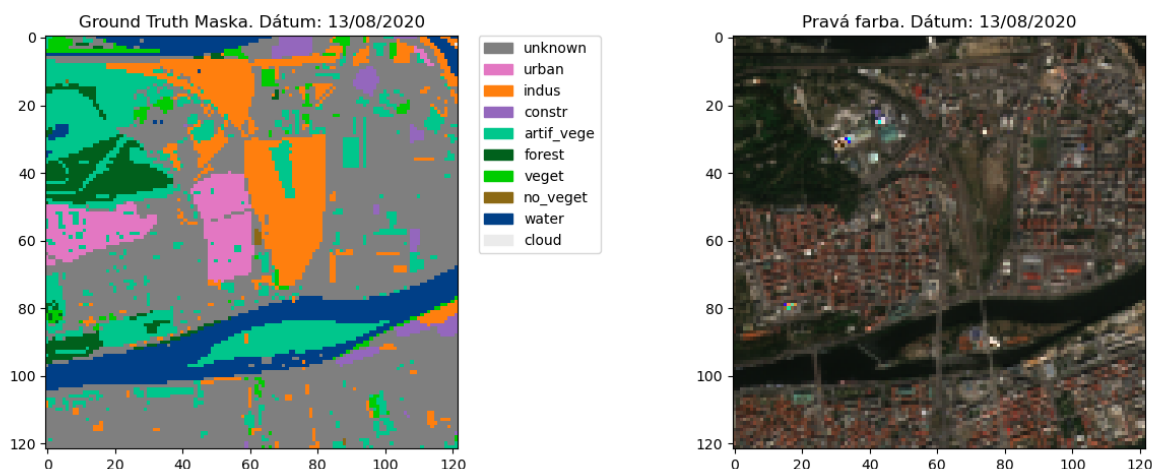


Obr. 4.6: Nevyváženosť *OSMLandCover* datasetu vzhľadom k počtu pixelov na triedu. Názvy tried sú skrátené. Ako vidíme tak najpočetnejšia je trieda *Forest* (6.66×10^8) nasledovaná triedou *Arable* (4.88×10^8) čo je v konečnom dôsledku prirodzené nakoľko väčšinu vnútrozemského povrchu tvoria poľnohospodárske územia a lesy. Vysoké zastúpenie počtu pixelov majú aj triedy *Unknown* (1.98×10^8) a *Nodata* (1.05×10^7), ktoré budú vyžadovať odlišné spracovanie predovšetkým pri odhade klasifikačných výkonov klasifikátorov.

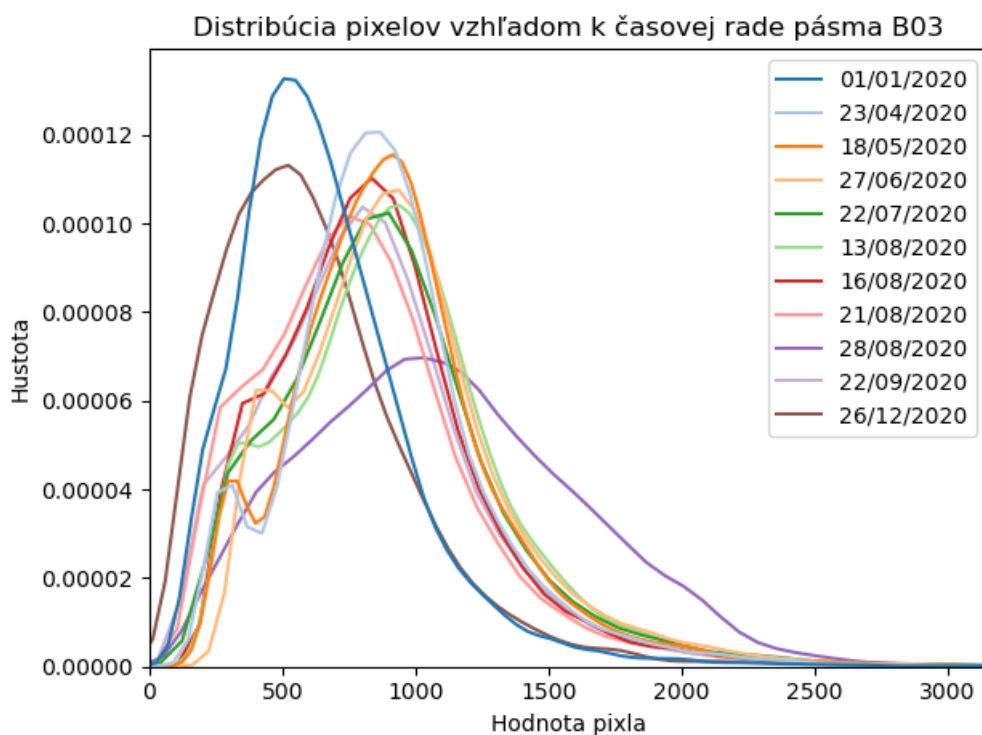
grafoch 4.9 môžeme vidieť podobnosť v spektrálnych profiloch tried *Artificial non-agricultural vegetated areas* a *Forest*. Náhlý vzrast v pásmach B05, B06 a B07 je známy ako vegetačný červený okraj. V prípade triedy *Industrial & commercial* vidíme zmes pixelov zjavne z dvoch rôznych povrchov pričom časť z nich je podobná spektrálnej signatúre tried *Artificial non-agricultural vegetated areas* a *Forest*. Takáto podobnosť či už spektrálnych profilov tried alebo povrchov na samotných snímkach v pravých farbách bola pozorovaná aj v prípade iných použitých obrázkov. Opäť to naznačuje možnú zníženú separabilitu takýchto tried s podobnou signatúrou. V prípade sémantickej segmentácie bude ale použitý aj model neurónovej siete využívajúci priestorovú informáciu, ktorá môže pomôcť pri separabilite takýchto spektrálne podobných sémantických tried povrchov.

Príklad segmentačnej masky spolu so snímkou v pravých farbách nad územím zahŕňajúcim Prahu-Holešovice je na obrázku 4.7. Táto snímka bola použitá aj na vygenerovanie prezentovaných spektrálnych profilov.

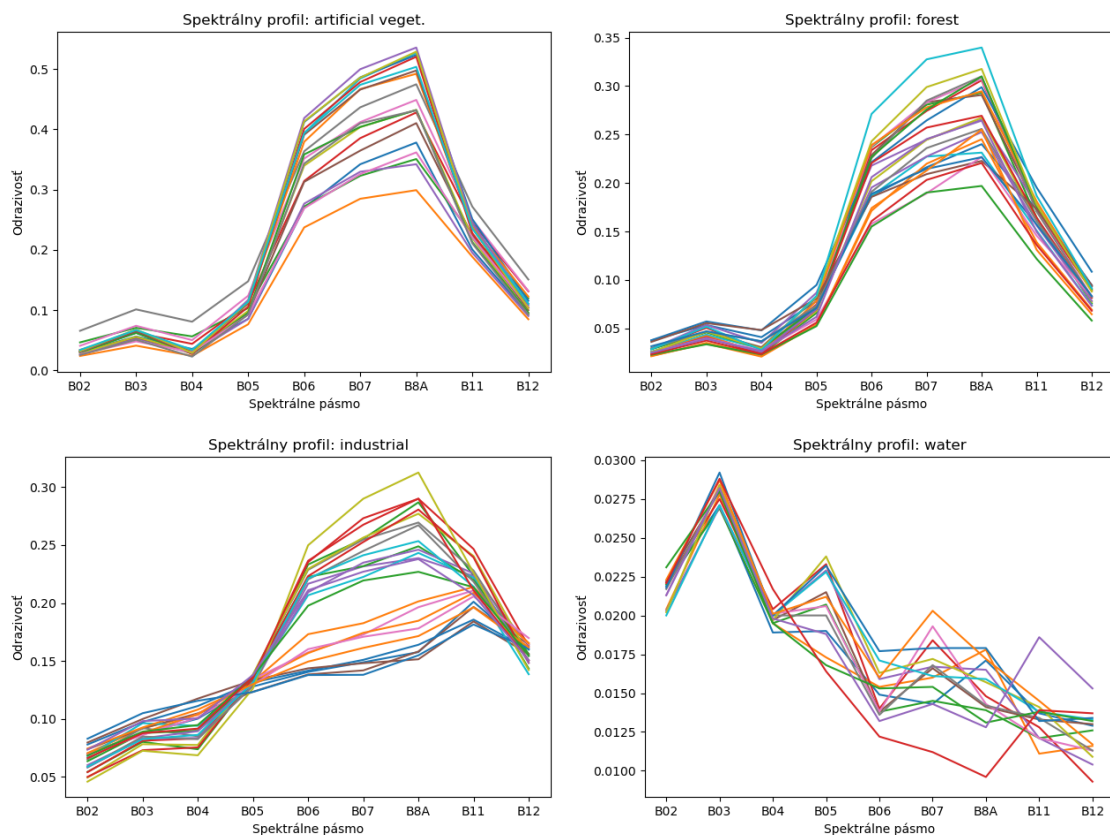
Okrem spektrálnych signatúr resp. profilov sme analyzovali obrázky aj vzhľadom k zmene jasú pomocou histogramov resp. jadrových odhadov hustôt rozložení pixelov obrázka v RGB pásmach (B04, B03, B02). Touto analýzou zmeny jasú vzhľadom k času sme chceli overiť vplyv zmeny ročného obdobia na jas a v konečnom dôsledku potrebu mať rozsiahlejšiu časovú radu snímok kvôli zabezpečeniu robustnosti modelu voči týmto zmenám v jase resp. potrebu štandardizácie obrázkov aby bol minimalizovaný vplyv rozdielnosti v jase.



Obr. 4.7: Segmentačná maska (naľavo) a snímka v pravých farbách (napravo) územia nad Prahou (Holešovice) zachyteného 13.8.2020. Rozmery obrázka sú 122x122 pixelov (2440x2440 m). Takmer polovica tohto obrázku je označená ako *Unknown* aj keď zjavne by mala patriť triede *Urban* alebo *Industrial & Commercial*. Toto bolo spôsobené kvôli použitiu značiek z OSM zachytávajúcich len súvislé obytné oblasti (*landuse=residential*) a nie jednotlivé budovy.



Obr. 4.8: Odhad hustôt rozložení pixelov pásma B03 na snímke územia nad Holešovicami. Výrazný posun smerom k nižšej odrazivosti charakterizujúci zníženie jasu snímky boli zaznamenané v prípade zimných mesiacov. Rozdiel v rozložení zo snímky z 28.8.2020 bol spôsobený prítomnosťou oblačnosti.



Obr. 4.9: Spektrálne signatúry resp. profily tried *Artificial non-agricultural vegetated areas*, *Forest*, *Industrial & commercial* a *Water* odhadnuté s použitím časti snímky nad Holešovcami. Na grafoch vidíme podobnosť v spektrálnej signatúre tried *Artificial non-agricultural vegetated areas* a *Forest*. Pixely triedy *Industrial & commercial* naznačujú prítomnosť dvoch rôznych povrchov pričom jeden z nich je podobný signatúre tried *Artificial non-agricultural vegetated areas* a *Forest*. Pixely triedy *Water* vykazujú nízku odrazivosť vo všetkých spektrálnych pásmach.

Na analýzu tohto sezónneho vplyvu sme vytvorili nový dataset vytvorený len z jednej dlaždice T33UVR a obsahujúci okrem už použitých štyroch snímok z obdobia 1.6.2020-31.8.2020 aj dodatočných 7 snímok zhruba pokrývajúcich celý rok 2020. Mesiace, ktoré nie sú zahrnuté boli február, marec, október a november a to kvôli príliš veľkému množstvu oblačnosti v snímkach. Ostatné mesiace majú aspoň jedného zástupcu medzi snímkami. Na kvantifikovaný odhad jasu sme použili výberový priemer intenzít pixelov reprezentujúcich odrazivosť (*reflectance*) v RGB pásmach a pozorovali sme rozdiel resp. posun v týchto rozložení pixelov v histogramoch. Graf na obrázku 4.8 reprezentuje odhad hustôt rozdelení pre pásmo B03 pre prípad snímky nad Holešovcami. Ako môžeme vidieť výraznejší rozdiel resp. posunutie vykazujú dva snímky z 1.1.2020 a 26.12.2020 oba pochádzajúce zo zimného obdobia. Toto posunutie reprezentuje miernu zmenu v jase na týchto snímkach čo bolo zistené aj vizuálnou inšpekciou daných dvoch snímok. Ďalší významný rozdiel vidno aj na hustote rozloženia snímky z 28.8.2020 čo bolo ale spôsobené viditeľnou prítomnosťou oblačnosti na danej snímke 28.8.2020. Analýzou aj iných snímok a to aj v pásmach B02 a B04 sa ukázalo, že výraznejší pokles v jase dochádza len v zimných mesiacoch. Snímky z týchto mesiacov ale často vykazujú príliš veľké zastúpenie oblačnosti alebo snehu a preto neboli použité ani v našej práci. Taktiež sa ukázala zmena v rozloženiach pixelov

aj v prípade ak snímka pokrývala územia s poľnohospodárskymi poľami resp. územia ktoré sa menia vplyvom zmeny ročného obdobia. Na väčšiu robustnosť klasifikátora na zmenu v spektrálnej charakteristike daných území by bolo vhodné využiť väčšiu časovú radu snímok pre dané územie. Presnejší resp. kvantifikovaný odhad významnosti zníženia jasů by vyžadoval štatistický testovanie čo ale nebolo našim zámerom.

Kapitola 5

Klasifikácia a sémantická segmentácia povrchov

V prvej časti tejto kapitoly priblížime samotnú metodiku použitú na riešenie úlohy, implementáciu jednotlivých použitých algoritmov na klasifikáciu povrchov a objektov na obrázkoch vyrezaných zo satelitných snímok a následne zhodnotíme a porovnáme výsledky jednotlivých algoritmov.

V druhej časti sa pozrieme na úlohu klasifikácie typu povrchov na úrovni pixelov teda sémantickú segmentáciu v satelitných snímkach. Opäť popíšeme prístup k riešeniu úlohy, implementáciu použitých algoritmov a zhrnieme a porovnáme ich výsledky.

5.1 Klasifikácia

Klasifikácia povrchov v satelitných snímkach typicky znamená priradenie triedy povrchu každému pixelu v snímke. V našej práci ale striktno rozlišujeme klasifikáciu snímok od klasifikácie jednotlivých pixelov teda sémantickej segmentácie popísanej v ďalšej časti. V tejto časti teda riešime úlohu kde predpokladáme, že získame snímku na ktorej bude práve jedna trieda povrchu o ktorú sa zaujíname. Okrem toho predpokladáme, že máme informáciu o hraniciach daného objektu, ktorá umožní výpočet spektrálnych prediktorov bez vplyvu okolia. V praxi takto priaznivé podmienky a zdroj informácií nemáme a preto je táto úloha zamýšľaná hlavne na preskúmanie možnosti aplikácie algoritmov strojového učenia na dáta reprezentujúce čisto spektrálnu charakteristiku daných objektov a povrchov v satelitných snímkach.

5.1.1 Metodika

Pre túto úlohu bol vytvorený samostatný dataset popísaný v predchádzajúcej sekcii 4.1. Samotný dataset obsahuje zhruba 103 000 obrázkov rôznych rozmerov. Nakoľko sme sa v tejto úlohe zamerali na použitie algoritmov nezhrňajúcich neurónové siete tak bolo treba vytvorený dataset pretransformovať do reči prediktorov (*features*) charakterizujúcich vlastnosti danej triedy odvodené z obrázka.

Túto transformáciu predchádzalo predspracovanie datasetu čo slúžilo na vylúčenie chybných obrázkov s rozmermi jeden pixel. Taktiež boli vylúčené obrázky s menej ako 20% zastúpením pixelov nesúcich relevantnú informáciu teda v podstate nezahŕňajúc oblačné, nodata a pixely patriace pozadiu. Okrem vylúčenia chybných obrázkov zahŕňalo predspracovanie aj voliteľnú normalizáciu obrázkov a to konkrétne *min-max* škálovanie a normovanie podľa L1 a L2 normy pričom

sa normalizovalo vzhľadom k obrázku ako celku zahŕňajúc tak všetky spektrálne pásma naraz aby nedošlo k strate relevantnej informácie obsiahnutej v prirodzenom rozložení hodnôt jednotlivých spektrálnych pásiem. Tento prístup ale nepriniesol žiadné významné zlepšenie výkonov klasifikátorov, či dokonca niektoré testy vykazovali zhoršenie kvality predikcie preto sme takúto normalizáciu vypustili ako nepotrebnú.

Prirodzene sme ako prediktory (*features*) vybrali výberový priemer a štandardnú odchýlku vypočítanú pre každé spektrálne pásmo, agregujúce spektrálnu informáciu obsiahnutú v spektrálnych pásmach obrázka. Spolu bolo teda vytvorených 28 prediktorov charakterizujúcich každý objekt resp. obrázok. Prediktory založené napr. na tvare objektu neboli použité kvôli zjednodušeniu ale aj kvôli značnej variácii vo veľkostiach obrázkov a možným zkriveniam tvaru vzniknutým pri vylúčení chybných (oblačnosť, nodata) pixelov.

Výsledkom všetkých popísaných predspracovaní vznikol nový dataset obsahujúci 72 712 vzoriek pričom každá z nich je v 4 časových okamihoch umožňujúci tak štvornásobné rozšírenie datasetu na 290 848 vzoriek.

Použité algoritmy pozostávali z klasifikátorov založených na skupinovom učení (*ensemble learning*) a to konkrétne *random forest* a *gradient boosting* obe poskytujúce špičkové výsledky v oblasti strojového učenia s učiteľom na tabulovaných dátach. Okrem týchto dvoch algoritmov sme na učenie využívali aj algoritmus *k-nearest neighbours* vzhľadom k jeho jednoduchšej implementácii a ladeniu. Bližší popis uvedených troch algoritmov bol uvedený v kapitole 3 o strojovom učení.

Rozdelenie datasetu bolo uskutočnené na dve časti a to na trénovací a testovací dataset. V trénovacom datasete bolo 80% a v testovacom 20% vzoriek z celkového množstva v datasete. Rozdelenie do trénovacieho a testovacieho datasetu prebehlo náhodne pričom sme využili stratifikáciu zabezpečujúcu rovnaké percentuálne zastúpenie jednotlivých tried v trénovacom aj testovacom datasete. Na zníženie vplyvu nevyváženosti trénovacieho datasetu sme v niektorých testoch skúšali použiť podvzorkovanie na 5200 vzoriek na triedu. Samotný testovací dataset nebol nijakým spôsobom vyvažovaný kvôli zachovaniu prirodzenej distribúcie počtu vzoriek na triedu a obsahoval vždy 14 379 vzoriek v jednom časovom okamihu.

Hodnotenie klasifikačných výsledkov a generalizačných vlastností klasifikátorov ako aj porovnanie použitých prístupov prebiehalo na základe metrík *accuracy*, *recall*, *precision* a *F1 score* popísané v sekcii 3.1.1.4. Použitie metriky *F1 score* vyplýva z faktu, že lepšie popisuje kvalitu klasifikátora vzhľadom k použitiu nevyváženého datasetu. Ako sa však ukáže, nevyváženosť datasetu nebola výrazná a preto postačovalo používať na celkové hodnotenie výsledku metriku *accuracy*.

Implementácia algoritmov bola zabezpečená pomocou pythonovskej knižnice scikit-learn [66] slúžiacej na jednoduchú aplikáciu strojového učenia a knižnice XGBoost [25] vytvorenú na efektívne, rýchlu a škálovateľnú implementáciu *gradient boosting*-ových modelov. Na zjednotenie implementácie medzi XGBoost a scikit-learn sme využívali API poskytované XGBoost-om napodobňujúce scikit-learn rozhranie.

Na zrýchlenie učenia sme používali paralelnú implementáciu s využitím 48-jadrového procesora Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz a v prípade XGBoost-u sme akcelerovali učenie využitím grafickej karty NVIDIA Tesla V100. Výpočetné kapacity boli poskytnuté Ústavom teórie informácie a automatizácie AV ČR.

5.1.2 Experimenty a porovnanie

Testy a porovnanie klasifikátorov zahŕňali niekoľko nastavení vzhľadom k počtu a typu použitých prediktorov, k využitiu časovej rady každej vzorky v datasete ale aj vzhľadom k vyváženosti resp. nevyváženosti datasetu. Vo všetkých testovaných nastaveniach popísaných nižšie sa ukázal algoritmus XGBoost ako najvhodnejší a dosahujúci najvyššie hodnoty celkových sledovaných metrick *accuracy* a *F1 score* na testovacom datasete.

5.1.2.1 Hľadanie optimálnych hyperparametrov

Prvé experimentovanie zahŕňalo hľadanie optimálnych hyperparametrov. Na ich získanie sme využili framework *tune* [56] knižnice Ray pre tzv. ladenie hyperparametrov na akejkol'vek škále a konkrétne sme používali randomizovanú verziu hľadania. Cross-validácia pri ich hľadaní nebola vykonaná predovšetkým kvôli časovej náročnosti daného úkonu. Optimálna hodnota parametra bola teda určená po natrénovaní klasifikátora na náhodne vybratej podmnožine parametrov a vyhodnotení jeho výsledku na testovacom datasete na základe metriky *accuracy*. Takto vybratých podmnožín parametrov bolo niekoľko stoviek pre každý algoritmus. Presný počet skúmaných podmnožín parametrov záležal na časovej náročnosti učenia daného algoritmu. Hľadanie hyperparametrov prebiehalo na podvzorkovanom tréningovom datasete s 5200 vzorkami na triedu a bolo použitých všetkých 28 prediktorov. Z každej vzorky bol použitý len jeden časový okamih resp. člen časovej rady.

Popis jednotlivých uvažovaných hyperparametrov spolu s ich charakteristikou a optimálnou hodnotou pre jednotlivé algoritmy nájde čitateľ v prílohe 5.2.2.3. Všetky získané hyperparametre boli použité aj v ďalších experimentoch a už neboli ďalej optimalizované.

- **K-Nearest neighbours**

Pri experimentoch s hyperparametrami algoritmu *k-nearest neighbours* sa ukázalo ako najzásadnejšie nastaviť parametre *weight* a *n_neighbours*. Použitím *weight='uniform'* bolo dosiahnutá na testovacom datasete *accuracy* 67 % pričom *accuracy* na tréningovom datasete bola o pár percent vyššia. Na druhú stranu pri použití *weight='distance'* sa ukázal náhly vzrast tréningovej *accuracy* na 100 % pričom táto metrika na testovacom datasete dosiahla 69 % pri optimálnej hodnote parametra *n_neighbours=17* čo poukazuje na pretrénovanie (*overfitting*) algoritmu, teda zhoršenie generalizačnej schopnosti klasifikátora na vopred nevidených dátach. Toto pretrénovanie sa ďalej nepodarilo zredukovať inými zmenami parametrov preto za najlepší výsledok získaný týmto algoritmom bola považovaná hodnota *accuracy* **69.22%** na testovacom datasete. Priemerné *F1 score* dosiahnuté na testovacom datasete bolo **69.14%**.

- **Random forest**

Ďalším v poradí bol algoritmus *random forest* (náhodný les). Najzásadnejšími hyperparametrami ovplyvňujúcimi jeho výkon sú *n_estimators* teda počet rozhodovacích stromov v lese, *max_depth* a *max_features* a ďalšie ovplyvňujúce pretrénovanie a komplexitu modelu. Po získaní optimálnych parametrov bolo dosiahnutá testovacia *accuracy* **72.29%** a priemerné *F1 score* **72.14%** pričom algoritmus bol zjavne pretrénovaný dosahujúci tréningovú *accuracy* 100 %. Všetky pokusy o zredukovanie pretrénovania pomocou nastavení hyperparametrov ako *max_depth* ovplyvňujúcu komplexitu samotného rozhodovacieho stromu alebo *min_samples_split* boli neúspešné a viedli k zníženiu celkovej testovacej *accuracy*.

- **Gradient boosting**

	<i>k</i> -NN	Random forest	XGBoost
Accuracy	69.21%	72.28%	74.43%
F1 score	69.14%	72.12%	74.35%

Tabuľka 5.1: Porovnanie výsledkov algoritmov po ladení hyperparametrov. Môžeme vidieť, že v oboch celkových pozorovaných metrikách dosahuje najlepšie výsledky klasifikátor založený na *gradient boosting*-u (XGBoost). Trénovací dataset obsahoval vždy 5200 vzoriek na triedu.

Posledným analyzovaným algoritmom bol *gradient boosting* v implementácii XGBoost. Načo ide o aditívny model, učiaci sa iteratívne tak bolo najzásadnejšie určiť počet iterácií zvaných *num_rounds*. Tento parameter sa v scikit-learn API nazýva *n_estimators* a bol nastavený na hodnotu 1000 pričom bolo využité skoré zastavenie (*early stopping*), ktoré nastane ak sa metrika medzi istým počtom iterácií nezmení prípadne zhorší. Tento model znovu vykazoval značné pretrénovanie a pokusy o jeho redukciu boli sprevádzané významným znížením metriky *accuracy* na testovacom datasete. V konečnom dôsledku sa ale podarilo dosiahnuť v danom nastavení najlepší klasifikačný výsledok spomedzi všetkých troch algoritmov čo sa týka metriky *accuracy* a to až **74.43%**. Priemerné *F1 score* dosahovalo hodnotu **74.35%**.

Dalšou analýzou klasifikačných výsledkov týchto troch algoritmov sme si všimli, že algoritmus *k-nearest neighbours* významne nadhodnotil triedu *cemetery* vo svojej predikcii v porovnaní so skutočným zastúpením tejto triedy v testovacom datasete. Vo výsledku to znamenalo relatívne vysokú hodnotu metriky *recall* (72.44%) ale na druhú stranu nižšiu hodnotu metriky *precision* (49.95%) čo sa dá interpretovať tak, že model dokáže detekovať túto triedu povrchu ale samotná trieda obsahuje veľké množstvo šumu v konečnom dôsledku vyplývajúceho zo vzhľadu pohrebísk a cintorínov (*cemetery*) kde sa nachádzajú často trávne plochy, stromy, náhrobky a ale aj menšie budovy. Tento výsledok bol ale očakávaný.

Algoritmus *k-nearest neighbours* ďalej vykazoval najväčší problém v detekcii triedy *grass* a *building* čo je podporené hodnotami metriky *recall* postupne 55.99% a 58.14%.

Algoritmus *random forest* nepriniesol zlepšenie detekcie triedy *cemetery* oproti *k-nearest neighbours* ale takmer o 10% zvýšil hodnotu metriky *precision* reprezentujúcu spoľahlivosť pri predikcii danej triedy.

Algoritmus XGBoost priniesol ešte ďalšie zlepšenie každej sledovanej metriky oproti *random forestu* o pár percent.

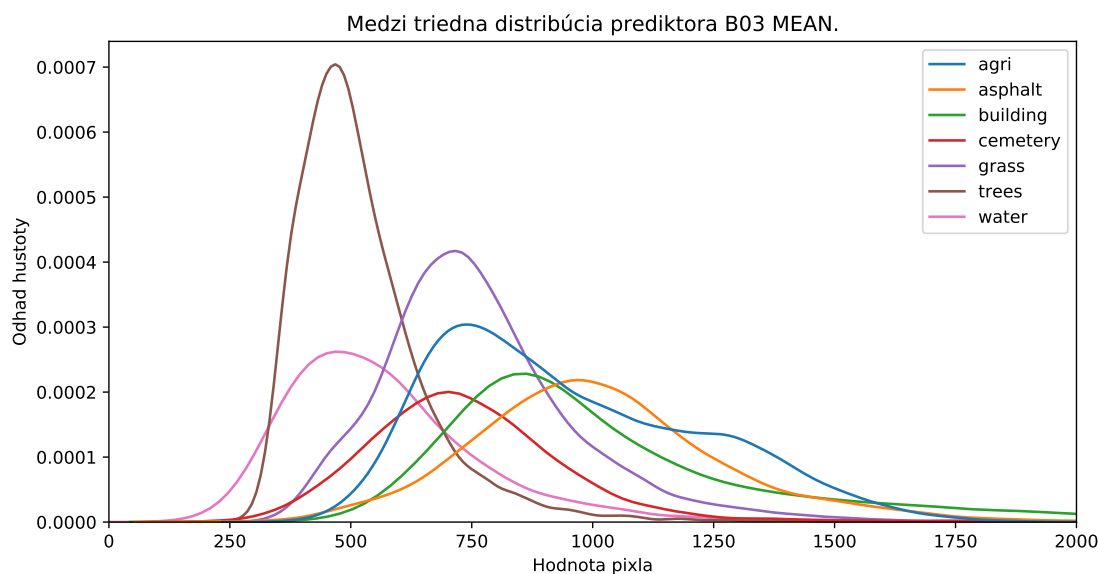
Predpokladané výsledky vyplývajúce z analýz rozložení pixelov v daných spektrálnych pásmach medzi triedami a to konkrétne, že algoritmy budú vykazovať viditeľnú mieru zámennosti medzi triedami *agri* a *grass*, *building* a *asphalt* sa ukázali byť pravdivé ako sa čitateľ môže presvedčiť sám na obrázku 5.1 reprezentujúcom konfúznou maticu (*confussion matrix*) kompaktné zaznamenávajúcu najlepší výsledok klasifikácie získaný algoritmom XGBoost spolu so sledovanými metrikami *recall*, *precision* a *F1 score* určenými pre každú triedu ale aj celkovou testovacou *accuracy* a priemerným *F1 score*. V danej konfúznej matici možno vidieť aj to, že trieda *water* je najľahšie odlíšiteľná a klasifikovateľná čo je doložené hodnotami metrick *recall* a *precision* postupne 87.39% a 88.02%, ktoré sú najvyššie spomedzi ostaných skúmaných tried. Porovnanie výsledkov troch algoritmov získaných po ladení hyperparametrov je zhrnuté v tabuľke 5.1.

		agri	asphalt	building	cemetery	grass	trees	water	Suma(skutoč)	Recall			
Skutočné	agri	2038 14.17%	15 0.10%	18 0.13%	93 0.65%	391 2.72%	48 0.33%	14 0.10%	2617 18.20%	77.88%	0.00%	agri	
	asphalt	14 0.10%	1282 8.92%	353 2.45%	103 0.72%	44 0.31%	17 0.12%	10 0.07%	1823 12.68%	70.32%	0.00%	asphalt	
	building	35 0.24%	380 2.64%	1368 9.51%	77 0.54%	22 0.15%	6 0.04%	11 0.08%	1899 13.21%	72.04%	0.00%	building	
	cemetery	27 0.19%	39 0.27%	31 0.22%	978 6.80%	75 0.52%	84 0.58%	76 0.53%	1310 9.11%	74.66%	0.00%	cemetery	
	grass	329 2.29%	122 0.85%	38 0.26%	128 0.89%	1560 10.85%	289 2.01%	15 0.10%	2481 17.25%	62.88%	0.00%	grass	
	trees	44 0.31%	29 0.20%	18 0.13%	125 0.87%	286 1.99%	2022 14.06%	60 0.42%	2584 17.97%	78.25%	0.00%	trees	
	water	9 0.06%	30 0.21%	26 0.18%	85 0.59%	22 0.15%	38 0.26%	1455 10.12%	1665 11.58%	87.39%	0.00%	water	
Suma(pred)	2496 17.36%	1897 13.19%	1852 12.88%	1589 11.05%	2400 16.69%	2504 17.41%	1641 11.41%	14379 100%				Celkovo	
Precision	81.65%	67.58%	73.87%	61.55%	65.00%	80.75%	88.67%		74.43%			Accuracy	
F1 score	79.72%	68.92%	72.94%	67.47%	63.92%	79.48%	88.02%				74.35%	mean F1	
		agri	asphalt	building	cemetery	grass	trees	water	Celkovo	Accuracy	mean F1		
		Predikované											

Obr. 5.1: Konfúzna matica zachytávajúca výsledok klasifikácie algoritmu XGBoost implementujúci *gradient boosting* trénovaným na podvzorkovanom datasete s 5200 vzorkami na triedu pričom každá vzorka bolo obsiahnutá len v 1 časovom okamihu. Klasifikátor dosiahol hodnotu **74.43%** v prípade celkovej testovacej *accuracy*. Podobne môžeme vidieť, že trieda *water* je najľahšie odlišiteľná a klasifikovateľná čo potvrdzuje hodnota metriky *recall* (87.39%) a *precision* (88.67%) resp. ich harmonický priemer *F1 score* (88.02%). Klasifikátor na druhú stranu vykazuje najväčší problém s detekciou a odlíšením triedy *grass*, ktorá je vo veľkej miere zamieňaná s triedami *agri* a *trees* ale aj triedami *cemetery* a *asphalt*. Táto zámennosť bola ale predpokladaná.

		agri	asphalt	building	cemetery	grass	trees	water	Suma(skutoč)	Recall		
Skutočné	agri	2182 15.17%	6 0.04%	9 0.06%	24 0.17%	336 2.34%	57 0.40%	3 0.02%	2617 18.20%	83.38%	0.00%	agri
	asphalt	15 0.10%	1327 9.23%	335 2.33%	52 0.36%	63 0.44%	26 0.18%	5 0.03%	1823 12.68%	72.79%	0.00%	asphalt
	building	34 0.24%	306 2.13%	1488 10.35%	25 0.17%	34 0.24%	9 0.06%	3 0.02%	1899 13.21%	78.36%	0.00%	building
	cemetery	23 0.16%	27 0.19%	21 0.15%	1000 6.95%	70 0.49%	98 0.68%	71 0.49%	1310 9.11%	76.34%	0.00%	cemetery
	grass	280 1.95%	88 0.61%	32 0.22%	30 0.21%	1792 12.46%	255 1.77%	4 0.03%	2481 17.25%	72.23%	0.00%	grass
	trees	47 0.33%	22 0.15%	11 0.08%	44 0.31%	243 1.69%	2191 15.24%	26 0.18%	2584 17.97%	84.79%	0.00%	trees
	water	10 0.07%	16 0.11%	19 0.13%	63 0.44%	14 0.10%	26 0.18%	1517 10.55%	1665 11.58%	91.11%	0.00%	water
	Suma(pred)	2591 18.02%	1792 12.46%	1915 13.32%	1238 8.61%	2552 17.75%	2662 18.51%	1629 11.33%	14379 100%			Celkovo
	Precision	84.21%	74.05%	77.70%	80.78%	70.22%	82.31%	93.12%		79.96%		Accuracy
	F1 score	83.79%	73.42%	78.03%	78.49%	71.21%	83.53%	92.11%			80.08%	mean F1
		agri	asphalt	building	cemetery	grass	trees	water	Celkovo	Accuracy	mean F1	
		Predikované										

Obr. 5.2: Konfúzna matica po experimente na modely XGBoost s využitím dodatočných členov časovej rady ako nových spektrálnych prediktorov. Takýto prístup priniesol zlepšenie v takmer každej sledovanej metrike.



Obr. 5.3: Medzi triedna distribúcia prediktora výberový priemer pásma B03. Na grafe vidíme jasnú variáciu v rozložení výberových priemerov medzi skúmanými triedami. Podobnosť v rozloženiach týchto priemerov triedy *asphalt* a *building* naznačuje možnú zhoršenú separabilitu týchto dvoch tried medzi sebou.

5.1.2.2 Nevyvážený dataset

Problém s nevyváženým datasetom sme riešili dvoma spôsobmi a síce už spomínaným podvzorkovaním a priradovaním váh proporčne k zastúpeniu jednotlivých tried v tréningovom datasete podľa vzťahu

$$\frac{N_s}{N_c N_i(y)} \quad (5.1)$$

kde N_s vyjadruje počet vzoriek v datasete, N_c je počet tried a $N_i(y)$ je počet vzoriek patriacej do i -tej triedy v datasete. Využitie váh bolo testované len na algoritmoch *random forest* a *gradient boosting* nakoľko pri algoritme *k-NN* váženie nemá zmysel.

Jednou s nevýhod použitia podvzorkovania je to, že môže dôjsť k strate relevantnej informácie a zhoršeniu kvality klasifikátora. Testami na nevyváženom resp. nepodvzorkovanom datasete sme odhalili, že toto sa zrejme stalo aj v našom prípade nakoľko použitie takpovediac plného datasetu prinieslo zlepšenie metriky *accuracy* ako aj priemerného *F1 score* v prípade všetkých troch klasifikátorov asi o 2% oproti prístupu s podvzorkovaným datasetom použitým pri hľadaní a ladení hyperparametrov modelov. Použitie váh neprinieslo zlepšenie oproti prístupu s podvzorkovaním a ani bez podvzorkovania. Navyváženosť datasetu teda asi nie je významným faktorom ovplyvňujúci klasifikáciu v našom prípade. Algoritmus XGBoost založený na *gradient boosting*-u opäť priniesol najlepšie klasifikačné výsledky. Porovnanie výsledkov na základe celkovej metriky *accuracy* pri použití a nepoužití podvzorkovania ako aj použitia váženia jednotlivých tried sú zhrnuté v tabuľke 5.2. Treba podotknúť, že sa na učenie aj v tomto prípade použil len jeden člen časovej rady. Všetky algoritmy vykazovali značné pretrénovanie.

Celková testovacia accuracy			
	<i>k</i> -NN	Random forest	XGBoost
Priradenie váh jednotlivým triedam	X	74.21%	75.79%
Podvzorkovanie trénovacieho datasetu (5200 vzoriek na triedu)	69.21%	72.28%	74.43%
Žiadna úprava nevyváženého trénovacieho datasetu	71.25%	74.40%	76.16%

Tabuľka 5.2: Porovnanie výsledkov klasifikácie na základe metriky *accuracy* pri použití resp. nepoužití podvzorkovania a váženia. Najlepšie výsledky boli dosiahnuté s použitím nevyváženého datasetu. XGBoost opäť dosiahol najlepšie výsledky spomedzi porovnávaných klasifikátorov.

5.1.2.3 Analýza spektrálnych prediktorov a ich dôležitosť

Ďalšie testy a experimenty zahŕňali analýzu dôležitosti jednotlivých použitých prediktorov za účelom zjednodušenia modelu pri zachovaní alebo zlepšení predikčnej a generalizačnej schopnosti klasifikátora. Taktiež je známe, že výber vhodných prediktorov môže významne znížiť čas na učenie.

Vizuálnou analýzou rozloženia konkrétneho prediktora porovnávaných medzi triedami sme vypozerovali, že významný rozdiel nastáva len v rozloženiach vypočítaných výberových priemerov pričom štandardné odchýlky vykazujú veľmi podobné distribúcie. Toto by mohlo napovedať, že štandardné odchýlky použité ako prediktor nebudú významne prispievať k výslednej klasifikácii a sú v podstate nadbytočné. Príklad rozloženia prediktora výberový priemer pásma B03 porovnaný medzi triedami je zobrazený na obrázku 5.3.

Ďalej sme na určenie dôležitosti prediktorov pre jednotlivé algoritmy použili metódu známu ako *permutation importance* alebo *mean decrease accuracy*. Táto metóda umožňuje odhad dôležitosti prediktorov použité daným klasifikátorom bez nutnosti opakovaného trénovania čo môže ušetriť drahocenný čas. Hlavnou myšlienkou za touto metódou je predpoklad zníženia celkovej metriky merajúcej predikčnú silu klasifikátora, v našom prípade *accuracy*, pri náhodnej permutácii hodnôt skúmaného prediktora pri zachovaní ostatných prediktorov ako aj *ground truth*. Daná metóda ale môže priniesť veľmi podhodnotený odhad dôležitosti prediktora v prípade výraznej multikolinearity resp. korelácie prediktorov čo je pravda aj v našom prípade. Odhad permutačných dôležitostí pre algoritmus *k*-NN je uvedený v prílohe 5.2.2.3. Z tohto odhadu sa zdá, že štandardné odchýlky majú v porovnaní s výberovými priermi minimálnu výpovednú hodnotu. Odhady pre algoritmus *random forest* a *gradient boosting* majú podobný vzhľad. Zaujímavý a neočakávaný výsledok tohto odhadu je napríklad, že pásma B11 a B12 resp. ich výberové priemery sa v najväčšej miere podieľajú na rozhodnutí o klasifikácii. Tieto dve pásma by ale nemali mať tak významný vplyv nakoľko slúžia na detekciu oblačnosti. Podobný záver plyní aj o indexe NBRI, ktorý v praxi slúži na detekciu oblastí zasiahnutých požiarom a nemal by teda významne prispievať k rozhodnutiu o klasifikácii. Kvôli týmto zvláštnym záverom ale aj kvôli významnej korelácii treba brať tento permutačný odhad dôležitostí prediktorov s rezervou.

Najväčšiu výpovednú hodnotu má ale v konečnom dôsledku samotný experiment s rôznymi kombináciami použitých prediktorov. Niektoré z týchto testovaných kombinácií spolu s výslednou testovacou *accuracy* pre všetky tri skúmané algoritmy sú uvedené v tabuľke 5.3. Ako vidíme v

tabuľke tak použitie všetkých 28 prediktorov prinieslo najlepšie výsledky celkovej testovacej *accuracy* v prípade algoritmu *random forest* a *gradient boosting*. V prípade algoritmu *k-NN* bolo dosiahnuté najvyššej metriky *accuracy* pri použití len 26 prediktorov zahŕňajúcich všetky prediktory okrem prediktorov súvisiacich s indexom NBRI. V prípade použitia len výberových priemerov všetkých pásiem boli dosiahnuté mierne nižšie hodnoty metriky *accuracy* v porovnaní s použitím všetkých prediktorov. Experimenty s použitím len štandardných odchýlok spektrálnych pásiem a indexov a len s použitím RGB pásiem prinieslo rapídne zhoršenie výslednej metriky *accuracy* v prípade všetkých troch algoritmov. Zaujímavý a očakávaný výsledok priniesol aj experiment s použitím všetkých prediktorov okrem prediktorov vypočítaných na pásmach B11, B12 a indexu NBRI. V prípade tohto nastavenia boli výsledky testovacej *accuracy* porovnateľné s výsledkom získaným pri všetkých prediktorech a teda poukazujúci na pravdepodobnú redundanciu prediktorov založených na pásmach B11, B12 a indexe NBRI. Výsledky získané pri experimentovaní s rôznymi nastaveniami použitých prediktorov môžu opäť slúžiť len ako náznak relevantnosti prediktorov nakoľko závery by vyžadovali exaktnú štatistickú kvantifikáciu na nejakej úrovni významnosti. Takéto štatistické hodnotenie je ale nad rámec našej práce a preto sa uspokojíme so záverom, že nemá zmysel nevyužívať všetkých 28 prediktorov napriek ich možnej redundancii pretože čas učenia nebol významnou limitáciou v našich testoch a nepoužitie všetkých 28 prediktorov neprinieslo viditeľné zlepšenie sledovanej metriky *accuracy* v prípade žiadneho zo sledovaných algoritmov. V prípade potreby urýchlenia a zjednodušenia modelov navrhujeme vypustiť ako prediktory štandardné odchýlky a prediktory vypočítané pomocou pásiem B11, B12 a indexu NBRI.

5.1.2.4 Použitie časovej rady

Posledným z experimentov vykonaných na troch klasifikátoroch bol vzhľadom k použitiu 4-člennej časovej rady vzoriek. V základnom nastavení sme porovnávali a trénovali klasifikátory len s použitím jedného člena časovej rady. Ďalšou možnosťou, ktorá sa naskytla bolo 4-násobné zväčšenie datasetu tak, že každá vzorka bola obsiahnutá vo všetkých štyroch časových okamihoch. Toto v konečnom dôsledku slúžilo ako augmentácia datasetu. Poslednou možnosťou ako využiť časovú radu bolo použiť každý časový okamih ako prediktor tak že vo výsledku má každá vzorka 112 prediktorov.

Tieto tri prístupy boli analyzované za použitia datasetu bez podvzrokovania resp. bez váhena nakoľko toto podľa už prezentovaných výsledkov neprinieslo zlepšenie. Opäť boli testované algoritmy pri využití týchto troch prístupov zoradené od najlepšieho po najhorší ako *k-NN*, *random forest* a *gradient boosting*. Okrem toho všetky algoritmy vykazovali značné pretrénovanie.

Analýza troch prístupov ukazuje, že použitie všetkých 4 členov časovej rady na zväčšenie datasetu a teda akúsi augmentáciu prináša mierne horšie výsledky oproti použitiu len jedného člena časovej rady. Okrem toho toto zväčšenie datasetu prinieslo aj niekoľko násobné predĺženie učenia algoritmov.

Na druhú stranu použitie časovej rady na generovanie prediktorov sa ukázalo ako prínosné a zvýšilo metriku *accuracy* ako aj *F1 score* pri všetkých troch algoritmoch. Dosiahnuté boli postupne pre *k-NN*, *random forest* a *gradient boosting* hodnoty **75%**, **77.21%** a **79.96%**. V prípade algoritmov *random forest* a *gradient boosting* sme nezaznamenali predĺženie času učenia oproti prístupu len s 28 prediktormi čo sa ale nedá povedať o algoritme *k-NN*. Toto predĺženie vzhľadom k zväčšeniu dimenzie priestoru prediktorov z 28 na 112 sa ale neukázalo ako limitácia pre použitie algoritmu *k-NN* pretože bolo porovnateľné s časom potrebným na učenie modelu XGBoost.

Prístup s použitím časových prediktorov je ale v praxi problematický pretože vyžaduje prístup k rozsiahlemu datasetu obsahujúcemu časovú radu snímkov, ktoré ale nemusia byť prístupné napr.

Celková testovacia accuracy			
	<i>k-NN</i>	<i>Random forest</i>	<i>XGBoost</i>
<i>Použitých všetkých 28 prediktorov</i>	71.25%	74.40%	76.16%
<i>Použitie len výberové priemery</i>	69.51%	71.93%	72.40%
<i>Použitie len štandardné odchýlky</i>	54.83%	58.48%	60.57%
<i>Použitie len spektrálne pásma bez spektrálnych indexov</i>	67.31%	71.31%	72.25%
<i>Použitie len spektrálne indexy bez spektrálnych pásiem</i>	67.37%	70.01% *	70.12%
<i>Použitie len RGB (B04, B03, B02) pásma</i>	61.29%	61.43% *	61.29%
<i>Všetky prediktory bez NBRI</i>	71.42%	74.25%	75.90%
<i>Všetky prediktory bez NDVI</i>	70.66%	73.75%	75.38%
<i>Všetky prediktory bez B11, B12 a NBRI</i>	71.00%	74.14%	75.47%

Tabuľka 5.3: Porovnanie výsledkov klasifikácie na základe metriky *accuracy* pri použití rôznych kombinácií spomedzi 28 prediktorov. Zdá sa, že štandardné odchýlky nesú minimálnu informáciu relevantnú pre klasifikáciu v porovnaní s výberovými priemermi. Podobný záver plynie aj o prediktorech súvisiacich s pásmami B11, B12 a NBRI. Použitie spektrálnej informácie len s RGB pásiem zjavne nepostačuje. (*Bolo potrebné zmeniť hodnotu hyperparametra *max_features* z hodnoty 14 na 6 pričom v tomto prípade bola celková *accuracy* pri použití všetkých prediktorov 74.25%.)

Celková testovacia accuracy			
	<i>k</i> -NN	Random forest	XGBoost
Len 1 člen časovej rady snímky	71.25%	74.40%	76.16%
Všetky 4 členy použité ako samostatné vzorky (zväčšenie datasetu/augmentácia)	70.94%	73.56%	75.47%
Všetky 4 členy použité ako nové prediktory	75.05%	77.21%	79.96%

Tabuľka 5.4: Porovnanie výsledkov klasifikácie na základe metriky *accuracy* pri použití len jedného člena časovej rady vzorky a všetkých 4 členov ako nové vzorky alebo ako prediktory. Použitie všetkých 4 členov časovej rady ako dodatočné prediktory sa ukázalo ako najlepšie. XGBoost opäť vykazuje najvyššiu hodnotu metriky *accuracy* spomedzi testovaných algoritmov.

kvôli oblačnosti alebo nedostatočnému časovému rozlíšeniu snímiek. Každopádne sa tento prístup ukázal ako najlepší v porovnaní s ostatnými tu prezentovanými prístupmi a je dosť možné, že pri optimálnom nastavení a prístupe k dostatočne veľkej časovej rade snímok môže priniesť ešte výraznejšie zlepšenie v klasifikácii povrchov. Skúmanie tohto prístupu môže byť námetom pre ďalšiu prácu. Porovnanie a zhrnutie výsledkov pri použití časovej rady snímok sú uvedené v tabuľke 5.4. Na obrázku 5.2 môžeme vidieť konfúznou maticu po evaluácii modelu XGBoost na testovacom datasete po využití všetkých štyroch členov časovej rady ako nových prediktorov.

5.2 Sémantická segmentácia

Sémantická segmentácia povrchov v satelitných snímkach je častá úloha v oblasti vzdialeného prieskumu Zeme a preto sa tejto problematike venujeme aj v našej práci. Jednak vyskúšame prístup založený na spektrálnych prediktorech ako pri riešení klasifikačnej úlohy a to za pomoci algoritmu *gradient boosting*. Ale tentoraz sa pokúsime využiť aj priestorovú informáciu obsiahnutú v rozložení jednotlivých pixelov obrázka a to s využitím pokročilej architektúry plne konvulčnej neurónovej siete.

5.2.1 Metodika

Pre učenie bol pripravený dataset obsahujúci zhruba 147 000 obrázkov spolu s prislúchajúcimi segmentačnými maskami rozmerov 122 x 122 pixelov. Tento dataset bol dodatočne prefiltrovaný a boli vylúčené obrázky so zastúpením viac ako 50% pixelov označených ako *Unknown*. Tieto pixely neboli využité ani pri učení ani pri evaluácii modelov a určovaní výsledných metrick. Samotná klasifikácia pixelov triedy *Unknown* nebola hodnotená. Zhodnotenie presnosti klasifikácie pixelov tejto triedy bude vyžadovať dodatočný externý dataset s informáciou o *ground truth* všetkých pixelov. Ako sme videli tak aj trieda *Nodata* má vysoké zastúpenie v počte pixelov. Tieto pixely sme nevyklúčili z procesu učenia a vo výsledku teda dataset obsahoval 116 723 obrázkov.

V prípade neurónovej siete boli využité obrázky ako aj segmentačná maska v celku pričom na ne bola aplikovaná sada predspracovaní zahŕňajúc štandardizáciu vzhľadom k priemeru a štandardnej odchýlke každého spektrálneho pásma. Samotný priemer a štandardná odchýlka bola získaná výpočtom nad celým tréningovým datasetom pre každé spektrálne pásmo zvlášť. Ďalšie pred-

spracovanie bolo tzv. *padding* nulami po okrajoch (3 pixely) aby mal výsledný obrázok rozmery 128x128 pixelov čo vyžadovala architektúra použitej neurónovej siete. Okrem štandardizácie boli testované aj iné škálovacie techniky napr. voči L2 norme alebo maximum ale tieto nepriniesli zlepšenie oproti štandardizácii. Taktiež sme skúšali štandardizáciu lokálne na každom obrázku čo má význam pri väčších odchýlkach v jase medzi obrázkami. Tento prístup ale nepriniesol zlepšenie čo mohlo byť spôsobené nie príliš veľkou variáciou v jase obrázkov predovšetkým kvôli malej časovej rade snímok.

V prípade *gradient boosting*-u bola vykonaná len *padding* na 128x128 pixelov pretože štandardizácia ako v prípade neurónovej siete nepriniesla zlepšenie. Samotný *padding* nemal pre tento algoritmus žiaden význam ale bol vykonaný hlavne kvôli zhodnému počtu pixelov v datasete pri validácii. Učenie algoritmu XGBoost prebiehalo na jednotlivých pixeloch obrázka, ktoré tak boli použité ako vzorky pričom ako spektrálne prediktory slúžili hodnoty intenzít týchto pixelov resp. odrazivosti v jednotlivých použitých spektrálnych pásmach spolu s použitými indexami.

V základnom nastavení sme pre oba modely použili spektrálne pásma a indexy menovite B02, B03, B04, B05, B06, B07, B11, B12, NDVI, NDWI, NDBI. Teda sme vypustili oproti klasifikačnej úlohe v predchádzajúcej sekcii indexy NBRI a NDMI, ktoré považujeme za nepotrebné ako bolo prezentované aj v časti o klasifikácii. Pásma B11 a B12 boli ponechané nakoľko by mali slúžiť na detekciu oblačnosti o čo sa snažíme aj v našej segmentačnej úlohe. Učenie neurónovej siete prebiehalo bez použitia *transfer learningu* resp. predtrénovaného modelu.

Použité metriky boli opäť *recall*, *precision*, *F1 score* zvané aj *dice coefficient*, celkovú *accuracy* ale hlavnou sledovanou metriku bol *mIoU*. Sledovanie metriky *accuracy* ako hlavnej metriky sa neukázalo ako rozumné vzhľadom k výraznejšej nevyváženosti datasetu. Jednotlivé metriky boli popísané v sekcii 3.1.1.4. Ako už bolo spomenuté tak pri výpočtoch metrik boli vylúčené pixely triedy *Unknown*.

Použité algoritmy boli vybrané na základe predchádzajúcich výsledkov pri klasifikácii jednotlivých snímok. Na segmentáciu boli teda vybrané algoritmy *random forest* a *gradient boosting* v implementácii XGBoost. Okrem týchto algoritmov sme využili ešte model plne konvolučnej neurónovej siete v architektúre U-Net popísaný v časti 3.4.1.1.

Po počiatkových testoch sa ale ukázala nevýhoda prístupu s pixelami ako vzorkami na trénovanie pretože to znamenalo zhruba miliardu pixelov na trénovanie. Takéto učenie bolo veľmi nepraktické v prípade algoritmu *random forest*, ktorého evaluácia trvala niečo cez hodinu aj pri použití distribuovaného nastavenia, čo bola niekoľkokrát dlhšie v porovnaní s evaluáciou *gradient boosting*-u. Práve kvôli časovej limitácii ale aj predpokladu, že *random forest* neprinesie lepšie výsledky ako XGBoost sme s týmto algoritmom ďalej neexperimentovali. V konečnom dôsledku boli testy a experimenty vykonávané na modely plne konvolučnej siete U-Net a algoritme XGBoost.

Rozdelenie datasetu prebehlo štandardným spôsobom na tri časti v pomere 64:16:20 na trénovací, validačný a testovací. Vzorkovanie do jednotlivých datasetov prebiehalo náhodne. Na validačný a testovací dataset boli ďalej aplikované úpravy aby sa zabránilo prítomnosti duplicitných snímok rovnakého územia zaznamenaných v rôznych časoch ale aj snímky rovnakého územia v rôznych časoch vzhľadom k trénovaciemu datasetu. Po aplikácii všetkých týchto dodatočných úprav obsahoval výsledný trénovací dataset 74702 snímok, ďalej validačný dataset 2127 snímok a nakoniec testovací dataset 2439 snímok.

Implementácia algoritmov *random forest* a *gradient boosting* prebiehala opäť pomocou knižníc scikit-learn a XGBoost so scikit-learn API. Samotná neurónová sieť v architektúre U-Net ako aj niektoré účelové funkcie (*loss function*) boli prebraté z knižnice *segmentation_models_pytorch* [84] obsahujúcej implementáciu niektorých architektúr neurónových sietí používaných na sémantickú segmentáciu ako aj predtrénované modely. Dodatočné úpravy architektúry siete, implementácia doplnkového modulu pre interakciu s datasetom ale aj úpravy v účelových funkciách boli vykonané v rámci frameworku Pytorch [65]. Vzhľadom k potrebe akcelerácie a distribuovaného učenia v prípade neurónovej siete bol využitý framework RaySGD obsahujúci objekty umožňujúce distribuované učenie modelov z framework-ov Pytorch a TensorFlow. Na samotné distribuované učenie neurónovej siete boli použité 2 grafické karty a to konkrétne Nvidia Tesla V100 a Nvidia GeForce GTX 2080 Ti. V prípade algoritmu XGBoost bolo učenie vykonané hlavne pomocou 46 paralelných procesorov kvôli limitácii pamäte grafickej karty Nvidia Tesla V100 nakoľko algoritmus nepodporuje iteratívne učenie po tzv. *batch*-och a vyžaduje prístup k celému datasetu naraz. Výpočetné kapacity boli opäť poskytnuté Ústavom teórie informácie a automatizácie AV ČR.

5.2.2 Experimenty a výsledky

V tejto sekcii zhrnieme výsledky experimentov pri použití modelu plne konvolučnej neurónovej siete U-Net ako aj algoritmu XGBoost.

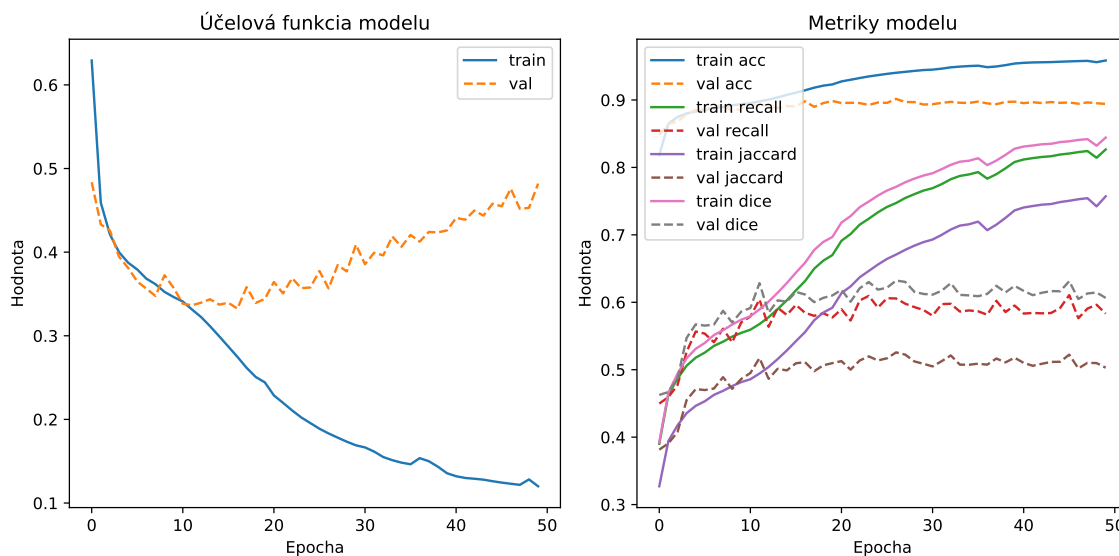
5.2.2.1 Model plne konvolučnej neurónovej siete

V prípade plne konvolučnej siete sme vybrali architektúru U-Net založenú na prístupe *encoder-decoder* popísanej v časti 3.4.1.1. Na optimalizáciu sme použili optimalizátor založený na algoritme Adam [53] pretože už v základnom nastavení priniesol značne lepšie výsledky oproti algoritmu *stochastic gradient descent*. Okrem toho nás pozbavil nutnosti naladiť správnu hodnotu hyperparametra zvaného *learning rate*, ktorý bol optimalizovaný automaticky. Tiež sme skúšali použitie rôznych *batchsize*, pričom zmeny tohto parametra sa neukázali ako podstatné a preto sme ho ponechali na hodnote 128. Tieto základné nastavenie siete boli testované pri použití spektrálnych pásiem popísaných v sekcii o metodike 5.2.1. Ako účelovú funkciu sme v základnom nastavení použili *cross-entropy* pričom do učenia resp. optimalizácie nebol zaradené pixely triedy *Unknown*. Ako *encoder* sme v základnom nastavení siete vybrali konvolučnú sieť v architektúre ResNet 18. Učenie prebiehalo 50 epôch pričom sme od 12. epochy pozorovali oscilácie hodnoty metriky *mIoU* okolo hodnoty 72 % čo sa ďalším tréňovaním nezlepšovalo. Okrem toho sme od 12. epochy pozorovali zvyšovanie hodnoty účelovej funkcie na validačnom datasete a teda pretrénovanie. Najlepší výsledok bol vybraný na základe najvyššej dosiahnutej hodnoty metriky *mIoU* na validačnom datasete. Najlepší výsledok teda nastal po 27. tréňovacej epoche pri hodnote *mIoU* zhruba **50.99%** na validačnom datasete. Daný model bol následne zhodnotený aj za použitia testovacieho datasetu pričom dosiahol mierne vyššiu hodnotu metriky *mIoU* rovnú **51.23%**.

Konfúzna matica po epoche s najvyššou hodnotou validačnej metriky *mIoU* je zobrazená na obrázku 5.4. Z tejto konfúznej matice môžeme vidieť, že model vykazuje veľmi dobré klasifikačné výsledky na triedach *Arable* a *Forest*, ktoré sú zastúpené vo väčšine. Okrem toho očakávaný výsledok na základe výsledkov z klasifikácie je, že trieda *Water* bude taktiež spôsobovať menšie problémy pri segmentácii čo vyplýva z jej špecifickej spektrálnej signatúry. Toto sa potvrdilo hodnotami všetkých pozorovaných metrík. Na druhú stranu na základe hodnôt metriky *IoU* vidíme, že mal model najväčšie problémy s klasifikáciou tried *Vegetation* a *Artificial Vegetation*. Toto mohlo byť spôsobené ich spektrálnou podobnosťou ale aj podobnosťou s inými triedami obsahujúcimi vegetáciu. Krivky učenia daného modelu sú zobrazené na obrázku 5.5.

	unknown	urban	indus	constr	artif_vege	arable	perma_crop	pasture	forest	veget	no_veget	in_wet_land	water	cloud	nodata	Suma(skutoč)	Recall	IoU
unknown	4.87e+05 1.22%	9.20e+04 0.23%	1.70e+04 0.04%	2.77e+04 0.07%	2.10e+06 5.25%	2.23e+04 0.06%	8.11e+05 2.03%	8.03e+05 2.01%	6.79e+04 0.17%	1.47e+03 0.00%	9.63e+03 0.02%	3.12e+04 0.08%	3.03e+05 0.76%	1.74e+02 0.00%	4.78e+06 11.95%	0.00%	0.00%	
urban	1.37e+06 3.42%	3.69e+04 0.09%	1.14e+02 0.00%	1.22e+04 0.03%	8.54e+04 0.21%	8.75e+02 0.00%	3.58e+04 0.09%	4.32e+04 0.11%	4.16e+03 0.01%		7.00e+00 0.00%	1.76e+03 0.00%	3.16e+04 0.08%	1.30e+01 0.00%	1.62e+06 4.05%	84.43%	66.29%	
indus	4.65e+04 0.12%	1.47e+05 0.37%	3.94e+03 0.01%	2.47e+03 0.01%	2.21e+04 0.06%	9.40e+01 0.00%	3.18e+03 0.01%	7.39e+03 0.02%	3.58e+03 0.01%			6.01e+02 0.00%	5.89e+03 0.01%	1.00e+00 0.00%	2.43e+05 0.61%	60.58%	41.43%	
constr	3.10e+03 0.01%	1.25e+04 0.03%	3.32e+04 0.08%	6.57e+02 0.00%	1.20e+04 0.03%	5.00e+00 0.00%	2.81e+03 0.01%	1.21e+04 0.03%	4.61e+03 0.01%	2.61e+02 0.00%	1.50e+02 0.00%	2.01e+03 0.01%	1.44e+03 0.00%	1.00e+00 0.00%	8.48e+04 0.21%	39.14%	32.14%	
artif_vege	6.72e+04 0.17%	4.39e+03 0.01%	1.26e+02 0.00%	5.09e+04 0.13%	1.16e+04 0.03%	2.96e+03 0.01%	1.15e+04 0.03%	2.23e+04 0.06%	3.13e+03 0.01%			1.04e+03 0.00%	4.47e+03 0.01%	5.00e+00 0.00%	1.80e+05 0.45%	28.35%	22.88%	
arable	1.18e+05 0.30%	1.76e+04 0.04%	4.09e+03 0.01%	3.83e+03 0.01%	8.75e+06 21.88%	1.32e+04 0.03%	3.45e+05 0.86%	1.23e+05 0.31%	1.52e+04 0.04%		6.64e+02 0.00%	3.20e+03 0.01%	1.08e+05 0.27%	5.50e+01 0.00%	9.50e+06 23.76%	92.09%	83.47%	
perma_crop	7.88e+03 0.02%	1.13e+02 0.00%	5.00e+00 0.00%	1.80e+03 0.00%	6.40e+04 0.16%	7.75e+04 0.19%	1.42e+04 0.04%	8.06e+03 0.02%	9.60e+02 0.00%		3.00e+01 0.00%	1.20e+01 0.00%	8.20e+03 0.02%		1.83e+05 0.46%	42.40%	34.63%	
pasture	7.13e+04 0.18%	4.64e+03 0.01%	9.67e+02 0.00%	5.23e+03 0.01%	3.47e+05 0.87%	1.24e+04 0.03%	1.67e+06 4.16%	2.06e+05 0.51%	1.65e+04 0.04%	1.79e+02 0.00%	3.46e+03 0.01%	3.11e+03 0.01%	4.56e+04 0.11%	7.00e+00 0.00%	2.38e+06 5.96%	69.94%	54.40%	
forest	5.12e+04 0.13%	4.46e+03 0.01%	3.47e+03 0.01%	6.27e+03 0.02%	1.33e+05 0.33%	3.46e+03 0.01%	1.44e+05 0.36%	1.28e+07 31.96%	2.54e+04 0.06%	1.76e+03 0.00%	2.38e+03 0.01%	1.07e+04 0.03%	1.29e+05 0.32%	6.20e+01 0.00%	1.33e+07 33.24%	96.12%	90.87%	
veget	4.27e+04 0.11%	1.24e+04 0.03%	1.29e+03 0.00%	8.35e+03 0.02%	1.01e+05 0.25%	6.61e+03 0.02%	9.03e+04 0.23%	1.59e+05 0.40%	8.27e+04 0.21%	6.57e+02 0.00%	2.29e+03 0.01%	4.70e+03 0.01%	1.13e+04 0.03%	7.00e+00 0.00%	5.23e+05 1.31%	15.82%	13.69%	
no_veget	4.76e+02 0.00%	1.48e+02 0.00%	6.90e+02 0.00%	1.45e+02 0.00%	2.50e+02 0.00%	1.05e+02 0.00%	4.34e+02 0.02%	6.27e+03 0.02%	3.67e+03 0.01%	2.05e+03 0.01%	4.00e+00 0.00%	7.06e+02 0.00%	2.12e+02 0.00%		1.52e+04 0.04%	13.52%	11.25%	
in_wet_land	6.19e+02 0.00%	4.10e+01 0.00%	1.70e+01 0.00%	1.21e+02 0.00%	5.27e+03 0.01%		1.30e+04 0.03%	4.08e+04 0.10%	1.59e+03 0.00%		1.29e+04 0.03%	2.26e+03 0.01%	4.00e+02 0.00%	2.00e+00 0.00%	7.70e+04 0.19%	16.75%	14.86%	
water	6.51e+03 0.02%	1.60e+03 0.00%	1.83e+03 0.00%	8.76e+02 0.00%	1.16e+04 0.03%	6.40e+01 0.00%	5.93e+03 0.01%	2.73e+04 0.07%	1.67e+03 0.00%	4.10e+01 0.00%	7.11e+02 0.00%	3.47e+05 0.87%	3.44e+03 0.01%	6.00e+00 0.00%	4.08e+05 1.02%	84.91%	76.21%	
cloud	2.69e+04 0.07%	1.73e+04 0.04%	1.94e+03 0.00%	1.08e+03 0.00%	1.89e+05 0.47%	1.24e+03 0.00%	1.47e+04 0.04%	1.13e+05 0.28%	1.01e+05 0.00%	1.50e+02 0.00%	8.10e+01 0.00%	1.64e+04 0.04%	2.20e+06 5.50%	2.00e+00 0.00%	2.58e+06 6.46%	85.20%	75.05%	
nodata	7.60e+01 0.00%	6.00e+00 0.00%		1.00e+00 0.00%	1.19e+02 0.00%		1.60e+01 0.00%	1.21e+02 0.00%	5.00e+00 0.00%			7.90e+01 0.00%	1.20e+01 0.00%	4.12e+06 10.31%	4.12e+06 10.31%	99.99%	99.99%	
Suma(pred)	0.00e+00 0.00%	2.30e+06 5.74%	3.51e+05 0.88%	6.87e+04 0.17%	1.22e+05 0.30%	1.18e+07 29.58%	1.41e+05 0.35%	3.16e+06 7.90%	1.43e+07 35.89%	2.32e+05 0.58%	6.57e+03 0.02%	3.23e+04 0.08%	4.24e+05 1.06%	2.85e+06 7.13%	4.12e+06 10.31%	4.00e+07 100%		
Precision	0.00%	75.53%	56.72%	64.25%	54.22%	89.92%	65.40%	70.99%	84.33%	50.39%	40.21%	56.85%	88.15%	86.29%	100.00%	89.84%		
F1 score	0.00%	79.73%	58.59%	48.65%	37.23%	90.99%	51.45%	70.46%	85.22%	24.09%	20.23%	25.87%	86.50%	85.74%	99.99%	51.23%		
	unknown	urban	indus	constr	artif_vege	arable	perma_crop	pasture	forest	veget	no_veget	in_wet_land	water	cloud	nodata	Celkovo	Accuracy	mIoU

Obr. 5.4: Konfúzna matica zachytávajúca evaluáciu modelu neurónovej siete U-Net s *encoderom* ResNet 18 v základnom nastavení na testovacom datasete. Najvyššie hodnoty metriky *IoU* dosahujú triedy *Arable*, *Forest* a *Water*.



Obr. 5.5: Priebeh učenia modelu U-Net v základnom nastavení. Model začína vykazovať pretrénovanie zhruba od 20. epochy.

Jedným z vykonaných experimentov bolo použitie účelovej funkcie *Focal loss*, ktorá bola špecificky vyvinutá na aplikáciu v detekcii objektov. Táto účelová funkcia modifikuje klasickú *cross-entropy* tak aby priradila väčšiu váhu chybné klasifikovaným vzorkám. Toto v konečnom dôsledku pomáha zmiernovať vplyv nevyváženosti datasetu [57].

Použitie *Focal loss* prinieslo porovnateľné výsledky s účelovou funkciou *cross-entropy* čo sa týka výsledných metrík. Nakoľko sme ale očakávali viditeľné zlepšenie, ktoré nenastalo tak sme *Focal loss* nezaradili do ďalších experimentov a pokračovali sme s použitím *cross-entropy*. Ďalšie experimenty sú popísané v nasledujúcich odstavcoch.

Porovnanie vzhľadom k encoder architektúram sme sa vykonali tak, že sme *encoder* ResNet 18 v modeli U-Net nahradili postupne architektúrami konvolučnej neurónovej siete DenseNet 121 a Xception. Týmto sme chceli aspoň zčasti overiť vhodnosť výberu ResNet 18 ako *encoder-u*. Pri použití oboch dodatočných konvolučných sietí začal vykazovať model pretrénovanie zhruba od 20. epochy. Najlepší výsledok vzhľadom k *mIoU* bolo dosiahnuté s *encoder-om* DenseNet 121 a to **56.52%** na validačnom datasete po 39 epoche z celkovo 80 epóch. Evaluácia na testovacom datasete ukázala mierne nižšiu hodnotu *mIoU* a to **55.64%**. Všeobecne platilo, že triedy s menším zastúpením v datasete boli ťažšie klasifikovateľné danými modelmi, s výnimkou triedy *Water*. Porovnanie vzhľadom k metrikám *mIoU*, priemerného *F1 score* a priemernej *accuracy* sú v tabuľke 5.5. Konfúzna matica zachytávajúca výsledok segmentácie s využitím encodera DenseNet 121 na testovacom datasete je na obrázku 5.6. Na tomto obrázku je viditeľné, že model založený na DenseNet 121 priniesol oproti modelu založenom na ResNet 18 zlepšenie vzhľadom ku všetkým triedam okrem triedy *Water* kde bol ale rozdiel minimálny. Výrazné zlepšenie sa prejavilo hlavne na detekcii málo zastúpených tried ako *Mine*, *dump & construction* (na obrázku ako *constr*), *Permanent crops*, *Vegetation*, *Open spaces with little or no vegetation* (na obrázku ako *no_veget*) a *Inland wet lands* (na obrázku ako *in_wet_land*) čo dokazujú hodnoty metrík *recall* a *IoU*.

	ResNet 18	DenseNet 121	Xception
<i>accuracy</i>	89.84%	90.76%	90.40%
<i>mIoU</i>	51.23%	55.64%	53.93%
<i>F1 score</i>	62.48%	67.62%	65.67%

Tabuľka 5.5: Porovnanie modelu U-Net vzhľadom k použítym *encoder*-om. Najlepší výsledok bol dosiahnutý s *encoder*-om DenseNet 121. Výsledky sú získané po evaluácii na testovacom datasete a v každom prípade po epoche kedy boli dosiahnuté najvyššie hodnoty sledovaných metrík pričom ďalej už nedochádzalo k ich zlepšovaniu.

	unknown	urban	indus	constr	artif_vege	arable	perma_crop	pasture	forest	veget	no_veget	in_wet_land	water	cloud	nodata	Suma(skutoč)	Recall	IoU	
unknown	4.93e+05 1.23%	8.59e+04 0.21%	1.71e+04 0.04%	2.51e+04 0.06%	2.25e+06 5.64%	2.83e+04 0.07%	7.60e+05 1.90%	6.87e+05 1.72%	1.25e+05 0.31%	2.85e+03 0.01%	1.66e+04 0.04%	3.08e+04 0.08%	2.56e+05 0.64%	1.06e+02 0.00%	4.78e+06 11.96%	0.00%	0.00%	unknown	
urban	1.41e+06 3.52%	2.36e+04 0.06%	2.43e+02 0.00%	8.33e+03 0.02%	8.11e+04 0.20%	1.56e+03 0.00%	3.36e+04 0.08%	3.21e+04 0.08%	6.80e+03 0.02%	2.00e+00 0.00%	1.20e+01 0.00%	1.40e+03 0.00%	2.17e+04 0.05%	2.10e+01 0.00%	1.62e+06 4.05%	86.98%	68.84%	urban	
indus	5.13e+04 0.13%	1.46e+05 0.35%	1.88e+03 0.00%	1.77e+03 0.00%	2.10e+04 0.05%	6.30e+01 0.00%	2.80e+03 0.01%	5.68e+03 0.01%	4.79e+03 0.01%	1.60e+01 0.00%	7.00e+00 0.00%	4.93e+02 0.00%	7.20e+03 0.02%	4.00e+00 0.00%	2.43e+05 0.61%	60.01%	44.57%	indus	
constr	3.19e+03 0.01%	9.65e+03 0.02%	4.52e+04 0.11%	8.91e+02 0.00%	7.43e+03 0.02%	3.78e+02 0.00%	1.76e+03 0.00%	9.00e+03 0.02%	3.24e+03 0.01%	4.45e+02 0.00%		2.31e+03 0.01%	1.30e+03 0.00%		8.48e+04 0.21%	53.32%	42.14%	constr	
artif_vege	7.06e+04 0.18%	3.09e+03 0.01%	1.66e+02 0.00%	5.79e+04 0.14%	1.09e+04 0.03%	2.26e+03 0.01%	8.08e+03 0.02%	1.84e+04 0.05%	5.02e+03 0.01%	1.00e+00 0.00%	4.50e+01 0.00%	1.02e+03 0.00%	2.23e+03 0.01%	5.00e+00 0.00%	1.80e+05 0.45%	32.20%	26.66%	artif_vege	
arable	1.12e+05 0.28%	1.50e+04 0.04%	4.69e+03 0.01%	3.21e+03 0.01%	8.86e+05 22.16%	2.60e+04 0.06%	2.69e+05 0.67%	8.44e+04 0.21%	2.29e+04 0.06%		2.42e+03 0.01%	1.87e+03 0.00%	9.56e+04 0.24%	7.20e+01 0.00%	9.50e+06 23.76%	93.29%	84.63%	arable	
perma_crop	7.09e+03 0.02%	6.60e+01 0.00%	1.79e+02 0.00%	1.62e+03 0.00%	4.61e+04 0.12%	1.05e+05 0.26%	1.04e+04 0.03%	5.97e+03 0.01%	2.12e+03 0.01%		9.40e+01 0.00%	1.60e+01 0.00%	4.41e+03 0.01%	2.00e+00 0.00%	1.83e+05 0.46%	57.33%	42.91%	perma_crop	
pasture	5.80e+04 0.15%	4.01e+03 0.01%	1.88e+03 0.00%	5.26e+03 0.01%	3.66e+05 0.91%	1.67e+04 0.04%	1.71e+06 4.27%	1.48e+05 0.37%	4.28e+04 0.11%	1.44e+02 0.00%	7.30e+03 0.02%	2.98e+03 0.01%	1.18e+04 0.03%	4.00e+00 0.00%	2.37e+06 5.94%	71.98%	57.95%	pasture	
forest	5.32e+04 0.13%	4.72e+03 0.01%	6.03e+03 0.02%	6.86e+03 0.02%	1.51e+05 0.38%	6.10e+03 0.02%	1.42e+05 0.36%	1.28e+07 31.95%	4.73e+04 0.12%	1.17e+03 0.00%	8.30e+03 0.02%	1.26e+04 0.03%	7.58e+04 0.19%	6.80e+01 0.00%	1.33e+07 33.24%	96.13%	91.91%	forest	
veget	4.29e+04 0.11%	1.13e+04 0.03%	3.12e+03 0.01%	7.19e+03 0.02%	1.00e+05 0.25%	7.04e+03 0.02%	6.51e+04 0.16%	1.28e+05 0.32%	1.36e+05 0.34%	1.35e+03 0.00%	9.12e+03 0.02%	4.08e+03 0.01%	7.56e+03 0.02%	5.00e+00 0.00%	5.23e+05 1.31%	25.98%	20.26%	veget	
no_veget	5.32e+02 0.00%	2.14e+02 0.00%	1.79e+02 0.00%	1.34e+02 0.00%	1.91e+02 0.00%	1.19e+02 0.00%	2.52e+02 0.00%	6.06e+03 0.02%	2.29e+03 0.01%	4.26e+03 0.01%	1.25e+02 0.00%	4.60e+02 0.00%	3.43e+02 0.00%		1.52e+04 0.04%	28.12%	22.95%	no_veget	
in_wet_land	4.69e+02 0.00%	3.70e+01 0.00%	1.10e+01 0.00%	8.90e+01 0.00%	5.53e+03 0.01%	5.40e+01 0.00%	8.94e+03 0.02%	3.07e+04 0.08%	4.36e+03 0.01%		2.35e+04 0.06%	2.98e+03 0.01%	2.89e+02 0.00%	3.00e+00 0.00%	7.70e+04 0.19%	30.51%	21.91%	in_wet_land	
water	6.71e+03 0.02%	1.59e+03 0.00%	2.64e+03 0.01%	6.37e+02 0.00%	1.05e+04 0.03%	3.90e+01 0.00%	4.32e+03 0.01%	2.08e+04 0.05%	3.78e+03 0.01%	1.58e+02 0.00%	1.62e+03 0.00%	3.37e+05 0.84%	1.84e+04 0.05%	1.50e+01 0.00%	4.08e+05 1.02%	82.56%	75.84%	water	
cloud	2.03e+04 0.05%	1.08e+04 0.03%	1.48e+03 0.00%	1.35e+03 0.00%	1.72e+05 0.43%	1.19e+03 0.00%	2.83e+04 0.07%	1.19e+05 0.30%	2.19e+03 0.01%	1.27e+02 0.00%	1.17e+03 0.00%	5.94e+03 0.01%	2.22e+06 5.55%	5.10e+01 0.00%	2.58e+06 6.47%	85.90%	78.41%	cloud	
nodata	1.80e+01 0.00%	1.00e+00 0.00%			7.30e+01 0.00%		1.00e+00 0.00%	2.80e+01 0.00%				2.90e+01 0.00%	9.20e+01 0.00%	4.12e+06 10.31%	4.12e+06 10.31%	99.99%	99.99%	nodata	
Suma(pred)	0.00e+00 0.00%	2.33e+06 5.82%	3.16e+05 0.79%	8.48e+04 0.21%	1.20e+05 0.30%	1.21e+07 30.23%	1.95e+05 0.49%	3.04e+06 7.61%	1.41e+07 35.19%	4.09e+05 1.02%	1.05e+04 0.03%	7.03e+04 0.18%	4.04e+05 1.01%	2.72e+06 6.81%	4.12e+06 10.31%	4.00e+07 100%			Celkovo
Precision	0.00%	76.75%	61.41%	66.77%	60.78%	90.12%	63.05%	74.83%	95.45%	47.92%	55.54%	43.73%	90.31%	90.00%	99.99%		90.76%		Accuracy
F1 score	0.00%	81.55%	61.66%	59.29%	42.10%	91.68%	60.05%	73.38%	95.79%	33.70%	37.34%	35.94%	86.26%	87.90%	99.99%			55.64%	mIoU

Obr. 5.6: Konfúzna matica zachytávajúca výsledok učenia neurónovej siete U-Net s *encoder*-om DenseNet 121, ktorý sa ukázal ako najlepší spomedzi porovnávaných *encoder*-ov.

	<i>Použité všetky spektrálne pásma vrátane indexov NDVI, NDWI a NDBI</i>	<i>Použité len spektrálne pásma bez indexov</i>	<i>Použité len RGB pásma (B04, B03, B02)</i>
<i>accuracy</i>	90.76%	90.50%	88.42%
<i>mIoU</i>	55.64%	55.14%	49.41%
<i>F1 score</i>	67.62%	67.14%	61.76%

Tabuľka 5.6: Porovnanie modelu U-Net s encoderom DenseNet 121 vzhľadom k použitým spektrálnym pásmam.

Porovnanie vzhľadom k použitým spektrálnym pásmam malo znovu slúžiť na odhad vplyvu a významnosti použitia týchto pásiem ako prediktorov. Vzhľadom k tomu, že sieť U-Net dosiahla najlepší výsledok pri použití *encoder*-a DenseNet 121, tak sme ho v týchto testoch využili namiesto *encoder*-a ResNet 18.

Vynechanie spektrálnych indexov spomedzi prediktorov prinieslo zanedbateľný pokles celkových sledovaných metrík. Použitie spektrálnych indexov má zrejme minimálny vplyv na zlepšenie klasifikačnej schopnosti modelu. Na druhú stranu povšimnutia hodný výsledok je, že použitie len čisto RGB pásiem neprinieslo tak výrazný pokles celkovej klasifikačnej schopnosti modelu ako sme očakávali čo poukazuje na významný vplyv priestorovej informácie a možnosti detekcie vzorov a štruktúr plynúcich z použitia celého obrázka. Optimistický výsledok v prípade experimentu len s čisto RGB pásmami ďalej poukazuje na možnú aplikáciu *transfer learning*-u založenom na využití predtrénovaného modelu napríklad na datasete ImageNet a jeho možný pozitívny vplyv na zlepšenie klasifikačnej schopnosti modelu. Takýto prístup sme v našej práci neuskutočnili vzhľadom k časovej limitácii ale mohol by byť námetom ďalších prác a zlepšení nášho modelu. Výsledky týchto experimentov sú zhrnuté v tabuľke 5.6. Na základe výsledkov sme v ďalších experimentoch pokračovali s použitím všetkých spektrálnych pásiem ako aj troch indexov.

Regularizácia sa ukázala ako potrebná nakoľko doteraz všetky testy poukazovali na výrazné pretrénovanie skúmaných modelov. Regularizáciu pomocou skorého zastavenia sme v našich pokusoch nevyužili ale nechávali sme modely trénovať dlhšie pričom sa nakoniec vybral ten s najlepšou dosiahnutou hodnotou *mIoU*. Prvou z použitých metód regularizácie bolo v podstate použitie jednoduchšieho modelu čo sa týka počtu optimalizovateľných parametrov. Toto sme popísali v odstavci 5.2.2.1 kde sme testovali použitie troch rôznych *encoder*-ov v modeli U-Net. Najmenej komplexný model konvolučnej siete bola architektúra DenseNet 121 s 6×10^6 optimalizovateľnými parametrami a najkomplexnejší model bol s architektúrou Xception s 22×10^6 optimalizovateľnými parametrami. Tento prístup s použitím menej komplexného modelu teda priniesol isté zlepšenie čo sa týka zmiernenia pretrénovania. Možné zlepšenie by teda mohlo byť dosiahnuté použitím ešte menej komplexného modelu alebo *encoder*-u. Takúto modifikáciu sme ale ďalej neskúmali z časových dôvodov.

Ďalšie testované regularizačné techniky zahŕňali regularizáciu pomocou tzv. *weight decay* parametra popísaných v sekcii 3.1.1 a použitie tzv. *spatial dropout* vrstiev. Umiestenie *dropout* vrstiev v upravenom modeli U-Net s *encoder*-om DenseNet 121 je taktiež popísané v časti 3.4.1.1.

Použitie *weight decay* bolo testované s tromi hodnotami (1×10^{-2} , 1×10^{-4} , 1×10^{-6}). V prípade hodnôt 1×10^{-2} a 1×10^{-4} sa model už po pár epochách zastavil na konštantných hodnotách *mIoU* a ďalej už nevykazoval učenie. V prípade hodnoty 1×10^{-6} model vykazoval učenie po dobu 35 epoch ale už asi 20. epoche došlo k pretrénovaniu a teda efekt regularizácie nebol úspešný. Metrika *mIoU* po zhruba 30. epoche začala oscilovať v intervale medzi 48-52 % a nepo-

	<i>accuracy</i>	<i>mIoU</i>	<i>F1 score</i>
<i>Dropout v decoder-y</i>	90.47%	54.61%	66.43%
<i>Dropout v encoder-y</i>	90.24%	53.83%	65.54%
<i>Dropout v encoder-y a decoder-y zároveň</i>	90.13%	51.18%	61.77%
žiadna regularizácia	90.76%	55.64%	67.62%

Tabuľka 5.7: Výsledky experimentov modelu U-Net s encoderom DenseNet 121 pri použití priestorových *dropout* vrstiev.

zorovali sme ďalšie zlepšenie. Ďalšie hodnoty parametra *weight decay* neboli skúšané vzhľadom k časovej náročnosti a predpokladaného neúspechu s danou regularizačnou technikou.

Na základe tejto práce [77], kde autori ukázali, že vhodné umiestenie priestorových *dropout* vrstiev môže priniesť zlepšenie celkovej metriky *mIoU* v prípade plne konvolučných neurónových sietí aj pri použití *batch* normalizácie, sme sa rozhodli zahrnúť niekoľko experimentov aj s využitím tejto regularizačnej techniky. *Dropout* vrstvy sme skúšali umiestniť v *encoder-y*, *decoder-y* a v oboch zároveň. Hodnotu parametra p definujúceho pravdepodobnosť *dropout*-u j -tej *feature* mapy v i -tom *batch*-i snímkov sme nastavili aby stúpala resp. klesala o hodnotu 0.1 pričom stúpanie resp. klesanie tohto parametra korešpondovalo so stúpaním resp. klesaním počtu konvolučných jadier v daných úsekoch siete. Výsledky experimentov s *dropout* vrstvami sú zhrnuté v tabuľke 5.7. Použitím *dropout* vrstiev v *encoder-y* a *decoder-y* samostatne sme pozorovali pretrénovania od 30. epochy pričom ďalej nedochádzalo k zlepšovaniu ani metriky *mIoU* a ani hodnoty účelovej funkcie. Podobný efekt sme pozorovali aj s použitím *dropout*-u v *encoder-y* a *decoder-y* zároveň pričom sa ale pretrénovanie dostavilo až okolo 80. tréningovej epochy. Vo všetkých troch prípadoch sme pozorovali ustálenie metriky *mIoU* v okolí istej hodnoty bez ďalšieho zlepšovania alebo zhoršovania. Žiaden z troch experimentov ale nepriniesol zlepšenie kvality modelu oproti modelu bez regularizácie.

Použitie váh priradeným jednotlivým triedam pixelov v účelovej funkcii bol posledným z vykonaných pokusov na modely plne konvolučnej neurónovej siete a chceli sme ním overiť možnosť zlepšenia predikcii modelu vzhľadom k málo zastúpeným triedam v tréningovom datasete, ktorý ako už bolo spomenuté vykazuje istú mieru nevyváženosti. Prezorkovacie metódy ako *oversampling* a *undersampling* sme neskúšali pretože pri tréningu modelu neurónovej siete používame celé snímky čo prináša problém vzhľadom k implementácii danej procedúry.

Váhy pixelov sme vypočítali na základe vzťahu navrhnutého v práci [32]. Konkrétne sú váhy dané predpisom:

$$\alpha_c = \frac{\text{median}(f_c)}{f_c} \quad (5.2)$$

kde f_c je podiel celkového počtu pixelov danej triedy c v datasete a celkového počtu pixelov vo všetkých obrázkoch datasetu kde sa pixel triedy c nachádza. Po aplikácii takto vypočítaných váh v účelovej funkcii *cross-entropy* sme pozorovali vzrast metriky *recall* v niektorých menej zastúpených triedach povrchov ako napríklad *Vegetation* a *Artificial non-agricultural vegetated areas*. Na druhú stranu sa ukázalo výrazne zhoršenie detekcie triedy *Open spaces with little or no vegetation*. V konečnom dôsledku metrika *mIoU* resp. *F1 score* zaznamenala mierny pokles z hodnoty 55.64% resp. 67.62% na hodnotu 51.33% resp. 63.59% oproti modelu bez váženej účelovej funkcie.

	<i>Použité všetky spektrálne pásma vrátane indexov NDVI, NDWI a NDBI</i>	<i>Použité len spektrálne pásma bez indexov</i>	<i>Použité len RGB pásma (B04, B03, B02)</i>
<i>accuracy</i>	81.77%	81.61%	73.42%
<i>mIoU</i>	33.39%	33.16%	22.30%
<i>F1 score</i>	40.93%	40.72%	28.10%

Tabuľka 5.8: Porovnanie algoritmu XGBoost vzhľadom k použitým spektrálnym pásmam. V prípade nepoužitia spektrálnych indexov nedošlo k výraznej zmene v klasifikačnej schopnosti algoritmu z čoho plynie ich nadbytočnosť. Použitím len RGB pásiem došlo k výraznému zhoršeniu schopnosti klasifikácie čo sa dalo očakávať keďže algoritmus má k dispozícii len spektrálnu informáciu na rozdiel od už popísanej neurónovej siete.

5.2.2.2 Algoritmus XGBoost

V prípade algoritmu XGBoost sa ukázala časová náročnosť učenia pri použití surových pixelov ako vzoriek na učenie pozbavených priestorovej informácie a nesúcich teda len čisto spektrálnu informáciu. Okrem toho algoritmus dosahuje najlepšie výsledky pri prístupe k celému datasetu naraz a neumožňuje tak efektívne učenie v iteráciách po menších *batch*-och obrázkov. Táto limitácia bola vyriešená náhodným výberom obrázkov z tréningového datasetu. Takto sme vyskúšali použitie 512, 1024, 4096 a 16384 náhodne vybraných obrázkov z ktorých boli následne extrahované pixely. V prípade 16384 vybraných obrázkov sme získali zhruba 2.68×10^8 pixelov ako tréningových vzoriek a v prípade 512 obrázkov to bolo 8.39×10^6 pixelov na tréningovanie. Využitie väčšieho množstva obrázkov ale neprinieslo výrazné zlepšenie oproti použitiu len 512 resp. 1024 obrázkov a preto sme sa rozhodli tréňovať vždy na 1024 obrázkoch. Toto nám okrem iného umožnilo využiť GPU na zrýchlenie učenia. Použitie väčšieho množstva obrázkov ako 16384 sme ďalej netestovali kvôli dosiahnutiu limitácie pamäte RAM.

Hodnoty hyperparametrov algoritmu XGBoost sme použili rovnaké ako pri klasifikačnej úlohe kvôli časovej náročnosti ladenia týchto hyperparametrov. Ich hodnoty sú uvedené v prílohe 5.2.2.3.

Na obrázky použité na učenie sme skúšali aplikovať štandardizáciu ako v prípade neurónovej siete ale táto nepriniesla žiadne zlepšenie a preto sme ju ďalej nepoužívali.

Výsledok v základnom nastavení algoritmu XGBoost pri tréningu na 1024 náhodne vybraných obrázkoch dosiahol hodnotu metriky *mIoU* **33.39%**. Evaluácia tohto modelu na testovacom datasete je kompaktné zaznamenaná v konfúznej matici na obrázku 5.7. Ako môžeme vidieť tak mal algoritmus problém s klasifikáciou menej zastúpených tried povrchov. Okrem toho algoritmus vykazoval opäť pretréningovanie, ktoré sa nám nepodarilo zredukovať ani v prípade použitia väčšieho množstva pixelov na tréningovanie.

Porovnanie vzhľadom k použitým spektrálnym pásmam sme testovali aj v prípade algoritmu XGBoost. Výsledky experimentov sú zhrnuté v tabuľke 5.8. Podobne ako v prípade neurónovej siete sa ukazuje malá významnosť použitia dodatočných spektrálnych indexov.

Na druhú stranu experiment s využitím len RGB pásiem ukázal výraznejšie zhoršenie klasifikačných výsledkov v porovnaní s použitím všetkých spektrálnych pásiem. Metrika *mIoU* sa líši až o 11 %.

Nevyvážený dataset vplýval aj na kvalitu klasifikácie modelu XGBoost, ktorý vykazoval zhoršenú schopnosť klasifikácie tried s nízkym zastúpením. Rozhodli sme sa preto vyskúšať prístup

	<i>Prevzorkovanie trénovacieho datasetu (6×10^5 pixelov na triedu)</i>	<i>Priradenie váh triedam proporčne k ich zastúpeniu</i>	<i>Použitie nevyváženého rozloženia tried</i>
<i>accuracy</i>	61.38%	72.76%	81.77%
<i>mIoU</i>	28.47%	31.49%	33.39%
<i>F1 score</i>	37.44%	40.16%	40.93%

Tabuľka 5.9: Porovnanie algoritmu XGBoost vzhľadom k použitiu váh a prevzorkovania datasetu. Žiadna z týchto metód nepriniesla zlepšenie oproti použitiu nevyváženého zastúpenia tried povrchov na tréovanie.

s priradením váh jednotlivým pixelom ale aj prevzorkovanie množiny pixelov na 6×10^5 vzoriek na triedu vybraných z obrázkov z celého trénovacieho datasetu. Váhy boli pixelom priradené na základe vzťahu 5.1 použitému aj v prípade klasifikačnej úlohy. Ako sme očakávali na základe výsledkov s vážením pri modeli neurónovej siete tak toto neprinieslo zlepšenie celkovej metriky *mIoU* ale mierne zhoršenie na hodnotu **31.49%**. Zjavne sa zlepšila schopnosť detekovať málo zastúpené triedy ale zhoršila schopnosť detekovať tie vysoko zastúpené. Zväčšením počtu *boost*-ovacích iterácií z 50 na 250 sa výsledok viditeľne nezmenil.

V prípade prevzorkovania sme očakávali zlepšenie ale toto sa nepreukázalo, práve naopak *mIoU* dosiahlo hodnotu len **28.47%** opäť sprevádzané zlepšením klasifikácie väčšiny málo zastúpených tried povrchov ale zhoršením tých so zastúpením vysokým. Napríklad tento klasifikátor vo vysokej miere zamieňal triedu *Forest* s triedou *Inland wet lands* čo sa v prípade nevyváženého datasetu nedialo v tak vysokej miere. Konfúzna matica zachytávajúca výsledok segmentácie po prevzorkovaní datasetu je na obrázku 5.8. V tabuľke 5.9 sú zhrnuté výsledky experimentov s vážením a prevzorkovaním.

5.2.2.3 Porovnanie algoritmov

Získané výsledky jednoznačne poukazujú na dominanciu modelu plne konvolučnej neurónovej siete U-Net oproti prístupu s takpovediac surovými pixelmi v prípade algoritmu XGBoost a to s ohľadom na všetky sledované metriky. Konkrétne dosiahol model U-Net v najlepšom nastavení hodnotu metriky *mIoU* až **55.64%** na testovacom datasete, kdežto model XGBoost dosiahol hodnotu tejto metriky v najlepšom nastavení iba **33.39%**. V oboch prípadoch modely vykazovali zhoršenie schopnosti klasifikácie menej zastúpených tried ako *Vegetation*, *Inland wet lands* a ďalších. Taktiež v oboch prípadoch dosahovala trieda *Water* jedny z najvyšších hodnôt metriky *mIoU* aj napriek jej relatívne nízkemu zastúpeniu v datasete. Model XGBoost preukázal značné problémy pri klasifikácii tried povrchov vykazujúcich podobnosť v spektrálnych signatúrach. Použitie váh na jednotlivé triedy a ani prevzorkovanie datasetu na vhodný počet pixelov na triedu neprinieslo zlepšenie no práve naopak. Zlepšenie modelu založeného na algoritme XGBoost vidíme v možnom rozšírení množiny prediktorov zahŕňajúc aj prediktory nesúce priestorovú informáciu obsiahnutú v obrázkoch. Nakoľko nás ale využitie neurónovej siete v podstate úplne zbavuje nutnosti návrhu vhodných prediktorov tak sa použitie algoritmu XGBoost zdá ako zbytočné.

Dané výsledky taktiež preukázali značný význam použitia priestorovej informácie pri klasifikácii ako bolo učinené v prípade modelu neurónovej siete. Použitie len čisto RGB pásiem pri tréovaní modelu neurónovej siete sa ukázalo ako úspešné. Ďalšie zlepšenie tohto modelu by mohlo byť dosiahnuté použitím *transfer learning*-u.

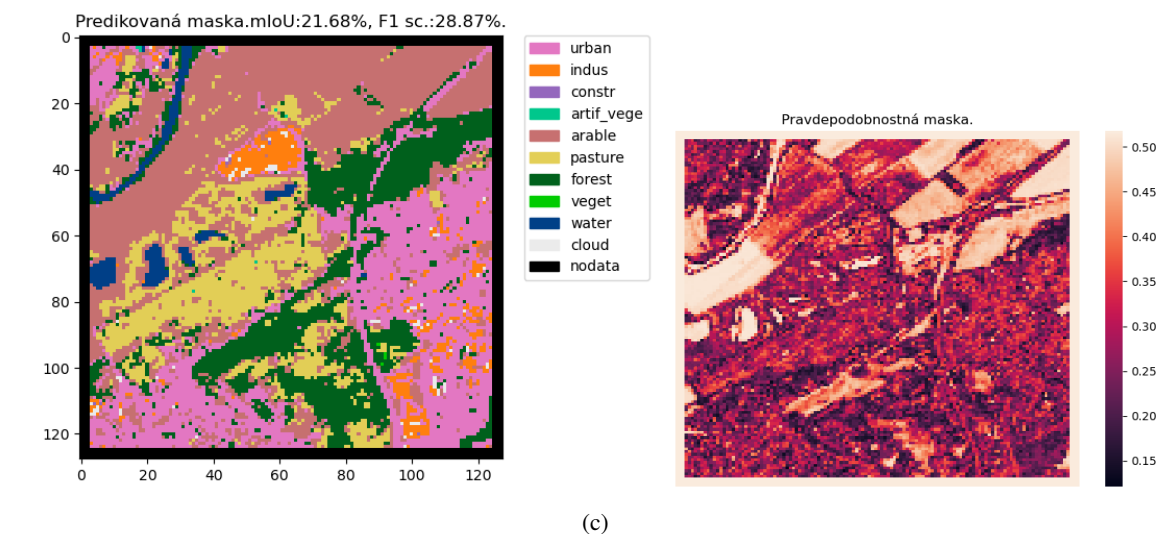
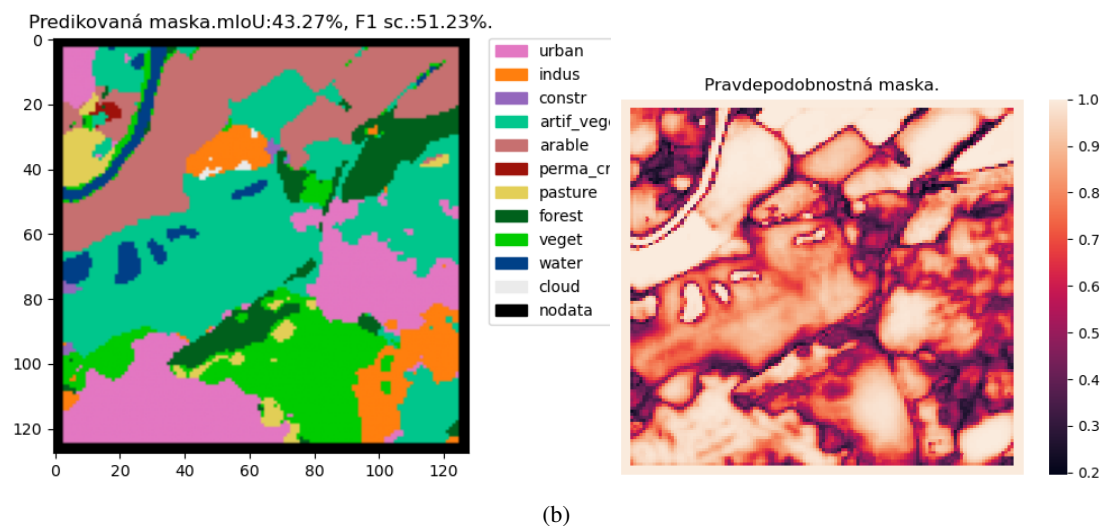
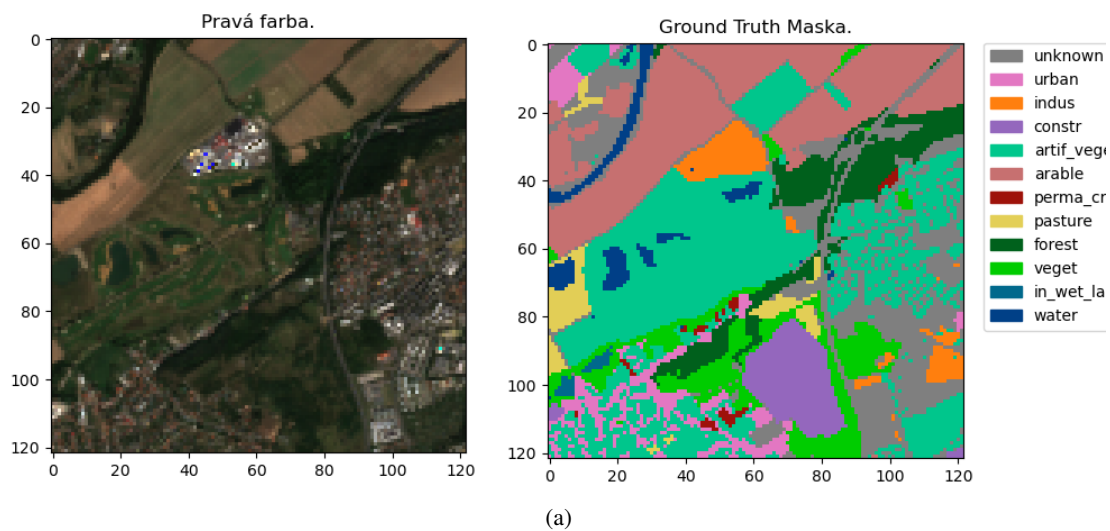
Na každom z obrázkov 5.9, 5.10, 5.11, 5.12, 5.13 môžeme vidieť v (a) postupne snímku v pravých farbách (naľavo) spolu s *ground truth* maskou (napravo). Ďalej v (b) resp. (c) predikciu (naľavo) a odhad istoty predikcie (napravo) segmentácie najlepších z modelov U-Net resp. XG-Boost. Vo všetkých prezentovaných prípadoch dosiahla neurónová sieť lepšie výsledky vzhľadom k metrikám *mIoU* a *F1 score*. Okrem toho model U-Net predikoval segmentačné masky s väčšou istotou, pričom väčšiu mieru neistoty pri predikcii vykazoval na rozhraniach medzi povrchni. Model XGBoost generoval predikcie s menšou istotou a v jeho predpovediach pozorujeme výraznú nehomogenitu a zrnitosť čo ale plynie z implementácie a použitia priamo pixelov ako vzoriek.

	unknown	urban	indus	constr	artif_vege	arable	perma_crop	pasture	forest	veget	no_veget	in_wet_land	water	cloud	Suma(skutoč)	Recall	IoU	
unknown	4.67e+05 1.17%	3.79e+04 0.09%	6.04e+03 0.02%	3.48e+02 0.00%	2.24e+06 5.61%	2.76e+02 0.00%	8.14e+05 2.04%	1.09e+06 2.74%	3.17e+03 0.01%	4.30e+02 0.00%	1.79e+03 0.00%	2.25e+04 0.06%	8.74e+04 0.22%		4.78e+06 11.95%	0.00%	0.00%	
urban	8.98e+05 2.25%	3.61e+04 0.09%	6.26e+02 0.00%	5.30e+01 0.00%	3.46e+05 0.87%	2.38e+02 0.00%	6.18e+04 0.15%	2.46e+05 0.62%	2.42e+02 0.00%	1.19e+02 0.00%	6.40e+01 0.00%	2.64e+03 0.01%	2.47e+04 0.06%		1.62e+06 4.05%	55.54%	36.61%	
indus	8.53e+04 0.21%	7.11e+04 0.18%	1.26e+03 0.00%	6.00e+00 0.00%	4.83e+04 0.12%		4.38e+03 0.01%	1.79e+04 0.04%	5.90e+01 0.00%	2.86e+02 0.00%	9.00e+01 0.00%	8.82e+02 0.00%	1.32e+04 0.03%		2.43e+05 0.61%	29.29%	22.14%	
constr	1.12e+04 0.03%	1.03e+04 0.03%	5.52e+03 0.01%	2.00e+00 0.00%	3.44e+04 0.09%		4.31e+03 0.01%	1.36e+04 0.03%	3.00e+01 0.00%	5.02e+02 0.00%	9.00e+00 0.00%	1.58e+03 0.00%	3.39e+03 0.01%		8.48e+04 0.21%	6.51%	6.02%	
artif_vege	6.34e+04 0.16%	1.27e+03 0.00%	6.80e+01 0.00%	3.93e+02 0.00%	3.32e+04 0.08%	8.00e+00 0.00%	1.78e+04 0.04%	5.95e+04 0.15%	1.79e+02 0.00%	1.90e+01 0.00%	5.00e+00 0.00%	7.87e+02 0.00%	2.98e+03 0.01%		1.80e+05 0.45%	0.22%	0.22%	
arable	2.16e+05 0.54%	1.23e+04 0.03%	2.53e+03 0.01%	8.80e+01 0.00%	8.27e+06 20.65%	8.12e+02 0.00%	5.45e+05 1.36%	3.91e+05 0.98%	8.22e+02 0.00%	2.51e+02 0.00%	4.93e+02 0.00%	1.64e+04 0.04%	4.19e+04 0.10%		9.50e+06 23.76%	87.07%	70.05%	
perma_crop	3.84e+04 0.10%	4.00e+01 0.00%	1.10e+01 0.00%		8.50e+04 0.21%	5.40e+01 0.00%	2.23e+04 0.06%	3.50e+04 0.09%	5.40e+01 0.00%	2.00e+00 0.00%	5.00e+00 0.00%	3.49e+02 0.00%	1.58e+03 0.00%		1.83e+05 0.46%	0.03%	0.03%	
pasture	5.79e+04 0.14%	6.27e+02 0.00%	1.07e+02 0.00%	3.35e+02 0.00%	4.11e+05 1.03%	4.75e+02 0.00%	1.53e+06 3.62%	3.58e+05 0.89%	2.25e+03 0.01%	4.60e+01 0.00%	3.01e+02 0.00%	2.13e+03 0.01%	1.40e+04 0.04%		2.37e+06 5.94%	64.31%	44.28%	
forest	1.39e+05 0.35%	7.18e+02 0.00%	3.67e+02 0.00%	1.03e+02 0.00%	3.03e+05 0.76%	5.50e+01 0.00%	2.23e+05 0.56%	1.26e+07 31.41%	2.57e+03 0.01%	3.70e+01 0.00%	6.74e+02 0.00%	8.18e+03 0.02%	5.58e+04 0.14%		1.33e+07 33.24%	94.48%	82.79%	
veget	4.64e+04 0.12%	1.76e+03 0.00%	1.05e+02 0.00%	2.27e+02 0.00%	1.21e+05 0.30%	2.70e+01 0.00%	1.19e+05 0.30%	2.26e+05 0.57%	3.72e+03 0.01%	2.10e+01 0.00%	6.60e+01 0.00%	1.04e+03 0.00%	3.46e+03 0.01%		5.23e+05 1.31%	0.71%	0.70%	
no_veget	1.82e+03 0.00%	2.13e+02 0.00%	1.07e+03 0.00%	1.00e+00 0.00%	6.38e+03 0.02%		1.10e+03 0.00%	3.56e+03 0.01%	2.00e+01 0.00%	1.21e+02 0.00%	3.00e+00 0.00%	2.17e+02 0.00%	6.68e+02 0.00%		1.52e+04 0.04%	0.80%	0.73%	
in_wet_land	6.35e+02 0.00%	2.00e+00 0.00%		4.00e+01 0.00%	7.39e+03 0.02%		1.86e+04 0.05%	4.91e+04 0.12%	7.00e+01 0.00%		2.40e+01 0.00%	8.59e+02 0.00%	2.50e+02 0.00%		7.70e+04 0.19%	0.03%	0.03%	
water	6.48e+03 0.02%	1.30e+03 0.00%	3.40e+02 0.00%	6.00e+00 0.00%	2.26e+04 0.06%	4.00e+00 0.00%	4.38e+03 0.01%	5.93e+04 0.15%	3.20e+01 0.00%	2.00e+00 0.00%	1.30e+02 0.00%	3.13e+05 0.78%	8.93e+02 0.00%		4.08e+05 1.02%	76.60%	66.94%	
cloud	1.70e+05 0.43%	1.37e+04 0.03%	4.91e+02 0.00%	3.70e+01 0.00%	8.90e+05 2.23%	8.00e+00 0.00%	5.19e+04 0.13%	4.18e+05 1.05%	5.31e+02 0.00%	6.20e+01 0.00%	6.40e+01 0.00%	2.39e+04 0.06%	1.01e+06 2.53%		2.58e+06 6.46%	39.23%	36.90%	
nodata															4.12e+06 10.31%	4.12e+06 10.31%	100%	
Suma(pred)	0.00e+00 0.00%	2.20e+06 5.51%	1.87e+05 0.47%	1.85e+04 0.05%	1.64e+03 0.00%	1.28e+07 32.08%	1.96e+03 0.00%	3.41e+01 0.00%	1.55e+07 38.84%	1.38e+04 0.03%	1.90e+03 0.00%	3.72e+03 0.01%	3.94e+05 0.99%	1.26e+06 3.16%	4.12e+06 10.31%	4.00e+07 100%		
Precision	0.00%	51.78%	47.56%	44.17%	30.44%	78.18%	3.21%	58.71%	86.99%	35.19%	8.24%	1.24%	84.15%	86.15%	100%		81.77%	
F1 score	0.00%	53.99%	36.25%	11.35%	0.43%	82.18%	0.06%	61.38%	90.58%	1.40%	1.45%	0.06%	80.20%	53.91%	100%		33.39%	
	unknown	urban	indus	constr	artif_vege	arable	perma_crop	pasture	forest	veget	no_veget	in_wet_land	water	cloud	nodata	Celkovo	Accuracy	mIoU

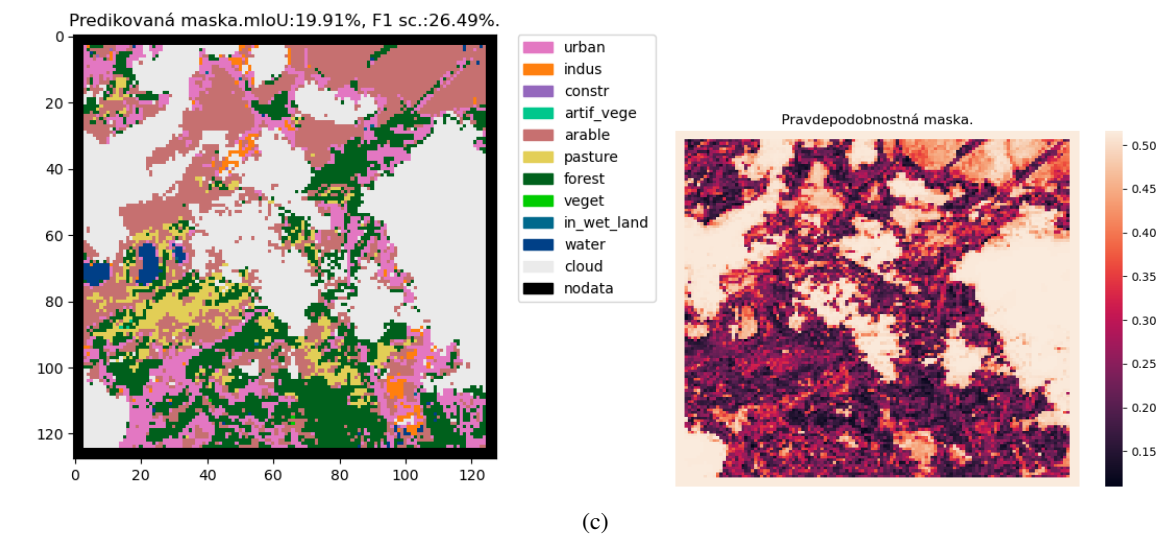
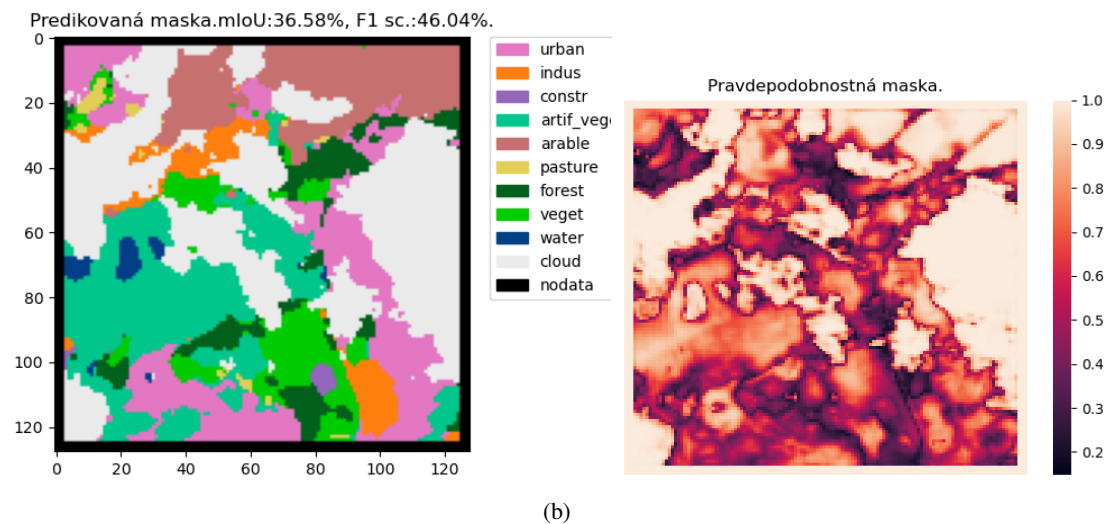
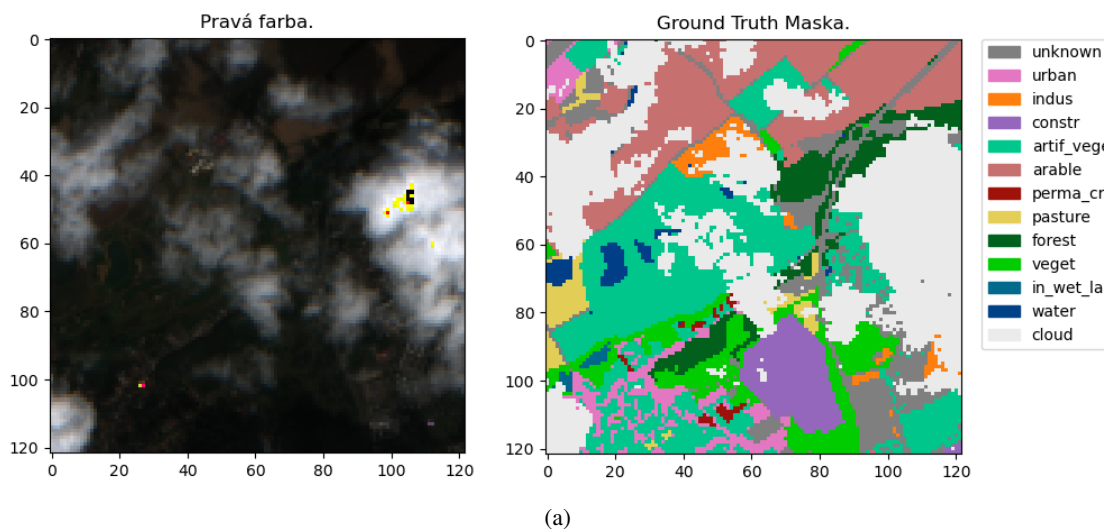
Obr. 5.7: Konfúzna matica zachytávajúca sémantickú segmentáciu pri použití algoritmu XGBoost na 1024 obrázkoch s použitím 50 boostovacích iterácií. Vidíme výrazný problém s klasifikáciou menej zastúpených tried povrchov čo dokladujú hodnoty metrick *recall* a *mIoU*. Výsledná hodnota metriky *mIoU* bola 33.39%.

	unknown	urban	indus	constr	artif_vege	arable	perma_crop	pasture	forest	veget	no_veget	in_wet_land	water	cloud	nodata	Suma(skutoč)	Recall	IoU	
unknown	395e+05 0.99%	210e+05 0.53%	197e+05 0.49%	327e+05 0.82%	101e+06 2.52%	481e+05 1.20%	701e+05 1.75%	263e+05 0.66%	225e+05 0.56%	165e+05 0.41%	445e+05 1.11%	901e+04 0.23%	268e+05 0.67%		4.78e+06 11.95%	0.00%	0.00%	unknown	
urban	670e+05 1.68%	214e+05 0.54%	197e+04 0.05%	234e+05 0.59%	291e+04 0.07%	193e+05 0.48%	510e+04 0.13%	292e+04 0.07%	377e+04 0.09%	403e+04 0.10%	362e+04 0.09%	127e+04 0.03%	507e+04 0.13%		1.62e+06 4.05%	41.39%	26.45%	urban	
indus	344e+04 0.09%	137e+05 0.34%	118e+04 0.03%	151e+04 0.04%	499e+03 0.01%	110e+04 0.03%	371e+03 0.01%	215e+03 0.01%	405e+03 0.01%	606e+03 0.02%	288e+03 0.01%	173e+03 0.00%	802e+03 0.02%		2.43e+05 0.61%	56.34%	16.14%	indus	
constr	662e+03 0.02%	179e+04 0.04%	227e+04 0.06%	450e+03 0.01%	373e+03 0.01%	602e+03 0.02%	274e+03 0.01%	434e+03 0.01%	370e+03 0.01%	437e+03 0.01%	326e+03 0.01%	228e+03 0.01%	260e+03 0.01%		8.48e+04 0.21%	26.77%	2.54%	constr	
artif_vege	350e+04 0.09%	730e+03 0.02%	165e+03 0.00%	531e+04 0.13%	230e+03 0.01%	239e+04 0.06%	110e+04 0.03%	112e+04 0.03%	731e+03 0.02%	535e+03 0.01%	139e+04 0.03%	210e+03 0.01%	567e+03 0.01%		1.80e+05 0.45%	29.54%	3.64%	artif_vege	
arable	440e+05 1.10%	202e+05 0.50%	570e+05 1.43%	191e+05 0.48%	542e+06 13.50%	927e+05 2.32%	659e+05 1.65%	844e+04 0.21%	226e+05 0.57%	184e+05 0.46%	246e+05 0.62%	786e+04 0.20%	269e+05 0.67%		9.50e+06 23.76%	57.06%	54.16%	arable	
perma_crop	212e+04 0.05%	134e+03 0.00%	340e+03 0.01%	135e+04 0.03%	103e+04 0.03%	877e+04 0.22%	153e+04 0.04%	213e+03 0.01%	842e+03 0.02%	386e+03 0.01%	904e+03 0.02%	182e+03 0.00%	376e+03 0.01%		1.83e+05 0.46%	48.51%	4.47%	perma_crop	
pasture	509e+04 0.13%	987e+03 0.02%	324e+04 0.08%	103e+05 0.26%	890e+04 0.22%	191e+05 0.48%	136e+06 3.39%	634e+04 0.16%	111e+05 0.28%	454e+04 0.11%	265e+05 0.66%	119e+04 0.03%	445e+04 0.11%		2.37e+06 5.94%	57.15%	39.39%	pasture	
forest	169e+05 0.42%	356e+04 0.09%	546e+04 0.14%	523e+05 1.31%	765e+04 0.19%	290e+05 0.73%	183e+05 0.46%	784e+06 19.61%	486e+05 1.22%	203e+05 0.51%	283e+06 7.08%	180e+05 0.45%	416e+05 1.04%		1.33e+07 33.24%	58.99%	57.54%	forest	
veget	363e+04 0.09%	123e+04 0.03%	144e+04 0.04%	446e+04 0.11%	192e+04 0.05%	581e+04 0.15%	833e+04 0.21%	455e+04 0.11%	653e+04 0.16%	164e+04 0.04%	110e+05 0.28%	452e+03 0.01%	126e+04 0.03%		5.23e+05 1.31%	12.49%	4.51%	veget	
no_veget	958e+02 0.00%	953e+02 0.00%	187e+03 0.00%	130e+03 0.00%	446e+02 0.00%	990e+02 0.00%	535e+02 0.00%	734e+02 0.00%	396e+02 0.00%	610e+03 0.01%	899e+02 0.00%	438e+02 0.00%	544e+02 0.00%		1.52e+04 0.04%	33.64%	0.82%	no_veget	
in_wet_land	811e+02 0.00%	163e+02 0.00%	235e+02 0.00%	325e+03 0.01%	165e+03 0.00%	312e+03 0.02%	974e+03 0.02%	181e+04 0.05%	359e+03 0.01%	139e+03 0.00%	318e+04 0.08%	193e+03 0.00%	111e+03 0.00%		7.70e+04 0.19%	41.35%	0.87%	in_wet_land	
water	430e+03 0.01%	586e+03 0.01%	421e+03 0.01%	634e+03 0.02%	478e+03 0.01%	525e+03 0.01%	284e+03 0.01%	188e+04 0.05%	227e+03 0.01%	357e+03 0.01%	125e+04 0.03%	335e+05 0.84%	269e+03 0.01%		4.08e+05 1.02%	82.02%	45.09%	water	
cloud	114e+05 0.28%	977e+04 0.24%	932e+04 0.23%	139e+05 0.35%	265e+05 0.66%	921e+04 0.23%	475e+04 0.12%	546e+04 0.14%	330e+04 0.08%	935e+04 0.23%	560e+04 0.14%	358e+04 0.09%	146e+06 3.65%		2.58e+06 6.46%	56.53%	42.94%	cloud	
nodata															4.12e+06 10.31%	4.12e+06 10.31%	100%	100%	nodata
Suma(pred)	0.00e+00 0.00%	1.98e+06 4.95%	9.52e+05 2.38%	1.03e+06 2.57%	1.66e+06 4.15%	6.93e+06 17.35%	2.37e+06 5.94%	3.13e+06 7.83%	8.43e+06 21.11%	1.21e+06 3.04%	7.77e+05 1.94%	4.06e+06 10.17%	7.59e+05 1.90%	2.54e+06 6.37%	4.12e+06 10.31%	4.00e+07 100%			Celkovo
Precision	0.00%	42.29%	18.45%	2.73%	3.88%	91.44%	4.69%	55.90%	95.91%	6.60%	0.83%	0.88%	50.04%	64.11%	100%		61.38%		Accuracy
F1 score	0.00%	41.84%	27.79%	4.36%	7.02%	70.27%	8.55%	56.51%	73.05%	8.64%	1.63%	1.72%	60.08%	60.08%	100%			28.47%	mIoU

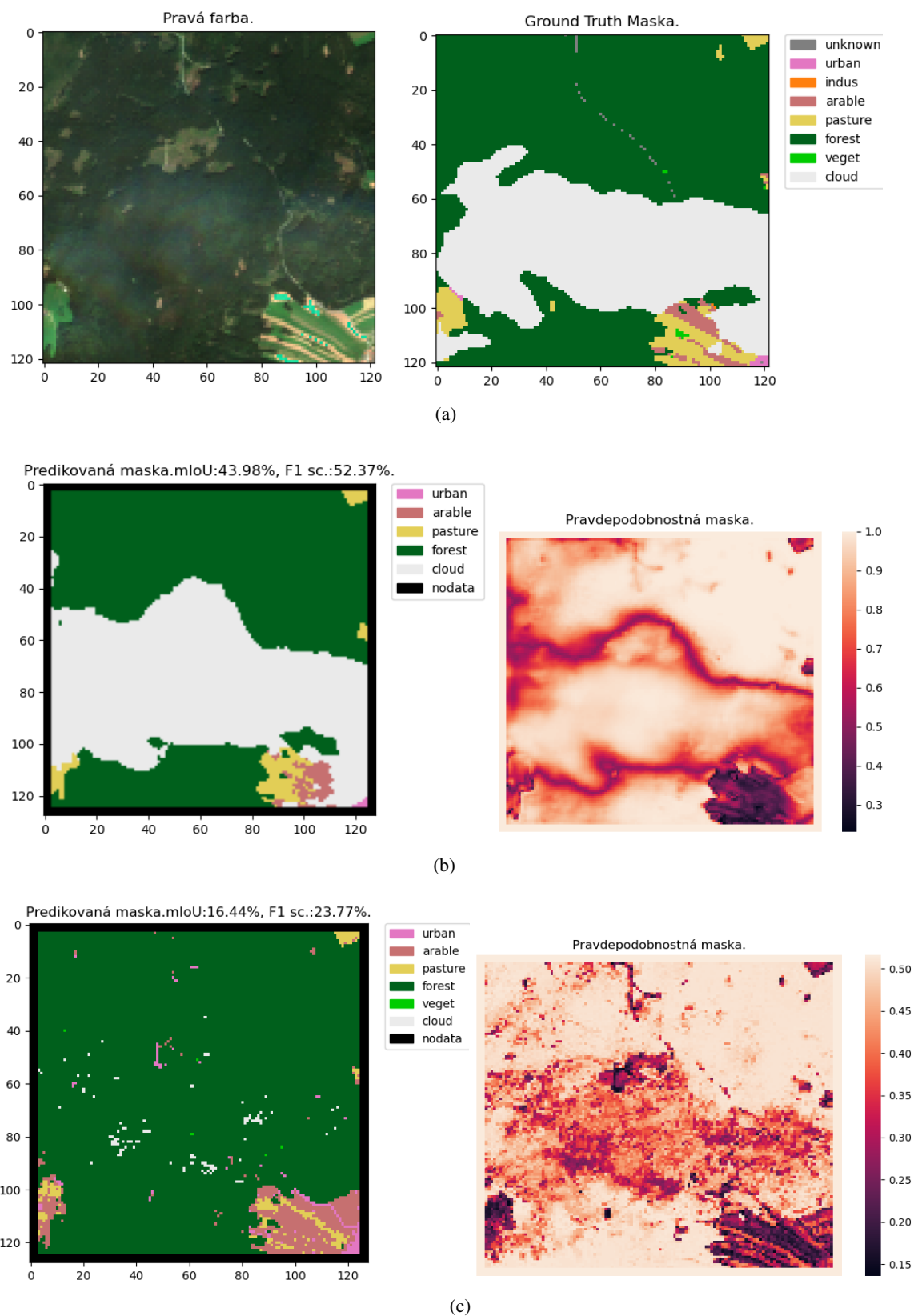
Obr. 5.8: Konfúzna matica sémantickej segmentácie algoritmom XGBoost po prevzorkovaní pixelov na 6×10^5 vzoriek na triedu. Daný model si zlepšil schopnosť klasifikovať triedy s prírodným nízkym zastúpením ale mierne zhoršil schopnosť klasifikovať triedy s vysokým zastúpením čo vo výsledku viedlo k zhoršeniu celkovej klasifikačnej schopnosti.



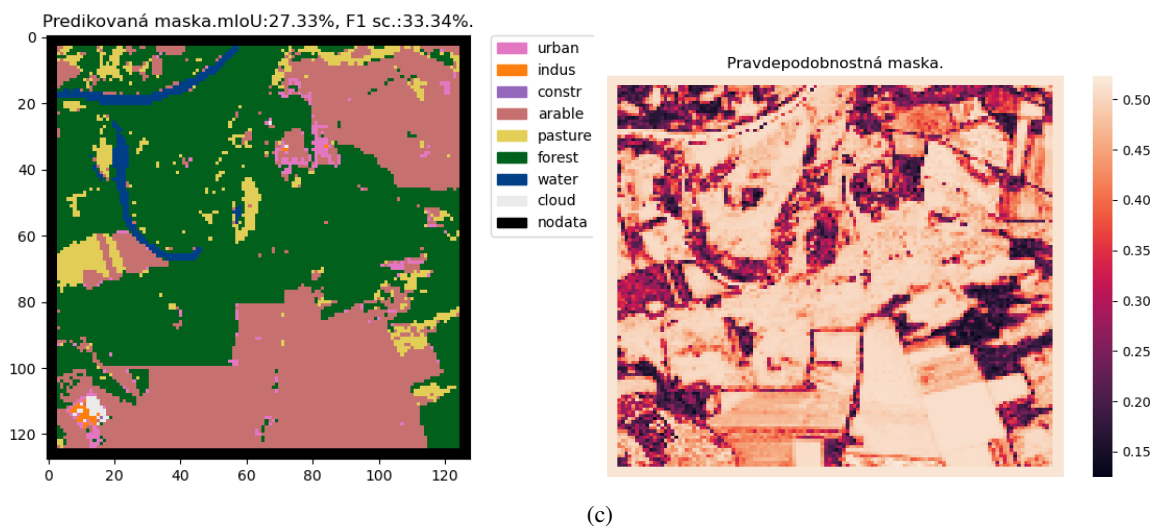
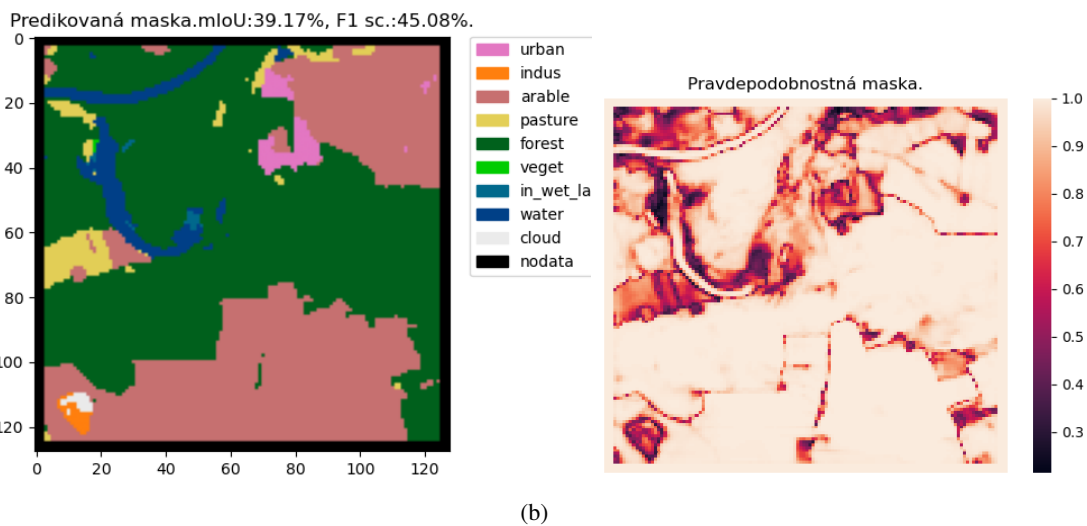
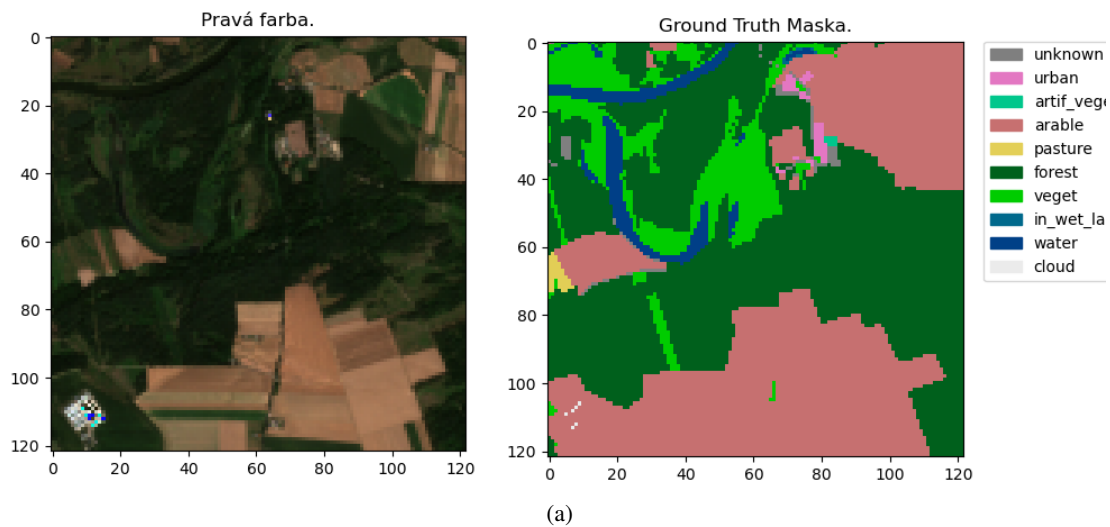
Obr. 5.9: V tomto prípade si oba modely poradili relatívne dobre. Oblasť s jazierkami označená v *ground truth* ako *Artificial vegetation* reprezentuje golfové ihrisko. Ako vidíme tak model U-Net dokázal identifikovať takýto typ povrchu, kdežto model XGBoost nie a priklonil sa k triede *Pastures* s podobnou spektrálnou signatúrou a zároveň väčším zastúpením v datase. Toto poukazuje na potrebu využitia priestorovej informácie medzi pixelami na odlíšenie dvoch povrchov s podstate rovnakou spektrálnou charakteristikou. (*True color a ground truth* (a), U-Net (b), XGBoost (c))



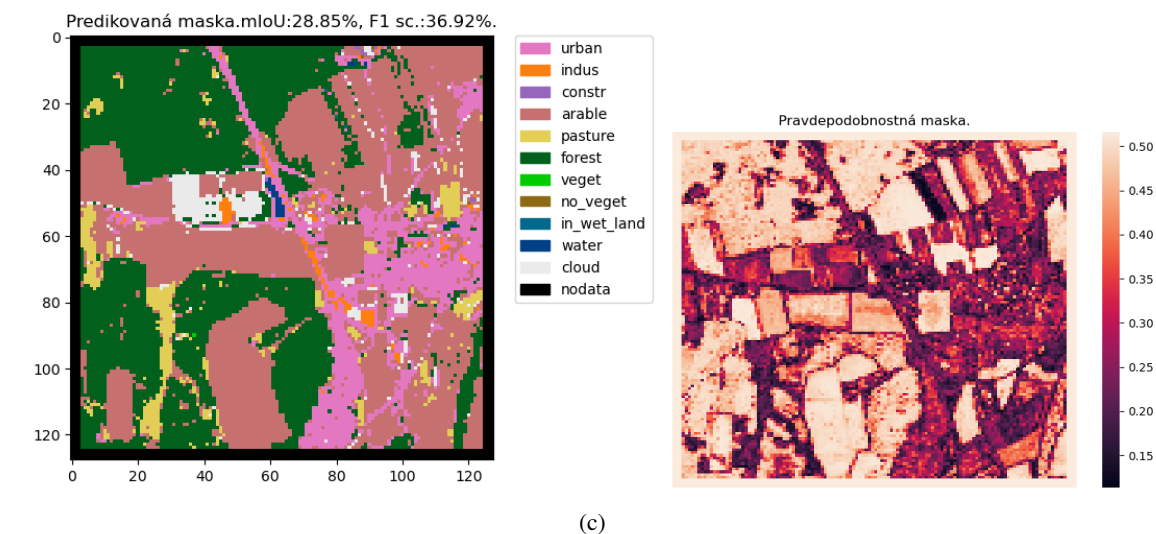
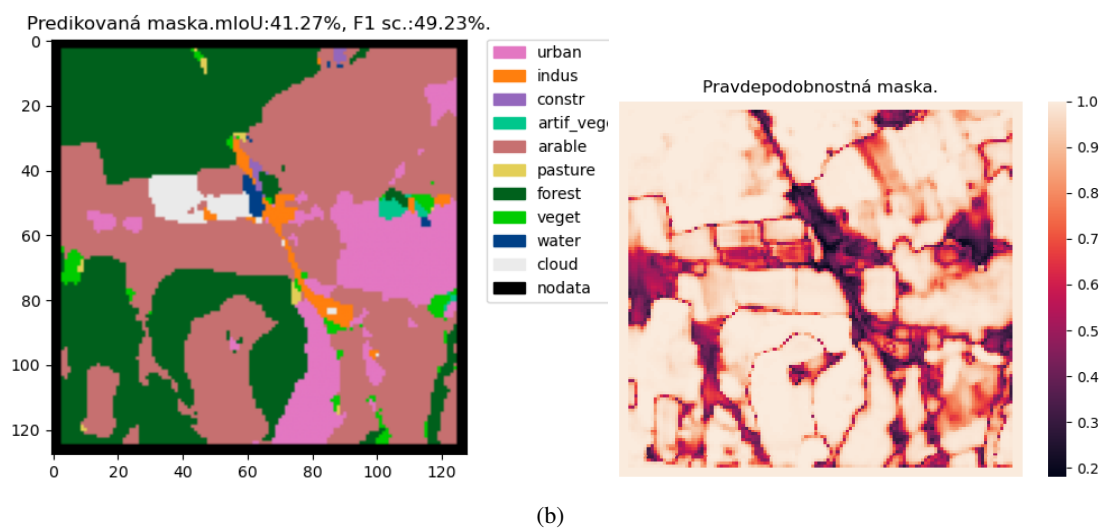
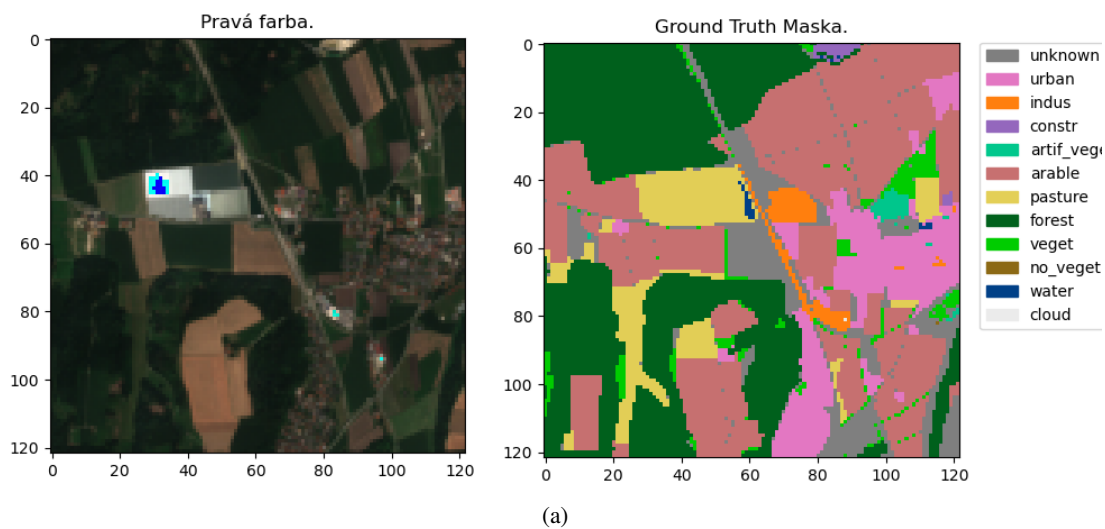
Obr. 5.10: Na tomto obrázku pozorujeme rovnaké územie ako v obrázku 5.9 ale zaznamenané v inom čase. Na obrázku je patrná oblačnosť a taktiež tieň spôsobený oblačnosťou. Oba modely si ale poradili s detekciou oblačnosti relatívne dobre. (*True color* a *ground truth* (a), U-Net (b), XGBoost (c))



Obr. 5.11: Na danom obrázku je väčšina územia pokrytá lesom pričom na zábere môžeme pozorovať výskyt tenkého cirrusu. Takto malá zmena v jase spôsobená jeho prítomnosťou nebola efektívne detekovaná modelom XGBoost kdežto model U-Net si s ňou poradil. (*True color a ground truth (a)*, U-Net *(b)*, XGBoost *(c)*)



Obr. 5.12: V tomto prípade vidíme, že oba modely mali problém klasifikácie vegetácie v okolí brehu rieky a preferovali radšej viac zastúpenú triedu *Forest*. (*True color* a *ground truth* (a), U-Net (b), XGBoost (c))



Obr. 5.13: Môžeme si všimnúť, že oblasť v *ground truth* označená ako *Pastures* (žltou) vykazuje príliš vysokú reflexivitu čo spôsobilo, že oba modely klasifikovali časť daného povrchu ako triedu *Clouds*. (*True color a ground truth* (a), U-Net (b), XGBoost (c))

Záver

V našej práci sme ukázali, že multispektrálne snímky z misie Sentinel-2 spolu s mapovými dátami OSM tvoria vhodný zdroj dát na automatizovanú tvorbu datasetov krajinného pokrytia. Vytvorené datasety boli následne použité na riešenie klasifikácie objektov a sémantickú segmentáciu pokrytia krajiny za pomoci metód strojového učenia. Algoritmy dosiahli relatívne vysoké hodnoty metrík vzhľadom k nevyváženosti datasetov ale aj podobnostiam v spektrálnych charakteristikách jednotlivých tried povrchov. Metódy aplikované na zmiernenie vplyvu nevyváženosti datasetov, menovite použitie váženej účelovej funkcie ale aj prevzorkovanie tréningového datasetu sa ukázali ako neefektívne ako v klasifikácii tak aj v sémantickej segmentácii. Taktiež v oboch prípadoch modeli vykazovali pretrénovanie, ktoré sa nám žiadnou skúšanou formou regularizácie nepodarilo efektívne odstrániť.

Čo sa týka klasifikačnej úlohy tak v nej sme dosiahli najlepšie výsledky s použitím algoritmu XGBoost a to až 76.16% čo sa týka hodnoty metriky *accuracy*. Zaujímavým výsledkom pri klasifikácii objektov bolo to, že najkvalitnejší model bol získaný pri využití časovej série snímok ako ďalších spektrálnych prediktorov na tréning pri čom dosiahol model XGBoost hodnotu *accuracy* až 79.96%. Takýto prístup s využitím rozsiahlejšej časovej rady snímok ako dodatočných spektrálnych prediktorov by mohol teda byť námetom ďalších vylepšení a výskumu. Použitie spektrálnych indexov ako dodatočných prediktorov sa ukázalo ako málo významným. Ďalšími vylepšeniami by mohlo byť použitie prediktorov založených na iných ako spektrálnych vlastnostiach obrázkov.

V prípade sémantickej segmentácie dosiahol model využívajúci neurónovú sieť v architektúre U-Net s *encoder*-om DenseNet 121 omnoho lepšie klasifikačné výsledky v porovnaní s použitím modelu XGBoost. Tento rozdiel bol ale očakávaný a poukazuje na význam využitia priestorovej informácie vyplývajúcej z uloženia pixelov v snímkach. Konkrétne najlepší model neurónovej siete dosiahol hodnotu metriky *mIoU* 55.64%, kdežto model XGBoost iba 33.39%. Použitie spektrálnych indexov ako dodatočných pásiem sa ukázalo byť aj v prípade sémantickej segmentácie ako nadbytočné, neprinášajúc významnú zmenu v hodnotách sledovaných metrík. Ďalším povšimnutia hodným výsledkom bolo to, že využitie len RGB pásiem na sémantickú segmentáciu nepriineslo výrazné zhoršenie kvality modelu U-Net. Na základe tohto výsledku sa dá predpokladať, že zlepšenie by mohlo priniesť použitie *transfer learning*-u pomocou predtrénovaného modelu napríklad na datasete ImageNet s použitím len RGB pásiem a teda bez nutnosti upravovať samotný model na využitie multispektrálnych kanálov. Každopádne zlepšenie klasifikačnej schopnosti modelu na základe *transfer learning*-u s využitím dodatočných spektrálnych pásiem by bolo vhodné taktiež preskúmať nakoľko je pravdepodobné, že prinesie zlepšenie kvality modelu. Veríme, že dodatočné zlepšenie modelu U-Net by prinieslo taktiež systematické ladenie hyperparametrov čo sme ale z časových dôvodov nestihli vykonať.

Ďalšie vylepšovanie modelov vidíme v použití väčšej oblasti záujmu na vytvorenie datasetu alebo použitím augmentácie snímok. Istá forma augmentácie bola vykonaná využitím časovej sé-

rie snímok vďaka zmene jasových podmienok týchto snímok, čo prinieslo zlepšenie robustnosti našich modelov a preto aj zväčšenie tejto časovej rady snímok je nami navrhovaná zmena vzhľadom k vylepšeniu modelu neurónovej siete. Konkrétne v prípade modelu neurónovej siete by bolo možné vyskúšať aj tzv. rekurentnú konvolučnú sieť na efektívnejšie využitie takejto časovej rady snímok.

Nakoniec podotknime, že vytvorené *ground truth* segmentačné masky ako aj získané predikované segmentačné masky by bolo vhodné verifikovať napríklad pomocou porovania pozíčnej a tematickej presnosti za pomoci dostupného datasetu pokrytia krajiny ako napríklad Corine Land Cover [3] dataset. Takúto verifikáciu sme ale z časových dôvodov nestihli vykonať a preto by mohla byť námetom ďalšej práce.

Literatúra

- [1] About SPOT Series. <https://earth.esa.int/eogateway/missions/spot>. online cit.: 2021-05-08.
- [2] Copernicus Open Access Hub. <https://scihub.copernicus.eu/>. online cit.: 2021-05-01.
- [3] CORINE Land Cover. <https://land.copernicus.eu/pan-european/corine-land-cover>. online cit.: 2021-05-01.
- [4] Databases and data access APIs. https://wiki.openstreetmap.org/wiki/Databases_and_data_access_APIs. online cit.: 2021-05-10.
- [5] EOBrowser. <https://www.sentinel-hub.com/explore/eobrowser/>. online cit.: 2021-05-03.
- [6] Level-1C: Cloud Masks. <https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/level-1c/cloud-masks>. online cit.: 2021-05-06.
- [7] Level-1C Product Formatting. <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-1c/product-formatting>. online cit.: 2021-05-06.
- [8] Level-2A Algorithm Overview. <https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm>. online cit.: 2021-05-08.
- [9] OpenStreetMap: Elements. <https://wiki.openstreetmap.org/wiki/Elements>. online cit.: 2021-05-01.
- [10] OpenStreetMap: Map features. https://wiki.openstreetmap.org/wiki/Map_features. online cit.: 2021-05-08.
- [11] OpenStreetMap: Research. <https://wiki.openstreetmap.org/wiki/Research>. online cit.: 2021-05-01.
- [12] Sen2Cor. <http://step.esa.int/main/snap-supported-plugins/sen2cor/>. online cit.: 2021-05-08.
- [13] Sentinel-2 Spectral Response Functions. https://sentinel.esa.int/documents/247904/685211/S2-SRF_COPE-GSEG-EOPG-TN-15-0007_3.0.xlsx. online cit.: 2021-05-01.

- [14] Sentinel-2 User Handbook. https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook. online cit.: 2021-05-01.
- [15] Sentinel Hub. <https://www.sentinel-hub.com/>. online cit.: 2021-05-03.
- [16] The Landsat Program. <https://landsat.gsfc.nasa.gov/>. online cit.: 2021-05-08.
- [17] The Sentinel-2 Toolbox. <https://sentinel.esa.int/web/sentinel/toolboxes/sentinel-2>. online cit.: 2021-05-08.
- [18] Sentinel Hub's cloud detector for Sentinel-2 imagery. <https://github.com/sentinel-hub/sentinel2-cloud-detector>, 2018. online cit.: 2021-05-08.
- [19] BACH, H., APPEL, F., FELLAH, K., AND DE FRAIPONT, P. Application of flood monitoring from satellite for insurances. In *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.* (2005), vol. 1, pp. 4 pp.–.
- [20] BADRINARAYANAN, V., KENDALL, A., AND CIPOLLA, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.
- [21] BANGIRA, T., ALFIERI, S., MENENTI, M., AND VAN NIEKERK, A. Comparing Thresholding with Machine Learning Classifiers for Mapping Complex Water. *Remote Sensing* 11 (06 2019).
- [22] BENBAHRIA, Z., SMIEJ, M. F., SEBARI, I., AND HAJJI, H. Land cover intelligent mapping using transfer learning and semantic segmentation. In *2019 7th Mediterranean Congress of Telecommunications (CMT)* (2019), pp. 1–5.
- [23] BOEING, G. OSMNX: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks. *Computers Environment and Urban Systems* 65 (07 2017), 126–139.
- [24] CHEN, L.-C., PAPANDREOU, G., KOKKINOS, I., MURPHY, K., AND YUILLE, A. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (06 2016).
- [25] CHEN, T., AND GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, ACM, pp. 785–794.
- [26] CHOLLET, F. Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1800–1807.
- [27] CORTES, C., AND VAPNIK, V. Support Vector Networks. *Machine Learning* 20 (1995), 273–297.
- [28] CRIPPEN, R. E. Calculating the vegetation index faster. *Remote Sensing of Environment* 34, 1 (1990), 71–73.
- [29] DEMETRIOU, D. Uncertainty of OpenStreetMap data for the road network in Cyprus. In *Fourth International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2016)* (2016), K. Themistocleous, D. G. Hadjimitsis, S. Michaelides, and G. Papada-vid, Eds., vol. 9688, International Society for Optics and Photonics, SPIE, pp. 43 – 52.

- [30] DOOSTFATEMEH, I., AND BALEGHI, Y. Comprehensive urban area extraction from multispectral medium spatial resolution remote-sensing imagery based on a novel structural feature. *International Journal of Remote Sensing* 37 (09 2016), 4225–4242.
- [31] DOPIDO, I., LI, J., GAMBA, P., AND PLAZA, A. A New Hybrid Strategy Combining Semisupervised Classification and Unmixing of Hyperspectral Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7 (08 2014), 1–11.
- [32] EIGEN, D., AND FERGUS, R. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture. In *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 2650–2658.
- [33] ESTIMA, J., AND PAINHO, M. Exploratory Analysis of OpenStreetMap for Land Use Classification. In *Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information* (New York, NY, USA, 2013), Association for Computing Machinery, p. 39–46.
- [34] FARIDATUL, M., AND WU, B. Automatic Classification of Major Urban Land Covers Based on Novel Spectral Indices. *ISPRS International Journal of Geo-Information* 7 (11 2018), 453.
- [35] FOODY, G. Status of Land Cover Classification Accuracy Assessment. *Remote Sensing of Environment* 80 (04 2002), 185–201.
- [36] FORGHANI, M., AND DELAVAR, M. R. A Quality Study of the OpenStreetMap Dataset for Tehran. *ISPRS International Journal of Geo-Information* 3, 2 (2014), 750–763.
- [37] FREUND, Y., AND SCHAPIRE, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* 55, 1 (1997), 119–139.
- [38] FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189 – 1232.
- [39] GAO, B. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment* 58, 3 (1996), 257–266.
- [40] GOODFELLOW, I., BENGIO, Y., AND COURVILLE, A. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [41] HAKLAY, M. How good is OpenStreetMap information? A comparative study of OpenStreetMap and Ordnance Survey datasets for London and the rest of England. *Environment and Planning B: Planning and Design* 37 (07 2010), 682–703.
- [42] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.
- [43] HECHT, R., KUNZE, C., AND HAHMANN, S. Measuring Completeness of Building Footprints in OpenStreetMap over Space and Time. *ISPRS International Journal of Geo-Information* 2, 4 (2013), 1066–1091.

- [44] HEROLD, M., SCEPAN, J., AND CLARKE, K. C. The Use of Remote Sensing and Landscape Metrics to Describe Structures and Changes in Urban Land Uses. *Environment and Planning A: Economy and Space* 34, 8 (2002), 1443–1458.
- [45] HINTON, G. E., SRIVASTAVA, N., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv e-print* (2012).
- [46] HUANG, B., ET AL. Large-Scale Semantic Classification: Outcome of the First Year of Inria Aerial Image Labeling Benchmark. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium* (2018), pp. 6947–6950.
- [47] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 2261–2269.
- [48] HUANG, S., TANG, L., HUPY, J., WANG, Y., AND SHAO, G. A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing. *Journal of Forestry Research* 32 (05 2020).
- [49] IDOL, T., HAACK, B., AND MAHABIR, R. Radar and optical remote sensing data evaluation and fusion; a case study for Washington, DC, USA. *International Journal of Image and Data Fusion* 6 (2015), 138 – 154.
- [50] IOFFE, S., AND SZEGEDY, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv e-print* (2015).
- [51] JAMES, G., WITTEN, D., HASTIE, T., AND TIBSHIRANI, R. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.
- [52] JOKAR ARSANJANI, J., MOONEY, P., ZIPE, A., AND SCHAUSS, A. *Quality Assessment of the Contributed Land Use Information from OpenStreetMap Versus Authoritative Datasets*. Springer International Publishing, 2015, pp. 37–58.
- [53] KINGMA, D. P., AND BA, J. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015), Y. Bengio and Y. LeCun, Eds.
- [54] LACROIX, P., BIÈVRE, G., PATHIER, E., KNISS, U., AND JONGMANS, D. Use of Sentinel-2 images for the detection of precursory motions before landslide failures. *Remote Sensing of Environment* 215 (2018), 507–516.
- [55] LI, E., DU, P., SAMAT, A., XIA, J., AND MEIQIN, C. An automatic approach for urban land-cover classification from Landsat-8 OLI data. *International Journal of Remote Sensing* 36 (12 2015), 5983–6007.
- [56] LIAW, R., LIANG, E., NISHIHARA, R., MORITZ, P., GONZALEZ, J. E., AND STOICA, I. Tune: A Research Platform for Distributed Model Selection and Training. *arXiv e-print* (2018).
- [57] LIN, T.-Y., GOYAL, P., GIRSHICK, R., HE, K., AND DOLLÁR, P. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2999–3007.

- [58] LOVELAND, T., REED, B., BROWN, J., OHLEN, D., ZHU, Z., YANG, L., AND MERCHANT, J. Development of a global land characteristics database and IGBP DISCover from 1 km AVHRR data. *International Journal of Remote Sensing* 21 (04 2000), 1303–1330.
- [59] MALINOWSKI, R., ET AL. Automated Production of a Land Cover/Use Map of Europe Based on Sentinel-2 Imagery. *Remote Sensing* 12 (10 2020).
- [60] MOONEY, P., CORCORAN, P., AND WINSTANLEY, A. Towards quality metrics for OpenStreetMap. pp. 514–517.
- [61] MURPHY, K. P. *Probabilistic Machine Learning: An introduction*. MIT Press, 2021.
- [62] NACHMANY, Y., AND ALEMOHAMMAD, H. Generating a Training Dataset for Land Cover Classification to Advance Global Development. *arXiv e-print* (11 2018).
- [63] PARIS, C., BRUZZONE, L., AND FERNANDEZ-PRIETO, D. A Novel Approach to the Unsupervised Update of Land-Cover Maps by Classification of Time Series of Multispectral Images. *IEEE Transactions on Geoscience and Remote Sensing PP* (02 2019), 1–19.
- [64] PARKER, A. L., CASTELLAZZI, P., FUHRMANN, T., GARTHWAITE, M. C., AND FEATHERSTONE, W. E. Applications of Satellite Radar Imagery for Hazard Monitoring: Insights from Australia. *Remote Sensing* 13, 8 (2021).
- [65] PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 2019, pp. 8024–8035.
- [66] PEDREGOSA, F., ET AL. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [67] PLAZA, A., ET AL. Recent Advances in Techniques for Hyperspectral Image Processing. *Remote Sensing of Environment* 113 (09 2009).
- [68] POLLATOS, V., KOUVARAS, L., AND CHAROU, E. Land Cover Semantic Segmentation Using ResUNet. *arXiv e-print* (10 2020).
- [69] POWERS, R., HERMOSILLA, T., COOPS, N., AND CHEN, G. Remote sensing and object-based techniques for mapping fine-scale industrial disturbances. *International Journal of Applied Earth Observation and Geoinformation* 34 (06 2015), 51–57.
- [70] RICHTER, R. Atmospheric / Topographic Correction for Satellite Imagery. *DLR report DLR-IB 565-02/11, Wessling, Germany* 7 (01 2011).
- [71] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (2015), Springer International Publishing, pp. 234–241.
- [72] RUMELHART, D., HINTON, G. E., AND WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536.

- [73] SADEGHI, M., BABAEIAN, E., TULLER, M., AND JONES, S. The optical trapezoid model: A novel approach to remote sensing of soil moisture applied to Sentinel-2 and Landsat-8 observations. *Remote Sensing of Environment* 198 (09 2017), 52–68.
- [74] SCHULTZ, M., VOSS, J., AUER, M., CARTER, S., AND ZIPF, A. Open land cover from OpenStreetMap and remote sensing. *International Journal of Applied Earth Observation and Geoinformation* 63 (2017), 206–213.
- [75] SEAN, G., ET AL. Rasterio: geospatial raster I/O for Python programmers. <https://github.com/mapbox/rasterio>, 2013–. online cit.: 2021-05-08.
- [76] SHELHAMER, E., LONG, J., AND DARRELL, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (05 2016).
- [77] SPILSBURY, T., AND CAMPS, P. Don't ignore Dropout in Fully Convolutional Networks. *arXiv e-print* (2019).
- [78] TOMPSON, J., GOROSHIN, R., JAIN, A., LECUN, Y., AND BREGLER, C. Efficient object localization using Convolutional Networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 648–656.
- [79] ULMAS, P., AND LIIV, I. Segmentation of Satellite Imagery using U-Net Models for Land Cover Classification. *arXiv e-print* (2020).
- [80] VALI, A., COMAI, S., AND MATTEUCCI, M. Deep Learning for Land Use and Land Cover Classification Based on Hyperspectral and Multispectral Earth Observation Data: A Review. *Remote Sensing* 12 (08 2020), 2495.
- [81] VIANA, C. M., ENCALADA, L., AND ROCHA, J. The Value of OpenStreetMap Historical Contributions as a Source of Sampling Data for Multi-Temporal Land Use/Cover Maps. *ISPRS International Journal of Geo-Information* 8, 3 (2019).
- [82] WANG, T., SHI, J., LETU, H., MA, Y., LI, X., AND ZHENG, Y. Detection and Removal of Clouds and Associated Shadows in Satellite Imagery Based on Simulated Radiance Fields. *Journal of Geophysical Research: Atmospheres* 124 (06 2019).
- [83] XU, H. Modification of Normalized Difference Water Index (NDWI) to Enhance Open Water Features in Remotely Sensed Imagery. *International Journal of Remote Sensing* 27 (07 2006), 3025–3033.
- [84] YAKUBOVSKIY, P. Segmentation Models Pytorch. https://github.com/qubvel/segmentation_models_pytorch, 2020. online cit.: 2021-05-08.
- [85] YUSUF, Y., PRADHAN, B., AND IDREES, M. Spatio-temporal Assessment of Urban Heat Island Effects in Kuala Lumpur Metropolitan City Using Landsat Images. *Journal of the Indian Society of Remote Sensing* (04 2014).
- [86] ZHA, Y., GAO, J., AND NI, S. Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. *International Journal of Remote Sensing - INT J REMOTE SENS* 24 (02 2003), 583–594.

- [87] ZHANG, T., SU, J., LIU, C., CHEN, W.-H., LIU, H., AND LIU, G. Band selection in sentinel-2 satellite for agriculture applications. In *2017 23rd International Conference on Automation and Computing (ICAC)* (2017), pp. 1–6.

Prílohy

Použité OSM značky na tvorbu datasetu pre klasifikačnú úlohu

Každá trieda povrchu bola popísaná množinou tagov klúč=[hodnota_1, hodnota_2, ...] z OSM nasledovne:

- water
 - water=[reservoir, lake, river]
- trees
 - landuse=[forest]
 - natural=[wood]
- grass
 - landuse=[grass, meadow]
 - natural=[grassland]
 - landcover=[grass]
 - surface=[grass]
- building
 - amenity=[university, college]
 - building=[university, cathedral, church, supermarket, office, house, hospital, school, industrial, commercial, residential, semidetached_house, hotel, dormitory, bungalow, apartments]
- asphalt
 - amenity=[parking]
 - surface=[asphalt, paved]
- cemetery
 - landuse=[cemetery]
- square
 - place=[square]

- rock
 - natural=[bare_rock]
- agri
 - landuse=[farmland]

Použité OSM značky na tvorbu datasetu pre segmentačnú úlohu

Opäť bola množina tagov klúč=[hodnota_1, hodnota_2,...] popísaná pre jednotlivé triedy povrchov nasledovne:

- Urban
 - landuse=[residential, garages]
- Industrial & commercial
 - landuse=[railway, industrial, commercial, retail, harbour, port]
 - surface=[asphalt, paved, concrete, sett, paving_stones, unhewn_cobblestone]
 - amenity=[parking]
- Mine, dump & construction
 - landuse=[quarry, construction, brownfield, landfill]
- Artificial non-agricultural vegetated areas
 - landuse=[allotments, recreation_ground, village_green, cemetery, flowerbed]
 - leisure=[stadium, park, golf_course, sports_centre, playground, pitch, track, garden]
- Arable land
 - landuse=[farmland, farmyard, greenhouse_horticulture, farm]
- Permanent crops
 - landuse=[vineyard, orchard]
- Pastures
 - landuse=[meadow]
- Forest
 - landuse=[forest]
 - natural=[wood]
- Vegetation
 - landuse=[grass, greenfield]
 - natural=[scrub, heath, grassland]

- Open spaces with little or no vegetation

- natural=[cliff, fell, dand, scree, beach, mud, glacier, rock, bare_rock, shingle]

- surface=[grass_paver, pebblestone, rock, gravel, compacted]

- Inland wet lands

- natural=[wetland, marsh]

- Water bodies

- natural=[water]

- waterway=[riverbank]

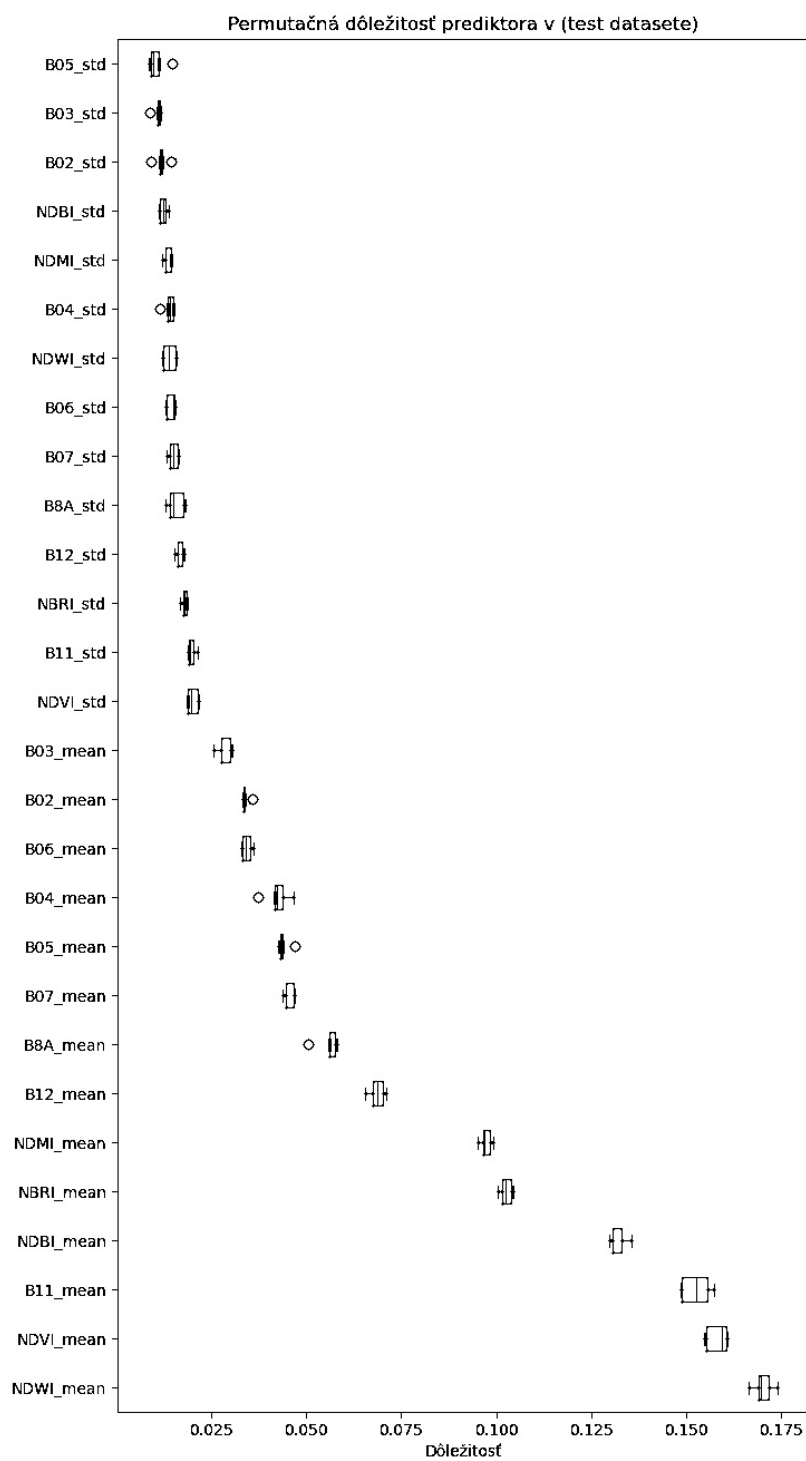
- water=[river, lake, canal, pond]

- landuse=[basin, reservoir, pond]

Hyperparametre určené pre algoritmy na klasifikačnú úlohu

Hyperparameter	Význam	Prehľadovaná množina	Optimálna hodnota
K-nearest neighbours			
n_neighbours	počet najbližších bodov v priestore prediktorov	{5, ..., 50}	17
weights	spôsob výpočtu váhy priradenej bodom v susedstve v priestore prediktorov.	{uniform, distance}	distance
Random forest			
n_estimators	počet rozhodovacích stromov v lese	{100, ..., 500}	286
max_depth	maximálna hĺbka rozhodovacieho stromu	{10, ..., 50}	42
max_features		{5, ..., 25}	14
min_samples_split	minimálny počet vzoriek na <i>split</i> v rozhodovacom strome	{2, ..., 10}	2
max_leaf_nodes	maximálny počet vrcholových uzlov (leaf) rozhodovacieho stromu	{None, 10, 25, 50, 75}	None
min_samples_leaf	minimálny počet vzoriek vyžadovaných vo vrcholovom uzle (leaf) rozhodovacieho stromu)	{1, ..., 50}	1
Gradient boosting (XGBoost cez scikit-learn API)			
n_estimators	počet boostovaných rozhodovacích stromov	použité <i>early stopping</i>	
max_depth	maximálna hĺbka rozhodovacieho stromu (komplexita modelu)	{5, ..., 25}	17
min_child_weight	minimálna hodnota sumy druhých derivácií účelovej funkcie v každom vrchole rozhodovacieho stromu	{1, ..., 8}	5
subsample	určuje podiel vzoriek z tréningového datasetu použitých na boosting každého rozhodovacieho stromu	použitá predvolená hodnota	0.8
colsample_bytree	určuje podiel prediktorov (<i>features</i>) použitých na boosting každého rozhodovacieho stromu	použitá predvolená hodnota	0.8
learning_rate	kontroluje príspevok jednotlivých boostovaných stromov do finálneho modelu	[0.0001, 0.1]	0.02
reg_lambda	kontroluje príspevok L2 regularizačného člena	{0, ..., 100}	0
gamma	definuje minimálnu hodnotu redukcie účelovej funkcie vyžadovanú na vykonanie <i>split</i> -u v rozhodovacích stromoch	[0.0, 0.8]	0.8

Permutačné dôležitosti prediktorov algoritmu k-nn



Permutačná dôležitosť prediktora v (train datasete)

