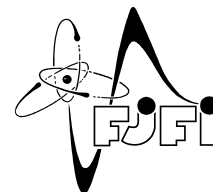


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Detekce mraků v družicových snímcích pomocí algoritmů strojového učení

Cloud detection in satellite images using machine learning algorithms

Diplomová práce

Autor: **Bc. Václav Lamich**
Vedoucí práce: **RNDr. Michal Šorel, Ph.D.**
Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Václav Lamich
Studijní program: Aplikované matematicko-stochastické metody
Název práce (česky): Detekce mraků v družicových snímcích pomocí algoritmů strojového učení
Název práce (anglicky): Cloud detection in satellite images using machine learning algorithms

Pokyny pro vypracování:

- 1) Proveďte rešerši algoritmů pro detekci oblačnosti v satelitních snímcích. Věnujte pozornost rozdílům mezi algoritmy založenými na multispektrálních příznamech a přímými metodami strojového učení.
- 2) Vyberte vhodné trénovací a validační množiny satelitních snímků s příslušnou segmentací s přihlédnutím k celoroční použitelnosti minimálně na území EU.
- 3) Navrhněte vlastní algoritmus a srovnajte výsledky s algoritmy ESA a nejlepšími dostupnými algoritmy z provedené rešerše. Analyzujte schopnost navrženého algoritmu rozlišovat mraky od sněhu a bílých objektů s vysokou odrazivostí. Pokuste se kvantifikovat množství informace, které přinese využití závislosti mezi sousedními pixely ve srovnání s metodami, které pracují s jednotlivými pixely.

Doporučená literatura:

- 1) J. B. Campbell, R. H. Wynne, Introduction to Remote Sensing (5th ed.). Guilford Press, 2011.
- 2) S. Mahajan, B. Fataniya, Cloud detection methodologies: variants and development — a review. Complex & Intelligent Systems vol. 6, 2020, 251–261.
- 3) J. Jeppesen et al., A cloud detection algorithm for satellite imagery based on deep learning. Remote Sensing of Environment vol. 229, 2019, 247-259.
- 4) S. Mohajerani, P. Saeedi, Cloud-net: an end-to-end cloud detection algorithm for LANDSAT 8 imagery, IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019.
- 5) I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

Jméno a pracoviště vedoucího diplomové práce:

RNDr. Michal Šorel, Ph.D.

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

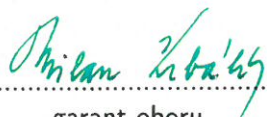
Jméno a pracoviště konzultanta:

Datum zadání diplomové práce: 31.10.2021

Datum odevzdání diplomové práce: 2.5.2022

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 1. listopadu 2021


.....

garant oboru


.....

vedoucí katedry




.....

děkan

Poděkování:

Chtěl bych zde poděkovat především svému školiteli doktoru Michalu Šorelovi za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé diplomové práce.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 5. ledna 2023

Václav Lamich

Lamich

Název práce:

Detekce mraků v družicových snímcích pomocí algoritmů strojového učení

Autor: Bc. Václav Lamich

Obor: Aplikované matematicko-stochastické metody

Druh práce: Diplomová práce

Vedoucí práce: RNDr. Michal Šorel, Ph.D., Ústav teorie informace a automatizace AV ČR, v.v.i.

Abstrakt: Sentinel 2 je družicová mise vedená Evropskou kosmickou agenturou pod programem Evropské unie Copernicus a poskytuje multi-spektrální snímky Země s vysokým rozlišením. Přesná detekce mraků ze satelitních snímků je důležitý krok před jakoukoliv vědeckou analýzou těchto snímků či automatizovaným zpracováním dat ze Sentinelu-2. Detekce oblačnosti na snímcích z družice Sentinel-2 je netriviální úkol. V naší práci popisujeme specifikaci družice Sentinel-2 a studujeme vliv oblačnosti na jednotlivé spektrální kanály. Pomocí metod strojového učení se snažíme vyvinout algoritmus, který bude mraky detekovat z jednoho družicového snímku srovnatelně nebo lépe než v současnosti používané algoritmy. Klíčem k tomu je tvorba dobré tréninkové množiny. Tuto množinu tvoříme odečtením dvou snímků stejné lokace v různých časových okamžicích, kdy jeden obsahuje oblačnost a druhý nikoliv. Díky tomu vytvoříme masku na které trénujeme algoritmy strojového učení, kterým dodáváme kombinaci informací o spektrálních pásmech a porovnááme jejich schopnost detekce mraků.

Klíčová slova: Sentinel-2, strojové učení, dálkový průzkum Země, detekce mraků

Title:

Cloud detection in satellite images using machine learning algorithms

Author: Bc. Václav Lamich

Abstract: Sentinel 2 is a satellite mission led by the European Space Agency under the European Union's Copernicus programme and provides high-resolution multi-spectral images of the Earth. Accurate cloud detection from satellite images is an important step before any scientific analysis of these images or automated processing of Sentinel-2 data. Detecting clouds in Sentinel-2 satellite images is a non-trivial task. In this paper, we describe the specification of the Sentinel-2 satellite and study the effect of clouds on the different spectral channels. Using machine learning methods, we are trying to develop an algorithm that will detect clouds from a single satellite image equally good or better than currently used algorithms. The key to this is the creation of a good training set. We form this set by subtracting two images of the same location at different points in time, where one contains clouds and the other does not. This allows us to create a mask on which we train machine learning algorithms that are equipped with combination of spectral band information and compare their ability to detect clouds.

Key words: Sentinel-2, machine learning, remote sensing, cloud detection

Obsah

Úvod	8
1 Zdroje dat	10
1.1 Sentinel 2	10
1.1.1 Specifikace Sentinelu 2	10
1.1.2 Obrazová data	11
2 Algoritmy pro rozpoznání mraků	14
2.1 Sen2Cor	14
2.2 FMask	18
2.3 MAJA	18
2.4 LaSRC	19
2.5 s2cloudless	20
2.6 CD-FCNN	21
2.7 Idepix	21
2.8 InterSSIM	22
3 Strojové učení	23
3.1 Učení s učitelem	23
3.1.1 Rozhodovací stromy	24
3.2 Učení bez učitele	25
3.3 Hybridní učení	26
3.4 Klasifikační metriky pro binární klasifikaci	26
4 Tvoření trénovacího datasetu	28
4.1 Časové rozdíly	28
4.1.1 Snímky bez sněhu	29
4.1.2 Snímky se sněhem	39
4.2 Další použitá data	48
4.2.1 Single-pixel dataset vytvořený Hollsteinem a spol.	48
4.2.2 Dataset vytvořený Baetensem a spol.	48
4.2.3 Validační snímky vytvořené Hollsteinem a spol.	51
5 Srovnání úspěšnosti variant algoritmů Náhodných lesů	54
5.1 Single-pixel náhodné lesy	54
5.1.1 Výsledky single-pixel modelů	55
5.1.2 Vizuelní srovnání single-pixel modelů	60
5.2 Multi-pixel Náhodné lesy	66

5.2.1	Výsledky multi-pixel modelu	66
5.2.2	Vizuální srovnání multi-pixel modelu	68
5.3	Srovnání single-pixel modelu, multi-pixel modelu a s2cloudless	70
	Závěr	71

Úvod

Oblačnost a její detekce je nevyhnutelnou součástí při zpracování dat z družic. Mraky signifikantně ovlivňují spektrální pásma optických senzorů. Jejich přítomnost může způsobit vážné problémy při řadě činností dálkového průzkumu Země, včetně: skládání snímků [6], korekce vlivu atmosféry [7], výpočtu vegetačních indexů [8] či klasifikaci půdního pokryvu [9]. Z tohoto důvodu je správná detekce mraků v satelitních snímcích prvním krokem při řadě činností dálkového průzkumu Země.

I přesto, že detekce mraků v obrázcích je velice složitá, fyzikální principy na kterých stojí množství metodologií jsou dobře známé [10]. Většina algoritmů detekujících mraky je založena na fyzikálních pravidlech a využívá spektrální ukazatele k rozlišení pixelu s mrakem a pixelu bez mraku [11]. V optické části spektra se mraky od jasných pixelů odlišují vysokou odrazivostí téměř ve všech vlnových délkách. Termální pásmo je užitečné pro identifikaci mraků z prostého důvodu, že mraky jsou chladnější než pixely bez mraků. Pro vylepšení detekce cirrů byla v nedávné době přidána krátkovlnná infračervená pásma obsahující indikátory absorpce vody.

Termální pásmo je klíčový zdroj dat pro detekci mraků a proto je detekce mraků pro snímky ze satelitu Sentinel-2 náročná. Některé přístupy k detekci mraků byly vyvinuty, ale pouze hrstka byla zdokumentována s důkladnou validací a hodnocením přesnosti [10]. Někteří výzkumníci experimentovali s využitím paralaxního efektu k identifikaci mraků, v případech kdy měřitelná paralaxa existuje [12]. Nedávno byly k řešení problémů s mraky u družice Sentinel-2 použity techniky strojového učení. Hollstein a spol. [5] aplikovali rozhodovací stromy a bayesovské modely na detekci mraků na snímcích Sentinel-2. Liu a spol. [13] použili ručně upravené masky mraků z produktů Sentinel Level-1C ke konstrukci sítě pro učení zbytků a sémantickou segmentaci k identifikaci oblačnostních příznaků, Shendryk a spol. trénovali hluboké konvoluční neuronové sítě ke klasifikaci mraků a zemského pokryvu na úrovni scény [14]. Tento přístup pomocí strojového učení do značné míry závisí na mracích a povrchových podmínkách ve snímcích použitých pro natrénování algoritmu. Nejdou proto dobře zobecnit pro širokou škálu scénářů [11].

Několik studií porovnávalo výkonost různých algoritmů pro detekci mraků. Jedna z těchto prací porovnávala algoritmus Fmask, používaný americkou vědeckovýzkumnou vládní agenturou United States Geological Survey pro operační zpracování dat ze satelitů Landsat. Dále pak algoritmus Sen2Cor používaný Evropskou kosmickou agenturou pro data z Sentinelu-2 a algoritmus MACCS-ATCOR Joint Algorithm (MAJA) používaný francouzskou vládní kosmickou agenturou French Center National d'études Spatiales. Tato studie zjistila, že Fmask a MAJA měli podobnou výkonost s celkovou přesností tvoření masek mraků okolo 90%, zatímco Sen2Cor dosáhl celkové přesnosti okolo 84% [16]. Aby bylo možno porovnat výsledky z těchto tří detektorů oblačnosti kvantitativně, byla potřeba referenční maska oblačnosti. Validace mrakových masek závisí na manuálně klasifikovaných snímcích nebo polygonech vybraných z velkého počtu snímků. Takovýto validační dataset pro Sentinel-2 byl vytvořen Hollsteinem a spol. [5]. Baetens a spol. nicméně uvádí, že při vybírání polygonů se autoři vyhýbali pixelům na hranách mraků, které jsou z hlediska detekce mraků klíčové a obtížné [16]. Pro kompletnější validaci a co nejmenší použití lidského operátora použili novou metodu založenou na strojovém učení Active Lear-

ning Cloud Detection. Operátor označil malý počet pixelů, které byly použity pro natrénování algoritmu strojového učení, který byl použit ke klasifikaci. Po tomto kroku lidský operátor vizuálně určil možné nepřesnosti klasifikace a označil nové pixely, kde byla klasifikace chybná nebo neurčitá. Tuto proceduru poté iterovali, dokud nezískali uspokojivou referenční masku oblačnosti [16].

Tarrío a spol. ve své studii porovnávali pět různých mraky detekujících algoritmů na snímcích z Sentinelu-2 (Sen2Cor, MAJA, LaSRC, Fmask a Tmask). Analyzovali 28 snímků ze šesti dlaždic Sentinelu-2 za použití vzorkovacího přístupu a referenčních dat interpretovaných analytikem [10]. Podle jejich zjištění nedosahuje žádný z algoritmů skvělých výsledků pro detekci mraků i stínů zároveň. Zkoušeli také integrovat dostupné algoritmy do ensamble modelu a zjistili, tento ensamble dosahuje průměrně o 2,7% lepších výsledků než nejlepší model. Tohoto zlepšení však dosáhli na úkor výpočetního výkonu. Přístup ke klasifikaci pomocí ensamble modelu je typicky založen na většinovém hlasování. Třída s největším počtem hlasů od modelů, které tvoří ensamble je vybrána jako výsledná třída. Alternativně lze zvolit model založený na náhodných lesech. Fundamentálním problémem při detekci mraků je nutný kompromis mezi falešně negativními a falešně pozitivními pozorováními. Neboli těmi pozorováními, které model chybně označí za bezmračná a respektive za obsahující mrak. Wolpert a Macready [18] vysvětlují, že pro jakýkoliv algoritmus je vysoký výkon v jedné dimenzi kompenzován nižším výkonem v dimenzích dalších. Tarrío a spol. zjišťovali, zda přístup pomocí ensamble modelu dokáže zmírnit tento kompromis.

Mezinárodní organizace The Committee on Earth Observation Satellites (CEOS) nedávno uspořádala cvičení s názvem Cloud Masking Intercomparison eXercise (CMIX). V rámci tohoto cvičení se srovnávalo deset mraky detekujících algoritmů pro data ze satelitů Sentinel-2 a Landsat 8 [17]. Tyto algoritmy se lišili v přístupu a konceptech, které byly založeny na různých spektrálních vlastnostech, prostorových a časových příznacích a také na metodách strojového učení. Výsledky algoritmů byly hodnoceny na základě existujících referenčních dat setech mrakových masek. Tyto datasety se lišili ve vzorkovacích metodách, geografickém rozdělení, vzorkovacích jednotkách (body, polygony, celé snímky) a také v generačních přístupech (experti, strojové učení). Celkově byla výkonnost algoritmů závislá na referenčním datasetu, což bylo přičteno rozdílu v tom, jak byl každý dataset vyprodukován. Srovnávané algoritmy se shodovaly v detekci hustých mraků. Naopak pro jemné a poloprůhledné mraky byla shoda mezi srovnávanými algoritmy nižší [17].

V naší práci se pokusíme pomocí technik strojového učení sestavit vlastní algoritmus. Budeme muset vytvořit a najít vhodné trénovací a validační množiny satelitních snímků s příslušnou segmentací s přihlédnutím k celoroční použitelnosti minimálně na území EU. Vytvoření takovéto množiny v dostatečné velikosti je netriviální úkol. Klíčem k našemu algoritmu bude vytvoření vlastní masky oblačnosti na které budeme tento algoritmus učit. Tuto masku se pokusíme vytvořit pomocí časových rozdílů. V podstatě se pokusíme najít dva identické snímky s tím, že jeden bude obsahovat oblačnost a druhý nikoliv a tyto snímky od sebe odečteme. Předpokládáme, že tímto přístupem bychom mohli získat skoro dokonalou masku mraků. Potřeba bezoblačného snímku tuto úlohu značně komplikuje. Protože bychom chtěli, aby algoritmus byl použitelný celoročně bude potřeba zajistit, aby sněhová pokrývka na odečítaných snímcích, byla co nejpodobnější. Sentinel-2 nám však zajistí snímek dané oblasti s 5-10 denními rozestupy, což je vzhledem k požadavku na podobnost sněhových pokrývek problém. Náš výsledný algoritmus se následně pokusíme srovnat s veřejně dostupnými algoritmy.

Kapitola 1

Zdroje dat

V naší práci se budeme snažit vyvinout algoritmus, který bude schopen rozpoznávat pixely s oblačností a pixely bez oblačnosti. Tento úkol by nebyl proveditelný bez dobrého zdroje dat a tím je pro nás dvojice družic Sentinel-2A a Sentinel-2B. V následující sekci se tedy budeme věnovat popisu dat pocházejících z těchto satelitů.

1.1 Sentinel 2

Informace vyskytující si v následující kapitole lze dohledat, jak na stránkách programu Copernicus <https://www.copernicus.eu/cs> tak na stránkách mise Sentinel <https://sentinel.esa.int/web/sentinel/missions/sentinel-2> Nebudeme se pro to na ně již dále odkazovat.

Sentinel 2 je misí v rámci programu Copernicus, která má za cíl systematicky poskytovat snímky pevniny a okrajových moří s vysokým rozlišením. Copernicus je program Evropské unie řízený evropskou komisí v koordinaci s členskými státy evropské unie, Evropskou kosmickou agenturou (ESA), Evropskou organizací pro využívání meteorologických družic (EUMETSAT) a dalšími organizacemi.

Sentinel-2A byl vynesena na oběžnou dráhu raketou Vega, která startovala z Kourou v Francouzské Guyaně v roce 2015. Po ní následovala družice Sentinel-2B v roce 2017. Další v pořadí by měla být družice Sentinel-2C v roce 2024 a po ní ještě družice Sentinel-2D.

Skvělé je, že se na data z těchto družic nevztahují žádné omezení na reprodukci a redistribuci. A to jak pro komerční, tak pro nekomerční využití. Data jsou poskytovány v plném rozsahu a zdarma. Lze je nalézt na následujícím odkazu. <https://scihub.copernicus.eu/dhus/#/home>

1.1.1 Specifikace Sentinelu 2

Mise Sentinel 2 se skládá ze soustavy dvou polárních družic (2A a 2B), které jsou umístěny na stejné slunečně synchronní oběžné dráze ve střední nadmořské výšce 786km. Tyto satelity jsou vzájemně rozfázovány v úhlu 180° a obsahují multispektrální optický senzor. Tato kompozice by měla zaručit průlet nad stejnou oblastí rovníku každých 5 dní. Pro jeden satelit se jedná o 10 dní. Například v Praze je tento čas pouhé 2-3 dny. Družice jsou schopny snímat zemský povrch mezi 56° jižní šířky (což odpovídá Hornovu mysu v Ohňové zemi) a 84° severní šířky (což odpovídá severnímu Grónsku) s orbitálním rozsahem 290km.

Životnost družice je 7,25 roku, což zahrnuje tříměsíční fázi uvedení do provozu na oběžné dráze. Baterie a pohonné hmoty byly zajištěny na 12 let provozu, včetně manévru na konci životnosti.

Nejdůležitější součástí satelitu je optický senzor MultiSpectral Instrument (dále jen MSI). MSI pasivně sbírá sluneční světlo odražené od Země. Nová data jsou do přístroje získávána při pohybu družice

po její dráze. Příchozí světelný paprsek je rozdělen na filtru a soustředěn na dvě samostatné ohniskové roviny v přístroji; jednu pro viditelná a blízká infračervená pásma (VNIR- Visible and Near-Infra-Red) a druhou pro krátkovlnná infračervená pásma (SWIR- Short Wave Infra-Red). Spektrální rozdělení každého pásma na jednotlivé vlnové délky se provádí pomocí proužkových filtrů namontovaných na horní straně detektorů.

MSI nám poskytuje třináct pásem od viditelného spektra (VIS-visible spectrum) až po blízké infračervené (NIR-Near-Infra-Red) a SWIR. Přehled těchto pásem naleznete v tabulce 1.1. Hodnoty v ní uvedené odpovídají družici Sentinel-2A. Rozdíly oproti Sentinel-2B jsou v řádu nižších jednotek, proto je neuvádíme.

Pásmo	Centralní vlnová délka [nm]	Šířka pásma [nm]	Rozlišení [m]	komentář
1	442,7	21	60	pobřežní aerosol
2	492,4	66	10	modrá
3	559,8	36	10	zelená
4	664,6	31	10	červená
5	704,1	15	20	vegetační červený okraj
6	740,5	15	20	vegetační červený okraj
7	782,8	20	20	vegetační červený okraj
8	832,8	106	10	NIR
8A	864,7	21	20	vegetační červený okraj
9	945,1	20	60	vodní pára
10	1373,5	31	60	SWIR, detekce cirru
11	1613,7	91	20	SWIR
12	2202,4	175	20	SWIR

Tabulka 1.1: Přehled pásem MSI

Cirrus je typ mraku, který je definován jako "oddělená oblaka ve formě bílých jemných vláken nebo bílých či převážně bílých skvrn, které jsou tvořeny ledovými krystalky". [1]

1.1.2 Obrazová data

1.1.2.1 Level 0

Data ze satelitů mise Sentinel 2 procházejí různými úrovněmi zpracování. První úroveň je Level 0, který obsahuje komprimovaná nezpracovaná obrazová data ve formátu ISP (Instrument Source Packet). Produkt levelu 0 je tvořen granulátem o rozměrech 25km kolmo na oběžnou dráhu a 23km podél oběžné dráhy. Průměrná oběžná dráha obsahuje 3500 těchto granulátů. Tato data nejsou poskytována uživatelům a tvoří základ následného zpracování na levelu 1.

1.1.2.2 Level 1A

Level 1 používá konsolidovaná data z levelu 0 jako input. Tento level je tvořen třemi sublevely. Jsou to Level 1A, kde je vyvinut geometrický model, který umožňuje lokaci jakéhokoliv pixelu. Jednotlivé

souřadnice vždy odkazují na střed daného pixelu. Data z tohoto levelu opět nejsou poskytována uživateli.

1.1.2.3 Level 1B

Další na řadě je Level 1B jehož procesní sekvence vypadá následovně.

1.
 - korekce tmavého signálu
 - korekce nerovnoměrnosti odezvy pixelů
 - crosstalk korekce
 - identifikace vadných pixelů
 - obnova pásem s vysokým prostorovým rozlišením (dekonvoluce a odstranění šumu)
 - binování 60 m spektrálních pásem
2. Převzorkování na společné geometrické síti pro registraci mezi globálním referenčním obrazem (GRI) a referenčním pásmem (ve výchozím nastavení B4).
3. Shromáždění vazebních bodů ze dvou snímků pro registraci mezi GRI a referenčním pásmem.
4. Filtrování vazebních bodů pro registraci obrazu-GRI: filtrování vazebních bodů v několika oblastech. Je vyžadován minimální počet vazebních bodů.
5. Zpřesnění modelu pohledu pomocí inicializovaného modelu pohledu a pozemních kontrolních bodů (GCP). Výstupní zpřesněný model zajišťuje registraci mezi GRI a referenčním pásmem.
6. Komprese snímků úrovně 1B využívá algoritmus JPEG2000.

Tato data také nejsou vydávána uživatelům.

1.1.2.4 Level 1C

Zpracování úrovně 1C zahrnuje radiometrické a geometrické korekce včetně ortorektifikace a prostorové registrace v globálním referenčním systému s přesností na subpixel.

Zpracování úrovně 1C je rozděleno do následujících kroků:

- Sdružení dlaždic: výběr předem definovaných dlaždic protínajících stopu požadovaného snímku.
- Výpočet převzorkovací sítě: umožňuje propojení nativního geometrického obrazu s cílovým geometrickým obrazem (ortorektifikovaným).
- Převzorkování každého spektrálního pásma v geometrii ortografického obrazu pomocí převzorkovacích mřížek a interpolačního filtru. V tomto kroku se také vypočítá odrazivost TOA (Top-Of-Atmosphere).
- Výpočet masek: generují se masky mraků.
- Komprese výsledných snímků úrovně 1C pomocí algoritmu JPEG2000 a hlavičky geografických snímků kódované v jazyce GML.

Výsledkem je dlaždice 100km na 100km.

Tato úroveň je již poskytována uživatelům. Na této úrovni se již také nachází maska oblačnosti. Také je zde vygenerován obrázek v pravých barvách (dále jen TCI-true color image).

1.1.2.5 Level 2A

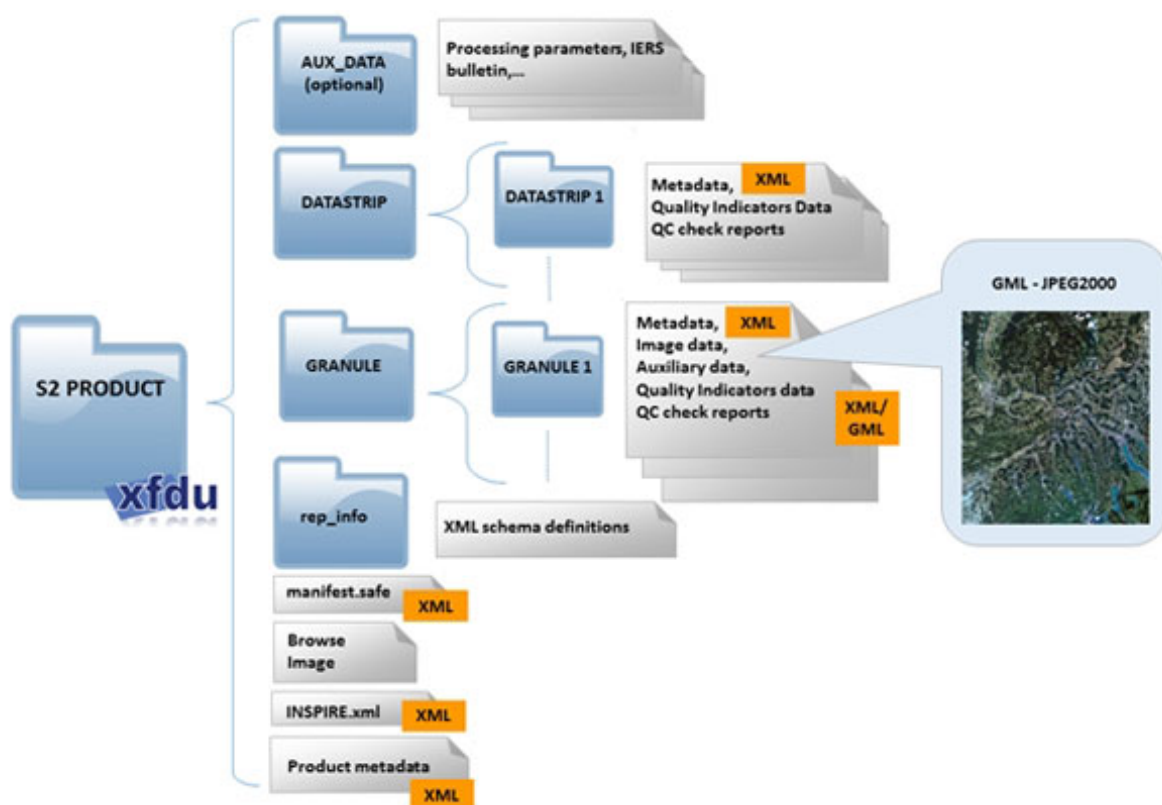
Po levelu 1 přichází na řadu Level 2A, který zahrnuje i klasifikaci scény (SCM-scene classification map), kterou rozděluje do následujících kategorií. Žádná data, saturovaná nebo vadná data, tmavé oblasti, stíny mraků, vegetace, žádná vegetace, voda, neklasifikováno, středně velká pravděpodobnost mraku, velká pravděpodobnost mraku, tenký cirrus a sněh.

Další užitečnou věcí je přítomnost grafického zobrazení pravděpodobností, že se na daném pixelu nachází oblačnost.

Také zde dochází k atmosférické korekci aplikované na ortofotografické produkty levelu 1C. Hlavním výstupem úrovně 2A je ortofotografický produkt s korekcí odrazivosti od dna atmosféry (BOA-Bottom-Of-Atmosphere). Dalšími výstupy jsou mapa optické tloušťky aerosolu (AOT-aerosol optical thickness), mapa vodní páry (WV-water vapour) a TCI, které jsou dostupné v rozlišení 10m, avšak po převzorkování z 20m.

1.1.2.6 Složka s daty

Uvádíme, zde strukturu složky s daty pro snadnější přístup k datům. Obrázek je k nalezení na <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/data-formats>



Obrázek 1.1: Struktura složky s daty

Kapitola 2

Algoritmy pro rozpoznání mraků

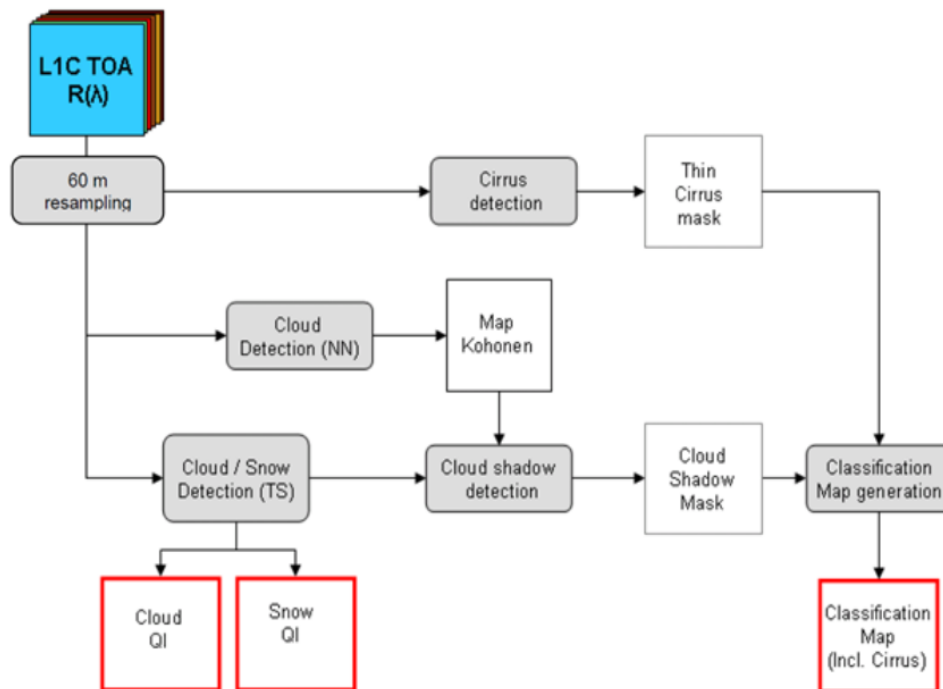
V této kapitole se budeme věnovat soudobým algoritmům, které mají schopnost vytvořit mrakovou masku pouze z dat ze satelitu Sentinel-2. Z jedné části se jedná o algoritmy používané předními světovými institucemi. Algoritmus Sen2Cor je používán Evropskou kosmickou agenturou, algoritmus Fmask je používán agenturou United States Geological Survey, algoritmus MAJA je používán agenturou French Center National d'études Spatiales a algoritmus s2cloudless je používán webovou službou pro satelitní snímky Sentinel Hub. Z druhé části se jedná o méně známé detekční algoritmy, které se objevili na mezinárodním cvičení Cloud Masking Intercomparison eXercise pořádaném organizací The Committee on Earth Observation Satellites, jako jsou CD-FCNN, Idepix a InterSSIM.

2.1 Sen2Cor

ESA nám spolu s daty poskytuje i masku oblačnosti pro jednotlivé dlaždice. Tato maska se nachází na úrovni 1C a lze jí najít ve složce Quality Indicators data.

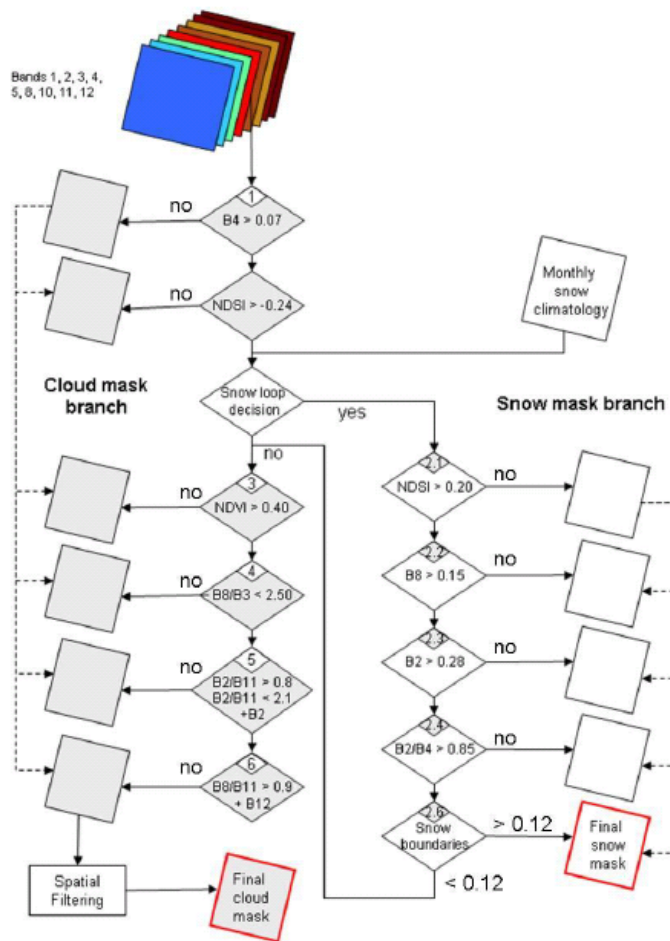
Sen2Cor je procesor pro generování a formátování produktu Sentinelu-2 úrovně 2A. Tento algoritmus mimo jiné počítá a vytváří mapu AOT, vodních par a klasifikace scény. Také vytváří mapu pravděpodobností přítomnosti sněhu či oblačnosti.

Algoritmus klasifikace scény zahrnuje detekci mraků a sněhu, detekci cirrů, detekci stínů oblačnosti a generaci klasifikační mapy. Jeho jednotlivé procesní kroky můžeme vidět na obrázku 2.1. Obrázek byl přejet z následujících stránek. <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm>



Obrázek 2.1: Klasifikační procesní moduly (šedé) a jejich výstupy (červené)

Nejjednodušší způsob, jak blíže představit fungování tohoto algoritmu je jeho grafická vizualizace, kterou můžete vidět na obrázku 2.2 nebo na stránkách <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm>



Obrázek 2.2: Algoritmus pro rozpoznání mraků/sněhu

Tento algoritmus je založen na sérii kroků prahové filtrace. Pro bližší vysvětlení zde uvedeme jednotlivé kroky.

- Krok 1a: Při tomto kroku vyloučíme pixely bez mraku aplikováním dvou prahů na odrazivost pásma B4, které odpovídá červené barvě. Pokud je odrazivost pásma B4 nižší než 0.07, je pixel považován za bezmračný. Pokud je odrazivost tohoto pásma vyšší než 0.25, pak se v pixelu nachází mrak. Tyto pixely, pak jdou do druhého kroku tohoto procesu. Pixelům, které mají odrazivost B4 mezi těmito prahy je lineárně přiřazena pravděpodobnost mraku a jdou do kroku 1b.
- Krok 1b: V tomto kroku je pro pixely spočítáno NDSI.
NDSI je definováno následovně:

$$\text{Normalised Difference Snow Index(NDSI)} = \frac{B3 - B11}{B3 + B11} \quad (2.1)$$

Pixely s hodnotami NDSI, které jsou menší než -0.1, jsou považovány za bezmračné. Pixely, které mají hodnotu NDSI vyšší než 0.2, považujeme za mrak. Pixelům mezi těmito hodnotami je lineárně přiřazena pravděpodobnost mraku, která je násobena jejich předchozí pravděpodobností. Pixely s pravděpodobností mraku vyšší než 0 pokračují do dalšího kroku.

- Krok 2: V tomto kroku je série prahové filtrace závislejší, jak na NDSI, tak na hodnotách odrazivosti v pásmech B8, B2 a poměru odrazivosti pásem $\frac{B2}{B4}$.
- Krok 3: V tomto kroku je pro pixely spočítáno NDVI.

NDVI je definováno následovně:

$$\text{Normalised Difference Vegetation Index(NDVI)} = \frac{B8 - B4}{B8 + B4} \quad (2.2)$$

Pixely s hodnotami NDVI vyššími než je náš práh T2 jsou považovány za bezmračné a klasifikovány jako vegetace. Pro pixely s hodnotou NDVI nižší než práh T1 se nic nezmění. Pixely s hodnotou NDVI mezi T1 a T2 je lineárně přiřazena pravděpodobnost mraku, která je násobena jejich předchozí pravděpodobností. Pixely s pravděpodobností mraku vyšší než 0 pokračují do dalšího kroku.

- Krok 4: Zde se počítá poměr odrazivosti pásem $\frac{B8}{B3}$. Pokud je vyšší než 0.4, pak jsou pixely považovány za bezmračné a jsou klasifikovány jako vegetace. Pro pixely s hodnotou poměru odrazivosti pásem $\frac{B8}{B3}$ nižší než 0.36 se nic nezmění. Pixelům mezi těmito hodnotami je lineárně přiřazena pravděpodobnost mraku, která je násobena jejich předchozí pravděpodobností. Pixely s pravděpodobností mraku vyšší než 0 pokračují do dalšího kroku.
- Krok 5.1: V tomto kroku identifikujeme pixely s půdou pomocí poměru odrazivosti pásem $\frac{B2}{B11}$. Pokud je tento poměr nižší než 1.5, pak je pixel považován za bezmračný a klasifikován jako půda v klasifikaci scény. Pro pixely s hodnotou poměru odrazivosti pásem $\frac{B2}{B11}$ vyšší než 2.5 se nic nezmění. Pixelům mezi těmito hodnotami je lineárně přiřazena pravděpodobnost mraku, která je násobena jejich předchozí pravděpodobností. Pixely s pravděpodobností mraku vyšší než 0 pokračují do dalšího kroku.
- Krok 5.2: V tomto kroku identifikujeme pixely s vodní plochou pomocí poměru odrazivosti pásem $\frac{B2}{B11}$. Pokud je tento poměr vyšší než práh T2, pak je pixel považován za bezmračný a klasifikován jako vodní plocha. Pro pixely s hodnotou poměru odrazivosti pásem $\frac{B2}{B11}$ nižší než práh T2 se nic nezmění. Pixelům mezi těmito hodnotami je lineárně přiřazena pravděpodobnost mraku, která je násobena jejich předchozí pravděpodobností. Pixely s pravděpodobností mraku vyšší než 0 pokračují do dalšího kroku.
- Krok 6: Zde identifikujeme vysoce reflexní skály a písek v poušti. Využíváme k tomu jejich odrazivost v pásmu 11, která je nižší než odrazivost v pásmu 8. Mraky mají tyto odrazivosti opačně. Pixely s poměrem odrazivosti pásem $\frac{B8}{B11}$ nižším než 0.9 jsou považovány za bezmračné a klasifikovány jako půdní/pouštní pixely. Pro pixely s hodnotou poměru odrazivosti pásem $\frac{B8}{B11}$ vyšší než 1.1 se nic nezmění. Pixelům mezi těmito hodnotami je lineárně přiřazena pravděpodobnost mraku, která je násobena jejich předchozí pravděpodobností. Pixely s pravděpodobností mraku vyšší než 0 pokračují do dalšího kroku.
- Krok 7: Jedná se o volitelný krok související s prostorovým filtrováním.

V této fázi jsou k dispozici masky pravděpodobností mraků a sněhu. Toto volitelné filtrování zohledňuje mírnou chybnou registraci spektrálních pásem S2 ve výšce mraků. Spektrální pásma S2 jsou totiž v přízemní výšce koregistrována pomocí DEM. Prostorová filtrace také pomáhá omezit falešnou detekci mraků vyskytující se na hranicích vysoce kontrastních oblastí, jako jsou obrysy řek nebo pobřeží. Na konečnou masku mraků se použije mediánový filtr a následně dilatační operátor. Velikost jádra filtrů může být 3 x 3 nebo 5 x 5 v závislosti na požadované úrovni redukce

artefaktů. Toto prostorové filtrování se provádí pro generování tří tříd oblačnosti a třídy řídkého cirru v klasifikační mapě.

2.2 FMask

FMask je algoritmus pro vyhodnocování mraků [19], který byl původně vyvíjen pro satelity Landsat 5 a 7, později také pro Landsat 8. Zahrnuje odrazivost povrchu a jas teplot na termálním infračerveném kanálu. Byla vyvinuta také varianta této metody pro Sentinel-2 bez informací z termálního infračerveného kanálu, který Sentinel-2 neobsahuje [20].

FMask nejprve spočítá potenciální mrakové vrstvy společně s vrstvou stínu mraků na základě prahu jedné hodnoty. Po tomto prvním kroku je spočtena pravděpodobnost mraku na základě statistiky spočtené na pixelech, které nejsou v potenciální mrakové vrstvě. Pixely s nejvyšší pravděpodobností jsou také zahrnuty do potenciální mrakové vrstvy [16]. Tyto kroky zahrnují viditelné pásmo, blízké infračervené pásmo, krátkovlnné infračervené pásmo a pásma cirrů. Dále je aplikována spektrálně-kontextuální optimalizace za účelem zjednodušení falešné identifikace mraků od světlých a bílých povrchů, jimiž se myslí například sníh nebo městská zástavba. Stíny mraků jsou detekovány projekcí segmentovaných mrakových objektů na základě geometrie mezi pozorovacími úhly, úhlem slunce a výškou mraků [17].

2.3 MAJA

Následující sekce čerpá z velké části z článku [21]. Nebudeme se na něj tedy, již dále odkazovat.

MAJA je aplikovatelná na satelity, které provádějí opakovaná pozorování v podobných pozorovacích úhlech, jako je právě Sentinel-2. Její mraky a stíny detekující metody zahrnují několik testů, které využívají multi-spektrální a multi-temporální vlastnosti povrchu, mraků a stínů, ke klasifikaci různých typů pixelů. Hlavní test přítomnosti mraku detekuje pixely, pro které povrchová odrazivost v modrém pásmu ostře vzroste [17]. K tomu abychom mohli tento test provést pro snímek ze dne D , tak potřebujeme bezmračný referenční snímek, který není vždy k dispozici a proto je často sestaven z částečně bezmračných snímků dané lokality. Pixel je označen multi-temporálním kritériem jako pixel obsahující mrak jestliže:

$$\rho_{blue}(D) - \rho_{blue}(D_r) > 0.03 \cdot \left(1 + \frac{D - D_r}{30}\right), \quad (2.3)$$

kde $\rho_{blue}(D)$ je odrazivost pixelu v modrém pásmu, korigovaná pro Rayleighův rozptyl, ve dni D . D_r je nejbližší bezmračného bezmračný den pro danou lokalitu. $D - D_r$ je vyjádřen ve dnech a prahovací hodnota závisí na počtu dní mezi dnem D a D_r . Když je počet dní mezi daty malý, pak je prahovací hodnota bližší hodnotě 0.03. Nicméně, pokud je $D - D_r = 30$, pak se prahovací hodnota zdvojnásobí.

Toto kritérium je velmi eficientní pro separaci mraků a bezmračných pixelů nad pevninou, nicméně není univerzální. Za prvé toto kritérium nad vodními plochami nacházejícími se na pevnině nefunguje dobře. Tyto plochy jsou náchylné ke změnám v odrazivosti v důsledku slunečního svitu, zákalu či pěny. Vodní pixely musí být vyřazeny před výpočtem mrakové masky. Za druhé je možná záměna tenkého mraku a vysoká optická tloušťka aerosolů. Některé mraky jsou moc tenké nato by bylo možné je detekovat, zatímco vysoká optická tloušťka aerosolů by mohla být brána jako mrak. Za třetí je možná náhlá variace v povrchové odrazivosti kvůli zemědělským intervencím nebo přírodním úkazům jako jsou požáry, sníh nebo extrémní sucho. Za účelem vypořádání se s těmito problémy byli přidány dva testy, které zkoumají zda náhlý nárůst povrchové odrazivosti zapříčinil mrak či nikoliv.

Pixel, který je na základě rovnice 2.3 označen jako pixel obsahující mrak není finálně označen jako pixel obsahující mrak pokud platí jedna z následujících podmínek.

1. Pokud je změna odrazivosti v červeném pásmu, mnohem větší než změna odrazivosti v modrém pásmu. Toto se děje poměrně často, pokud je pole sklizeno nebo zoráno nebo vegetace rychle uschne. Tomu odpovídá následující rovnice:

$$\rho_{red}(D) - \rho_{red}(D_r) > 1.5 \cdot (\rho_{blue}(D) - \rho_{blue}(D_r)), \quad (2.4)$$

kde $\rho_{red}(D)$ je odrazivost pixelu v červeném pásmu, korigovaná pro Rayleighův rozptyl

2. Pokud jsou odrazivosti v okolí pixelu korelované s těmi v okolí pixelu v jednom z deseti snímků pořízených před dnem D. Takový test byl již použit Lyapustinem a spol. [22] a je založen na předpokladu, že je velice nepravděpodobné, že mrak zůstane na stejném místě se stejným tvarem. Dobrý korelační koeficient může být způsobem pouze dobrou průhledností atmosféry. Použitím deseti předchozích snímků místo složeného snímku získáváme možnost se vypořádat s možnou počáteční chybou v složeném snímku. Byl zaznamenán případ, kde byli na pole instalovány plastové skleníky. Pixely těchto skleníků byli rovnici 2.3 označeny za obsahující mrak a z toho důvodu nebyly použity k aktualizaci složeného snímku a stejně tak následující dny. Protože je korelace mezi dvěma po sobě jdoucími snímky skleníku vysoká, je pixel překlasifikován na základě této podmínky na pixel bezmračný.

Multi-temporální detekční metoda, použitá v MAJA, je rekurentním algoritmem, který vyžaduje, aby byli snímky zpracovány v chronologickém pořadí. Rekurentní algoritmus tedy musí být inicializován a to na základě rovnice 2.3. První složený snímek je tedy první snímek bez pixelů s mraky. První práh je poměrně vysoký, tak aby nebyli jasné povrchy označeny jako pixely s mraky. Důsledkem tohoto přístupu je, že tenké mraky jsou téměř nezachyceny v první masce. Aby se zamezilo degradaci kvality pro první snímky z časové řady, je implementován režim zpětného zpracování. Prvních 6 až 10 snímků je zpracováno v opačném chronologickém pořadí, tak aby byla získána správná maska pro snímek, který je prvním v časové řadě. Následně jsou všechny snímky zpracovány v chronologickém pořadí s počáteční maskou dobré kvality.

MAJA je použitelná i pro satelity Landsat. Autoři MAJA nepoužili termální infračervený pás, který je dostupný pro Landsat, ale není dostupný pro Sentinel-2. Nicméně využili krátkovlnného infračerveného pásma, které Sentinel-2 obsahuje. Toto pásmo je využito u testu na sníh, který využívá NDSI a je stejný jako v rovnici 2.1. Mraky a sníh mají totiž vysokou odrazivost v zeleném pásmu, zatímco v krátkovlnném infračerveném pásmu je odrazivost sněhu nižší. Proto je jasný pixel označen jako sněhový, pokud je jeho $NDSI > 0.6$.

2.4 LaSRC

Následující sekce čerpá z velké části z článku [23]. Nebudeme se na něj tedy, již dále odkazovat.

The Land Surface Reflectance Code je obecný algoritmus, který provádí atmosférickou korekci a zaměřuje se na odstranění atmosférických efektů spojených s pořizováním optických satelitních snímků. LaSRC odhaduje povrchovou odrazivost, přičemž bere do úvahy absorpci atmosférickými plyny a rozptyl molekul a aerosolů. Během atmosférické korekce generuje LaSRC několik kvalitativních vrstev, které zahrnují i masku mraků [17].

Jedním z nejdůležitějších kroků v rámci zpracování je získání optické tloušťky aerosolů, které je založeno na vztahu mezi červeným, modrým a krátkovlnným infračerveným pásmem. Po získání mapy optické tloušťky aerosolů v 30 metrovém rozlišení odhaduje algoritmus LaSRC inverzní reziduum, které je metrikou, toho jak je inverze optické tloušťky aerosolů dobrá.

$$\text{Reziduum} = \sqrt{\frac{1}{3} \left((\rho_s^1 - r_{1,4}\rho_s^4)^2 + (\rho_s^2 - r_{2,4}\rho_s^4)^2 + (\rho_s^6 - r_{6,4}\rho_s^4)^2 \right)}, \quad (2.5)$$

kde $\rho_s^1, \rho_s^2, \rho_s^4$ a ρ_s^6 jsou hodnoty povrchové odrazivosti v pásmech pobřežních aerosolů a modrých, červených a krátkovlnných infračervených pásmech. $r_{1,4}, r_{2,4}$ a $r_{6,4}$ jsou poměry odrazivosti jednotlivých pásem s tím, že poměr konkrétních pásem je určen indexy.

Tato reziduální metrika je hlavním kritériem pro detekci hustých mraků, protože buď bude zabráněno inverznímu procesu optické tloušťce aerosolů od konvergence nebo budou hodnoty reziduální metriky vysoké. K určení pixelů obsahujících mraky a těch bezmračných by měl být zvolen specifický práh. Pro snímky ze satelitu Landsat 8 byl tento práh nastaven na 0.05. Na cvičení [17] byl pro snímky ze satelitu Sentinel-2 použit stejný práh. Pixely, které sousedí s pixelem označeným jako obsahující mrak, jsou klasifikovány jako 'sousedící s mrakem'. Logika pro takovéto označení je následující. Hrany mezi pixely obsahující mraky a těmi bezmračnými jsou ze své podstaty nejednoznačné a velká část nepřesnosti mnohých algoritmů pochází z klasifikace pixelů na těchto hranách, toto jsme ověřili empiricky při sestavování vlastního algoritmu. Taková to klasifikace zajistí, že nespolehlivé pixely jsou odstraněno z dalšího zpracování. Algoritmus LaSRC také odhaduje metriku, které ukazuje dopad aerosolů na hodnoty povrchové odrazivosti, které vypadá následovně:

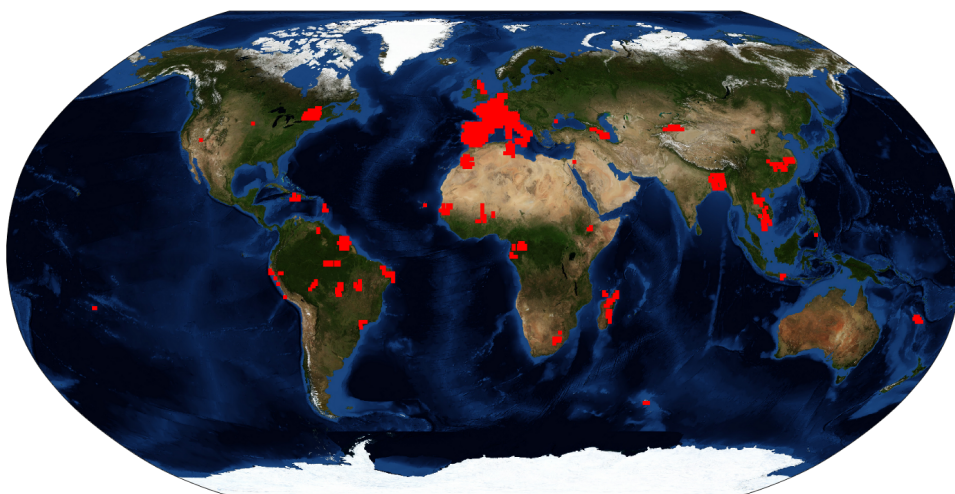
$$\text{Dopad aerosolů} = \rho_s^1 - \rho_T^1, \quad (2.6)$$

kde ρ_s^1 a ρ_T^1 jsou povrchové odrazivosti a povrchové odrazivosti pro aerosoly pro pásmo 1, které náleží pobřežním aerosolům. Pixely s hodnotou dopadu aerosolů větší než 0.03 jsou označeny jako 'vysoké hodnoty aerosolů' a měli by být vyloučeny z dalšího zpracování spolu s pixely detekovanými jako obsahující mrak.

2.5 s2cloudless

Následující sekce čerpá z velké části z článku [15]. Nebudeme se na něj tedy, již dále odkazovat.

s2cloudless je automatizovaný mraky detekující algoritmus pro snímky ze satelitu Sentinel-2, který je založen na gradient boostingu, konkrétně LightGBM. Byl vyvinut EO Research teamem z Siner-gise. Model byl trénován na 14140 dlaždicích Sentinelu-2 z nichž 596 bylo geograficky unikátních. Následující obrázek 2.3 tyto lokace zobrazuje a byl přejat z <https://medium.com/sentinel-hub/improving-cloud-detection-with-machine-learning-c09dc5d7cf13>



Obrázek 2.3: Distribuce lokací datasetu použitého k vývoji algoritmu s2cloudless

Pro označení bezmračných a mraky obsahujících pixelů v tomto datasetu použili algoritmus MAJA. Pro validaci natrénovaného algoritmu byl použit dataset vytvořený Hollsteinem [5]. Algoritmus je mono-temporální a nezohledňuje žádný prostorový kontext proto může být použit pro jakékoliv rozlišení. Může také na rozdíl od jiných algoritmů být použit na zprůměrovaných hodnotách odrazivosti ze Sentinelu-2 přes libovolnou uživatelem definovanou geometrii a stále poskytovat smysluplné výsledky [17]. Vstupními příznaky jsou Sentinel-2 Level-1C TOA hodnoty odrazivosti následujících pásem. B01, B02, B04, B05, B08, B8A, B09, B10, B11 a B12 s tím, že výstupem algoritmu je mapa pravděpodobností přítomnosti mraku. Uživatel může změnou prahu převést tuto mapu na mrakovou masku. Doporučenou hodnotou tohoto prahu je 0.4 kvůli minimalizaci falešně negativních pixelů. Uživatelé také mohou aplikovat další morfologické operace během konverze pravděpodobnostní mapy na mrakovou masku. Těmito operacemi jsou konvoluce pravděpodobnostní mapy a dilatace binární mrakové masky diskem [17].

2.6 CD-FCNN

Následující sekce čerpá z velké části z článku [17]. Nebudeme se na něj tedy, již dále odkazovat.

CD-FCNN byl navržen Image and Signal Processing skupinou z Univerzity Valencie. Je založen na přístupu pomocí hlubokého učení a je aplikovatelný na multispektrální snímky ze satelitů jako Landsat 8 a Sentinel-2. Trénování takového algoritmu vyžaduje obrovské datasety označených snímků, které musí odrážet velkou rozmanitost typů mraků, povrchu a atmosférických podmínek. Takovéto datasety většinou, ale bohužel buď neexistují nebo nejsou veřejně dostupné. Pro Landsat 8 existuje L8Biome dataset [24], který splňuje tyto požadavky a byl proto použit z 80% pro natrénování plně konvoluční neuronové sítě, která může být převedena na detekci mraků ze snímků Sentinelu-2. Zbýlých 20% datasetu tvořil dataset L8SPARCS [25]. Po minimální adaptaci na data ze Sentinelu-2, v oblasti selekce pásem a prostorového rozlišení, byl model natrénovaný na datech z Landsatu 8 přímo použit na snímky ze Sentinelu-2. Navrhovaná architektura neuronové sítě byla založena na upraveném U-Netu, s signifikantně menším počtem trénovacích parametrů a nižšími výpočetními náklady. Síť obsahuje zhruba 96000 parametrů, což je okolo 1% počtu parametrů původního U-Netu. Trénovaná byla na pixelech s rozlišením 30m. Vstupem byly snímky velikosti 32x32 pixelů a batch size byla 64. Model byl natrénován na minimalizaci pixelové binární nákladovou funkci křížové entropie. Byl použit Adamův optimalizační algoritmus stochastického gradientního sestupu. Inicializační rychlost učení byla 10^{-5} a 120 epoch bylo použito k natrénování finální neuronové sítě. Byly použity následující pásma. B2, B3, B4, B8, B11 a B12 ze Sentinelu-2. Výstup sítě byl dán sigmoidovou aktivační funkcí, které poskytuje spojité hodnoty mezi 0 a 1. Byl použit práh 0.5 pro rozlišení mezi 0 pixelem bez mraku a 1 pixelem s mrakem. Tento práh je klíčový s ohledem na rovnováhu mezi počtem falešně pozitivních a falešně negativních pixelů. Výkon algoritmu může klesat v komplexních případech, kdy je ve snímcích přítomnost sněhu, městských částí nebo pobřežních linií. Upravení prahu může zlepšit kompromis mezi počtem falešně negativních a falešně pozitivních pixelů pro specifické datasety.

2.7 Idepix

Následující sekce čerpá z velké části z článku [17]. Nebudeme se na něj tedy, již dále odkazovat.

IdePix (Identification of Pixel properties) je multisenzorový pixelový identifikační nástroj. Poskytuje identifikační algoritmy pro širokou škálu senzorů jako jsou Sentinel-2 MSI, Sentinel-3 OLCI, MERIS, Landsat 8, MODIS, VIIRS, Proba-V nebo SPORT VGT. IdePix klasifikuje pixely do serií kategorií pro další zpracování za použití mono-temporálního přístupu a dalších informací. Technický design Idepixu je specifický pro dané nástroje a může obsahovat rozhodovací stromy, pravděpodobnostní kombinace

kalkulovaných příznaků nebo neuronové sítě. IdePix pro Sentinel-2 je hlavně založen na technice rozhodovacích stromů pro výpočet mrakové masky, ale i na geometrických kalkulacích pro stíny mraků a stíny hor. Na rozdíl od mnoha jiných pixelově založených identifikačních nástrojů je finální klasifikace IdePixu neexkluzivní. To znamená, že jednomu pixelu, lze přiřadit více kategorií. Jeden pixel, tak může být označen jako zem a mrak, pro poloprůhledné mraky nad zemí nebo jako zem a sněh pro zem pokrytou sněhovou pokrývkou atd. Tato implementace umožňuje nejuniverzálnější použití při rozřazování do kategorií na základě požadavků uživatele. Pixely na kraji mraků jsou označeny za pomoci dilatačního filtru. V principu je pak vytvořena kolem mraku nárazníková zóna. Šířka této nárazníkové zóny může být nastavena uživatelem.

2.8 InterSSIM

Následující sekce čerpá z velké části z článku [17]. Nebudeme se na něj tedy, již dále odkazovat.

Algoritmus pro detekci mraků InterSSIM, vyvinut EO Research týmem z Sinergise [26], je multi-temporálním rozšířením algoritmu s2cloudless. Na rozdíl od algoritmu s2cloudless, InterSSIM bere v potaz časové a prostorové souvislosti při detekci mraků. Vstupní data a parametry jsou pro InterSSIM stejná jako pro s2cloudless, ale přidáme zde i předchozí satelitní pozorování. Algoritmus funguje na deseti pásmech Sentinelu-2 a kromě pravděpodobnosti výskytu oblačnosti zahrnuje i další funkce. Těmi jsou: prostorově zprůměrované hodnoty odrazivosti, minimální a průměrné hodnoty odrazivosti ze všech předchozích pozorování, maximum, průměr a směrodatná odchylka indexu strukturální podobnosti. Index strukturální podobnosti je vypočítán mezi pozorováními, pro které je počítána maska oblačnosti a všemi předchozími pozorováními. Výstupem algoritmu je mapa pravděpodobnosti výskytu oblačnosti pro cílový časový rámec. Tato mapa může být převedena do masky oblačnosti pomocí stejné procedury jako u algoritmu s2cloudless.

Kapitola 3

Strojové učení

Cílem strojového učení je navrhnout a vyvinout algoritmus, který umožňuje systémům používat empirická data, zkušenosti a adaptovat se na změny, které se objevují v jejich prostředí. Jedním z hlavních zaměření strojového učení je automaticky indukovat modely a nacházet pravidla a vzorce skrývající se v trénovacích datech. Strojové učení kombinuje nejrůznější techniky a přístupy z různých oborů, které zahrnují, jak pravděpodobnost a statistiku, tak například psychologii, informační teorii a umělou inteligenci [2].

Většina algoritmů strojového učení lze rozřadit do třech kategorií. První je tzv. učení s učitelem, které pracuje s označenými daty. Další je učení bez učitele, které pracuje s daty bez označení a třetí je kombinace dvou předešlých, tzv. hybridní učení.

Pro učení s učitelem jsou nejúspěšnější metody založeny buď na rozhodovacích stromech a nebo na neuronových sítích. Neuronové sítě jsou formou strojového učení, která využívá spojovací výpočetní uzly zvané neurony, které v podstatě popisují jakoukoli lineární nebo nelineární funkci. Jsou postaveny na schématu, které kopíruje strukturu našeho mozku. To umožňuje odvodit či klasifikovat výstup na základě daných vstupů, ale bez použití implicitního vzorce [29]. My se nicméně budeme věnovat hlavně metodám založených na rozhodovacích stromech protože pro problémy nižší dimenze bývají výsledky neuronových sítí a rozhodovacích stromů srovnatelné, ale algoritmy založené na rozhodovacích stromech jsou jednodušší a rychlejší.

3.1 Učení s učitelem

Při učení s učitelem jsou vstupní data označená. Tato data jsou sekvencí párů vstupních dat x_i a výstupních dat y_i , které jsou korektně označeny. Cílem algoritmu je naučit se funkci f , která zohledňuje dosud viděné x_i a y_i .

$$f(x_i) = y_i \quad \forall i \in n, \quad (3.1)$$

kde n je počet pozorování. Tuto funkci nazýváme klasifikátor, pokud jsou y_i diskrétní hodnoty a regresor pokud jsou výstupy y_i spojité. Cílem této funkce je správně predikovat výstupy y_i pro vstupy x_i , které algoritmus ještě neviděl. [2]

Exekuci učení s učitelem lze rozdělit do pěti hlavních kroků.

1. Shromážděte trénovací dataset
2. Určete prediktor (feature), který bude reprezentovat vstupní data
3. Zvolte učící se algoritmus

4. Natrénujte algoritmus

5. Vyhodnot' te přesnost algoritmu pomocí sady testovacích dat

V prvním kroku si tedy musíme zjistit, jaká data potřebujeme a jaké bude jejich dostatečné množství, tak abychom korektně mohli reprezentovat situace z reálného světa pro nějakou specifickou aplikaci. Obecnou poučkou je, že čím více dat máme, tím lépe můžeme natrénovat algoritmus. Shromáždit a správně označit velké množství dat může být časově a finančně velmi náročný proces.

Druhým krokem je určit set prediktorů nebo také vektor prediktorů, které budou reprezentovat vstupní data. Každý prediktor reprezentuje charakteristiku objektů nebo událostí, které klasifikujeme. Velký prediktorový vektor signifikantně zvyšuje komplexitu klasifikace, ale také může zvýšit přesnost klasifikace. Prediktorový vektor by měl tudíž být dostatečně velký, aby zajistil reprezentaci důležitých prediktorů, ale dostatečně malý, aby se vyhnul zbytečné komplexitě.

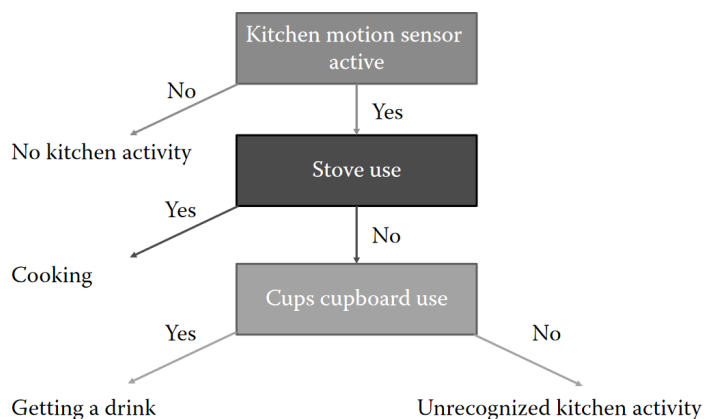
Při vybírání algoritmu, který se učí, pro specifický úkol, musíme zvážit mnoho faktorů. Důležitý je obsah a velikost trénovacího datasetu, šum v systému, přesnost označení, ale také heterogenitu a redundanci vstupních dat. Nejpoužívanější algoritmy strojového učení s učitelem zahrnují podpůrné vektorové stroje (support vector machines), Bayesovské klasifikátory, rozhodovací stromy, Markovské modely, podmíněné náhodné pole, a k-nejbližší souseď (k-nearest neighbor).

Ve čtvrtém kroku se algoritmus pomocí trénovacích dat naučí funkci, které nejlépe odpovídá vstupním/výstupním trénovacím instancím.

V posledním kroku vyhodnotíme přesnost naučené funkce s pomocí testovacího datasetu, kde testovací dataset se liší od trénovacího. Vyhodnocujeme tedy, jak přesně algoritmus klasifikuje vstupy z testovacího setu na základě funkce, kterou se naučil z trénovacího datasetu [2].

3.1.1 Rozhodovací stromy

Rozhodovací stromy se dají charakterizovat rychlostí výpočtu, jednoduchou interpretací pravidel a škálovatelností pro multidimenzionální datasety [3]. Cílem rozhodovacích stromů je vytvořit model, který bude predikovat výstupní hodnoty na základě vstupních hodnot v prediktorovém vektoru. Pro lepší představu, o čem mluvíme, když zmiňujeme rozhodovací stromy poslouží následující obrázek.



Obrázek 3.1: Jednoduchý příklad rozhodovacího stromu [2]

Rozhodovací stromy se skládají z uzlů z nichž každý koresponduje s nějakou proměnou z prediktorového vektoru. Z každého uzlu vedou hrany k potomkům uzlu ze kterého vycházíme. Počet hran

vedoucích z uzlu je závislý na počtu možných hodnot proměnné uložené v uzlu. Uzel ze kterého již nevede hrana nazýváme listem a představuje možnou výstupní hodnotu našeho rozhodovacího stromu. Strom na obrázku 3.1 má tři uzly, šest hran a čtyři listy. Nejvýše položený uzel ze kterého vycházíme nazýváme kořenem.

Rozhodovací stromy se snaží napodobit lidský rozhodovací proces více než jiné algoritmy. Poradí si s kategorickými proměnnými, spojitými proměnnými i jejich kombinací.

Hlavní nevýhodou rozhodovacích stromů je jejich sklon k přetrénování (overfitting). To znamená, že i malá změna v trénovacích datech může vést k velké změně ve výsledcích.

3.1.1.1 Náhodné lesy

Sklon k přetrénování rozhodovacích stromů ukazuje na nedostatečnou schopnost generalizace takovýchto algoritmů. Jednou z cest, jak vylepšit predikční schopnosti jednotlivých rozhodovacích stromů je tzv. bagging (bootstrap aggregating).

Algoritmus Náhodné lesy představený Breimanem v roce 2001 kombinuje bagging, tj. tvorbu nového převzorkovaného datasetu z původního trénovacího datasetu a vytvoření nového stromu z těchto dat, s náhodným výběrem prvků při tvorbě stromu. Tento přístup pomáhá vyhnout se přetrénování rozhodovacích stromů a zlepšuje jejich generalizační schopnosti. Opakováním tohoto postupu a průměrováním odhadů získáme algoritmus s lepšími generalizačními schopnostmi [4].

V případě klasifikační úlohy potom jednotlivé stromy hlasují o výsledné predikci. Toto hlasování je založeno na principu většinového hlasování, tj. výsledné třídění je třídění, které získá nejvíce hlasů od jednotlivých stromů.

Tento algoritmus strojového učení je jeden z nejpoužívanějších algoritmů pro klasifikaci a regresi díky své robustnosti a přesnosti. Náhodné lasy jsou také ideální volbou pro řešení našeho problému s ohledem na nízkou dimenzi úlohy. Pro menší dimenze jsou Náhodné lesy podobně úspěšné jako neuronové sítě, ale jsou ve srovnání s nimi rychlejší. Zvláště pokud použijeme rozumné množství stromů. Jedná se tedy o velmi efektivní volbu pro náš problém.

3.1.1.2 Gradient boosting

Posílené rozhodovací stromy (boosted decision trees) se skládají z iterativně indukovaných rozhodovacích stromů, vážených na základě jejich předchozího výkonu, přičemž predikce je stejná jako u náhodných lesů. V prvním kroku tedy natrénujeme rozhodovací strom na trénovacích datech a přidáme ho přenásobený konstantou do finálního modelu. Spočítáme reziduum tohoto stromu a další strom již trénujeme na tomto reziduu. Každý strom má tedy k dispozici informaci o chybě předchozího stromu.

Algoritmus gradient boosting je kombinací metod gradientní optimalizace a vážení stromů s špatnými výkony. Tento algoritmus je trénován na nepřesných predikcích za použití ztrátové funkce L (Loss function).

3.2 Učení bez učitele

Sběr označených dat spotřebovává zdroje a čas. Přesného označení dat je navíc často těžké dosáhnout. Učení bez učitele tedy pracujeme s neoznačenými daty. Cílem je najít inherentní vzorce v datech, které mohou být použity k určení správných výstupů pro nová data. Předpokládáme, že ve vstupních datech existuje struktura taková, že se určité vzory objevují častěji než jiné a chceme vidět co se obecně stává a co ne. Hledáme vlastně nějaký odhad hustoty pravděpodobnosti.

Jednou z forem učení bez učitele je tzv. shlukování (clustering), což je v podstatě klasifikace dat do skupin na základě vzorců vyznačených v datech [2].

Shlukování je technika strojového učení, která se používá ke seskupení dat do skupin podle podobnosti jejich vlastností. Cílem shlukování je nalézt přirozené skupiny v datech bez předem daného znalosti, do jakých skupin patří jednotlivá data. Tato technika se může použít pro řadu úloh, jako je například segmentace zákazníků, analýza textu nebo pro zjednodušení složitých datových struktur.

Existuje mnoho různých algoritmů pro clustering, které se liší svým přístupem k rozdělování dat do shluků. Mezi nejčastěji používané patří K-Means, Hierarchický clustering a DBSCAN.

K-Means je iterativní algoritmus, který se snaží nalézt shluky tak, aby suma vzdáleností mezi body v shluku a středem shluku byla co nejmenší. Algoritmus při každé iteraci vybírá několik počátečních středů shluků a poté přiřazuje data k nejbližšímu středu. Poté jsou středy shluků aktualizovány na základě nově přiřazených dat. Tento proces se opakuje, dokud se středy shluků nemění nebo dokud není dosaženo požadovaného počtu iterací. K-Means ale může mít problémy s nalezením shluků, které nejsou kulatého tvaru, a navíc je nutné předem zvolit počet shluků, což může být obtížné [28].

Hierarchický clustering je algoritmus, který vytváří hierarchii shluků pomocí rekurzivního aglomerativního nebo divizivního přístupu. V aglomerativním přístupu jsou data nejprve považována za jednotlivé shluky a postupně jsou seskupována do větších shluků. Tento proces pokračuje, dokud nejsou všechna data seskupena do jednoho velkého shluku. V divizivním přístupu je proces obrácený, tj. začíná se s jedním velkým shlukem a postupně se rozděluje na menší shluky. Hierarchický clustering může být pomalý pro velké datové soubory, ale dokáže zachytit i složité struktury v datech [28].

3.3 Hybridní učení

Algoritmy hybridního učení využívají, jak označených, tak neoznačených dat. Označená data typicky tvoří malé procento trénovacího datasetu. Cílem hybridního učení je pochopit, jak kombinování označených a neoznačených dat může změnit chování při učení a navrhnout algoritmus, který takovou kombinaci využije. Hybridní učení je velmi slibný přístup, protože dokáže využít neoznačená data k vylepšení učení s učitelem v případech, kde další označení dat je příliš nákladné nebo složité [2].

Jedním z příkladů hybridního učení je použití učení s učitelem pro vytvoření základního modelu a poté použití učení bez učitele k doladění modelu na základě nerozpoznaných případů. Toto je například užitečné při klasifikaci obrázků, kde můžeme použít učení s učitelem pro rozpoznání základních objektů a poté použít učení bez učitele k doladění modelu pro méně běžné objekty. Hybridní přístup může vést k lepším výsledkům než použití jednoho modelu, protože může využít silné stránky obou přístupů.

3.4 Klasifikační metriky pro binární klasifikaci

Přesnost klasifikátoru či regresoru je měřena pomocí klasifikačních metrik. Existují různé metriky, jak pro klasifikátor, tak pro regresor. Různé metriky se zaměřují na různé aspekty evaluace modelu a jsou vhodné pro různé situace.

Dříve jsme výstupy označili $y_i \quad \forall i \in n$. Nyní označíme ještě výstupy predikované naším klasifikátorem jako $\hat{y}_i \quad \forall i \in n$. Jsme tedy v situaci, kdy y_i a \hat{y}_i může nabývat pouze hodnot 0 nebo 1.

Při porovnání skutečných hodnot y_i a predikovaných \hat{y}_i mohou nastat čtyři různé případy. Tyto případy rozřadíme do následujících skupin.

1. pravdivě pozitivní (TP) - počet případů, kdy $\hat{y}_k = y_k = 1$
2. pravdivě negativní (TN) - počet případů, kdy $\hat{y}_k = y_k = 0$

3. falešně pozitivní (FP) - počet případů, kdy $\hat{y}_k = 1 \wedge y_k = 0$

4. falešně negativní (FN) - počet případů, kdy $\hat{y}_k = 0 \wedge y_k = 1$

Z tohoto rozřazení je patrné, že platí:

$$TP + TN + FP + FN = n. \quad (3.2)$$

Tyto hodnoty lze přehledně zapsat do tzv. konfuzní matice, která vypadá následovně.

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} \quad (3.3)$$

Všechny metriky, které si představíme a budeme používat je odvozena právě od této matice.

Jednou z nejpřirozenějších výkonnostních metrik, je přesnost (accuracy), která je definovaná jako poměr správně klasifikovaných pozorování oproti všem pozorováním, což lze zapsat následovně:

$$\text{přesnost} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{n}. \quad (3.4)$$

Tedy pokud nám vyjde, že přesnost = 0.9 znamená to, že náš klasifikátor správně predikoval výstupní hodnotu v 90% případů.

Další z důležitých metrik odvozených z konfuzní matice je tzv. precision a recall. Precision je poměr správně klasifikovaných prvků v druhém sloupci konfuzní matice oproti všem prvkům v druhém sloupci konfuzní matice. Recall je stejný, ale pracuje s druhým řádkem konfuzní matice. Tyto metriky vypadají následovně:

$$\text{Precision} = \frac{TP}{FP + TP} \quad (3.5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.6)$$

V ideálním případě bychom chtěli model, který správně identifikuje všechny pravdivě pozitivní případy a pouze pravdivě pozitivní případy, abychom dosáhli perfektní precision a perfektního recallu. V reálném světě se ale bohužel musíme potýkat s kompromisem mezi recalledem a precision.

Kdybychom chtěli maximalizovat například recall stačilo by, abychom náš model trénovali na datech ve kterých bychom označili všechny pixely jako $y_i = 1 \quad \forall i \in n$ bez ohledu na to zda se v daném místě mrak nachází či nikoliv. Takový model by, pak ve všech případech predikoval $\hat{y}_i = 1 \quad \forall i \in n$. Model by ve všech případech označil správně pixely ve kterých se vyskytuje oblačnost a neoznačil žádný pixel jako bezmračný a dosáhl tak $FN = 0$ a následujícího:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1}{1 + 0} = 1. \quad (3.7)$$

Netřeba říkat, že takový model by k ničemu i přes vynikající recall.

Ukážeme si tedy metriku, která bude pro ohodnocení našeho výsledného modelu lepší a to takzvané F1-skóre které v sobě kombinuje, jak precision, tak recall. To je definované následující rovnicí.

$$\text{F1-skóre} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.8)$$

Tato metrika tedy obsahuje, jak informaci o počtu falešně pozitivních klasifikací, tak o počtu falešně negativních klasifikací a bude tedy nejvhodnějším kritériem, pro hodnocení našeho výsledného modelu. Model totiž obdrží vysoké F1-skóre pokud bude, jak precision, tak recall vysoký a naopak nízké skóre, pokud budou precision i recall nízké.

Kapitola 4

Tvoření trénovacího datasetu

Vytvoření vhodného trénovacího datasetu je při tvorbě algoritmů strojového učení klíčová věc. Výběr trénovací množiny bude mít zásadní vliv na naše výsledky. Z povahy našeho problému existuje velmi málo dat, na kterých bychom mohli náš algoritmus natrénovat. Potřebujeme totiž získat dataset ve kterém budou jednotlivé pixely označeny jako pixely s mrakem, či pixely bez oblačnosti.

Takový dataset lze získat několika způsoby. Například tím, že si pixely označíme vlastnoručně, což by bylo extrémně zdlouhavé a neefektivní. Nehledě na to, že i zkušený člověk bude mít často problém rozlišit pixel ve kterém se nachází mrak či stín, či se na místě nachází průhledný mrak a tak dále.

Dalším způsobem je použití, již existujících algoritmů, které detekují mraky a použití jejich masek. Tento přístup zvolili Zupanc a spol. při tvorbě algoritmu s2cloudless [15]. Ti svůj model natrénovali na výsledcích algoritmu MAJA. My bychom chtěli vytvořit vlastní trénovací dataset, který nebude limitován potenciálními nedostatky jiných modelů.

4.1 Časové rozdíly

Myšlenkou, která stojí v samotném centru naší práce je vytvořit si trénovací množinu pomocí časových rozdílů. Vezmeme dva stejné satelitní snímky, z nichž jeden bude s oblačností a druhý bez. Poté od sebe odečteme hodnoty v jednotlivých pixelech a výsledkem by měla být téměř dokonalá maska oblačnosti. Nicméně budeme muset rozlišovat mezi snímky, které obsahují sněh a mezi těmi co nikoliv. Na základě naší zkušenosti tento přístup funguje lépe pro snímky, které neobsahují sněhové plochy.

Než budeme ale moci hodnoty v jednotlivých pixelech odečíst budeme muset vzít do úvahy rozdílné jas obou snímků. Budeme se tedy snažit najít konstantu, kterou přenásobíme jeden ze snímků, abychom dostali oba dva na přibližně stejnou úroveň jasu.

Náš problém můžeme zapsat takto:

$$m_1 \in \mathbf{R}_{+,0}^n, \quad m_2 \in \mathbf{R}_{+,0}^n, \quad K \in \mathbf{R}_{+,0} \quad (4.1)$$

$$m_2 = K \cdot m_1, \quad (4.2)$$

kde m_1 a m_2 jsou vektory obsahující seřazené hodnoty jednotlivých pixelů, které náležejí jednotlivým snímkům. Těchto pixelů je n a z povahy věci nemohou nabývat záporných hodnot. K je přenásobovací konstanta, kterou hledáme. Abychom ji našli budeme se snažit vyřešit následující úlohu.

$$\min_{K \in \mathbf{R}_{+,0}} \|m_2 - K \cdot m_1\|_2 = \min_{K \in \mathbf{R}_{+,0}} \sqrt{\sum_{i=1}^n (m_{2i} - K \cdot m_{1i})^2} \quad (4.3)$$

Zderivováním podle K a položením této derivace rovné nule nalezneme hledané minimum. Rovnice, kterou dostaneme vypadá následovně:

$$\frac{1}{2 \sqrt{\sum_{i=1}^n (m_{2i} - K \cdot m_{1i})^2}} \cdot 2 \sum_{i=1}^n (m_{2i} - K \cdot m_{1i}) \cdot (-m_{1i}) = 0 \quad (4.4)$$

Z této rovnice už pomocí několika ne příliš složitých úprav dostaneme výsledek.

$$K = \frac{\sum_{i=1}^n m_{2i} m_{1i}}{\sum_{i=1}^n m_{1i}^2} \quad (4.5)$$

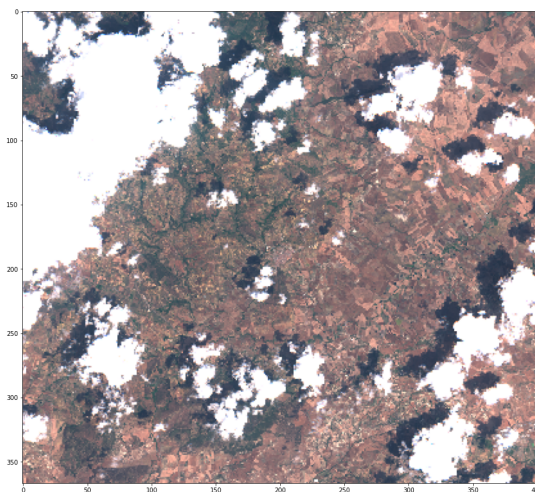
Poté, co jsme zjistili formuli pro výpočet naší hledané konstanty K se ji můžeme pokusit nalézt pro jednotlivé snímky, které budeme používat pro trénování modelu. Konstantu K z povahy věci nemůžeme hledat pokud jeden z vektorů m_1 nebo m_2 bude obsahovat i pixely obsahující mraky. Jako postačující odhad toho, kde se v snímku nachází mraky použijeme s2cloudless detektor a pixely, které podle tohoto algoritmu obsahují oblačnost z výpočtu vyřadíme.

4.1.1 Snímky bez sněhu

Důležité je také rozmyslet se v jakých pásmech budeme operovat. Po pečlivém zvážení a prohlédnutí všech pásem jsme se rozhodli pro kombinaci rozdílů pásem B2 a B10. Pásmo B2 odpovídá modré barvě a všechna oblačnost je v tomto pásmu vidět velice dobře a to platí i pro velmi jemné mráčky, které v ostatních pásmech téměř nelze pozorovat. Pásmo B10, kterému se někdy říká cirrusové pásmo je zase velice dobré pro pozorování právě mraků typu cirrus. Nyní si na dvou lokacích demonstrujeme tvorbu našich masek.



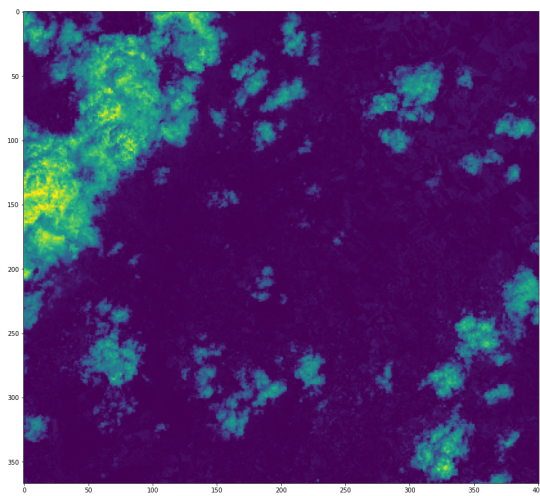
Obrázek 4.1: Sicílie bez oblačnosti



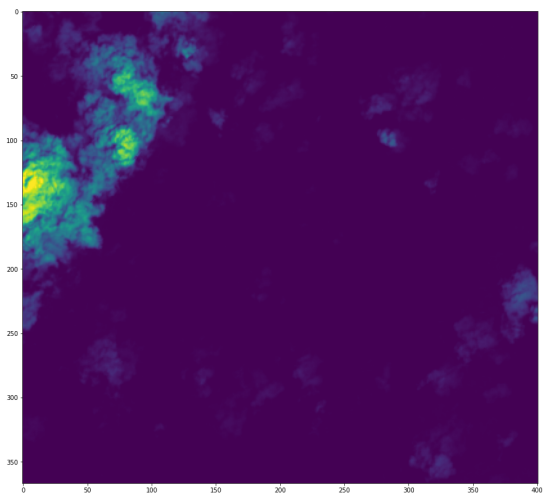
Obrázek 4.2: Sicílie s oblačností

Prvním krokem je tedy získání dvou snímků dané lokality s rozumným časovým odstupem. Na obrázcích 4.1 a 4.2 tento odstup tvoří cca 40 dní.

Dalším krokem je odečtení obou snímků od sebe pro jednotlivé pixely pro pásmo B2 (jeden z nich je přenásoben konstantou K 4.5) a následná normalizace tohoto rozdílu. To samé poté provedeme pro pásmo B10. Výsledek vidíme na obrázku 4.3 a 4.4.

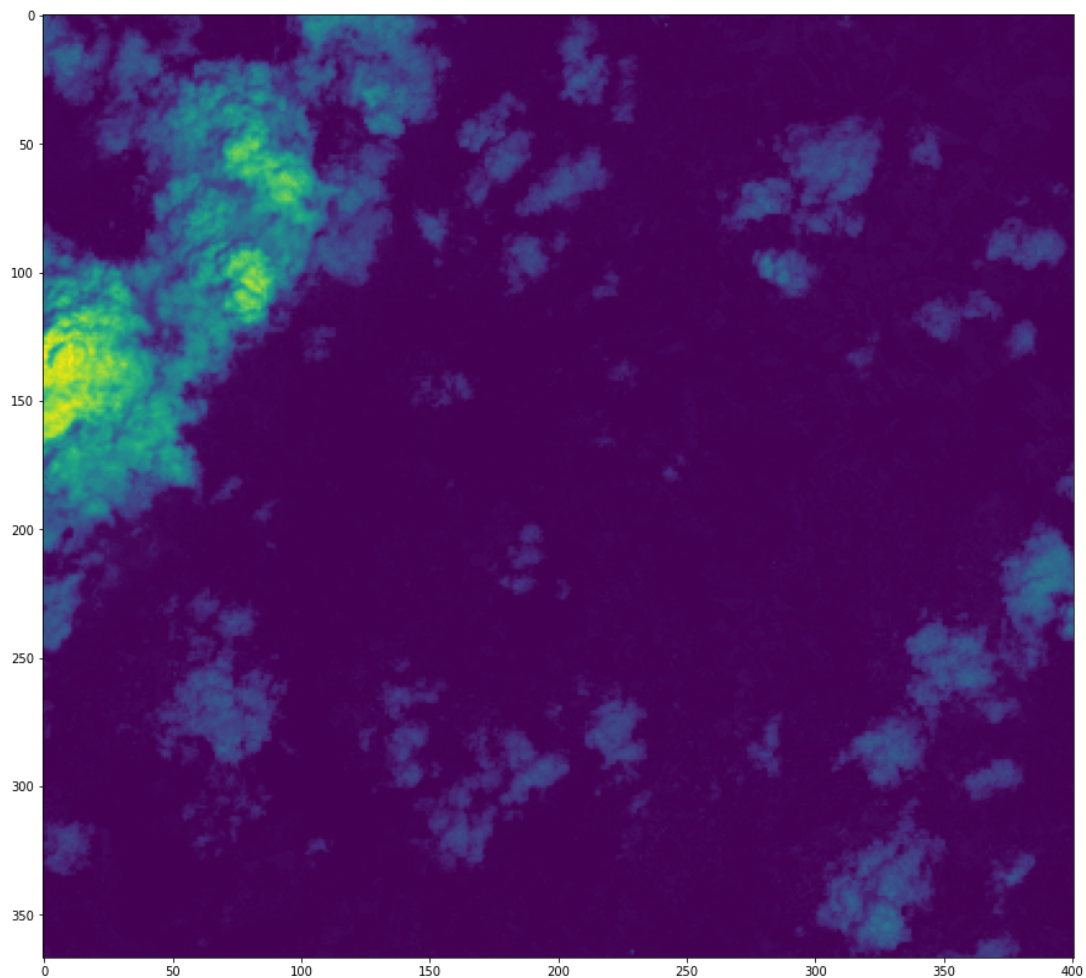


Obrázek 4.3: Sicílie rozdíl B2



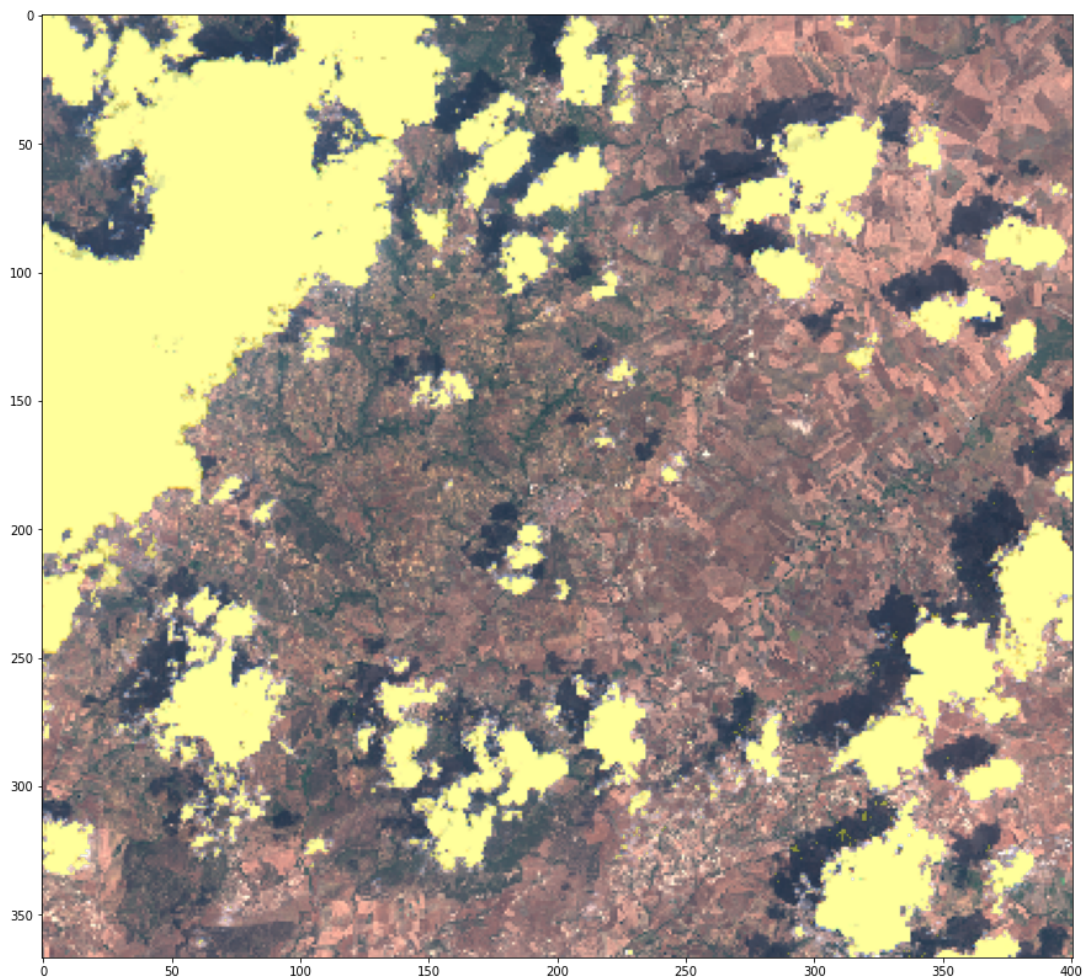
Obrázek 4.4: Sicílie rozdíl B10

Sečtením těchto rozdílů se pokusíme potlačit jejich nedostatky. Výsledek můžeme vidět na obrázku 4.5.



Obrázek 4.5: Sicílie součet rozdílů B2 a B10

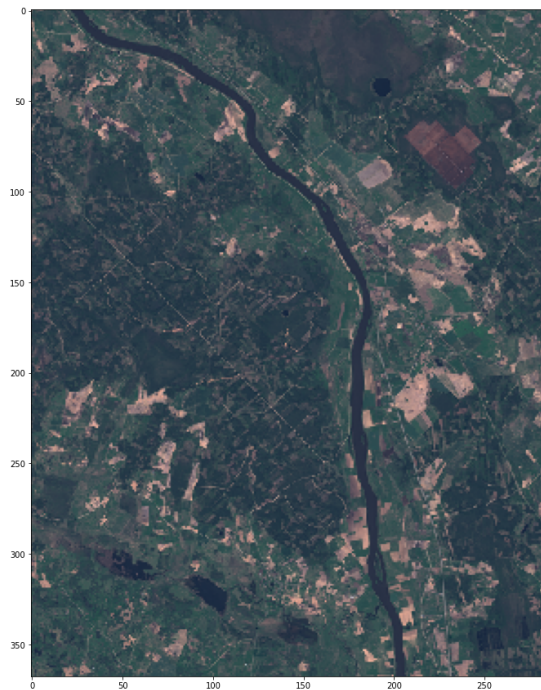
Jako poslední je třeba v součtu rozdílů na obrázku 4.5 zvolit vhodný práh. Cokoliv nad tímto prahem označíme jako oblačnost a cokoliv pod ním jako bezmračný pixel. Jako prvotní odhad tohoto prahu jsme použili procentuální zastoupení oblačnosti v snímku podle masky vytvořené modelem s2cloudless. Následnou iterací, která byla potřeba provést pro každý snímek zvlášť jsme došli k masce, kterou můžete vidět na obrázku 4.6.



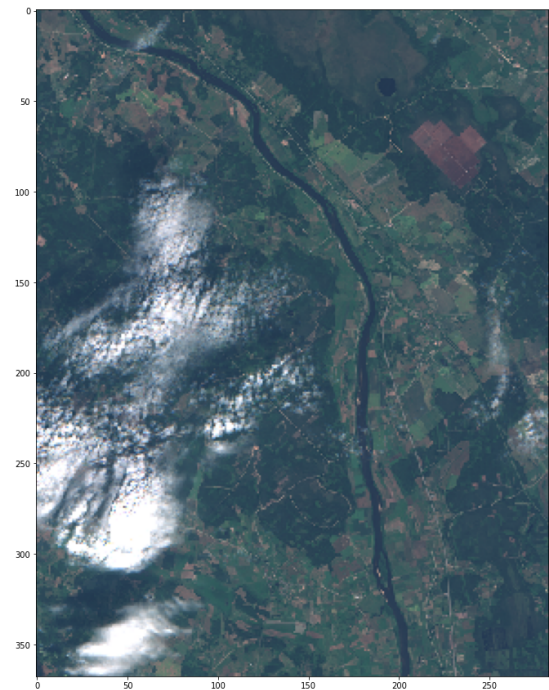
Obrázek 4.6: Sicílie s naší finální maskou

Nejčastějším problémem bylo označování některých polí či stínů jako pixely obsahující mrak na úkor některých okrajů oblačnosti, či velmi jemných mraků. Nicméně se jedná často o zanedbatelné množství pixelů a troufáme si tvrdit, že výsledná maska je velmi dobrá.

Podívejme se nyní ještě na další příklad tvorby naší masky pro snímky bez sněhu na snímku z lotyšského Rēzekne.

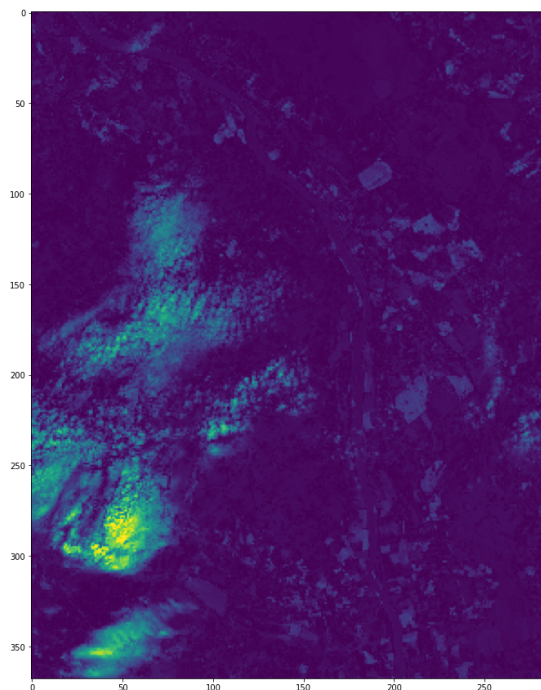


Obrázek 4.7: Řezekne bez oblačnosti

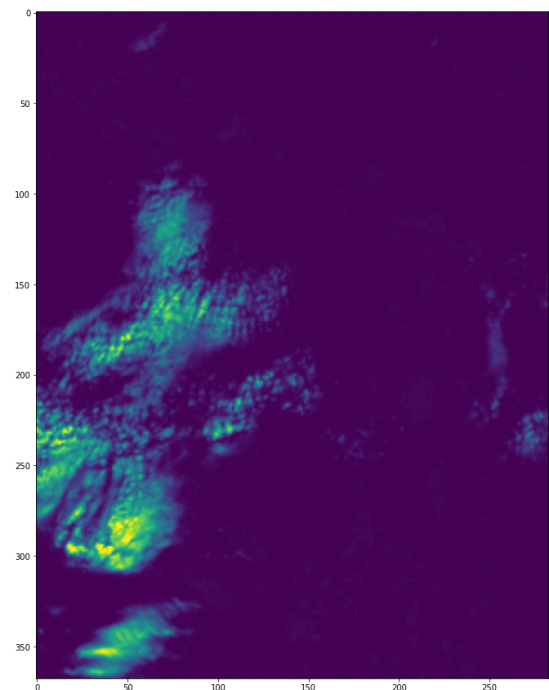


Obrázek 4.8: Řezekne s oblačností

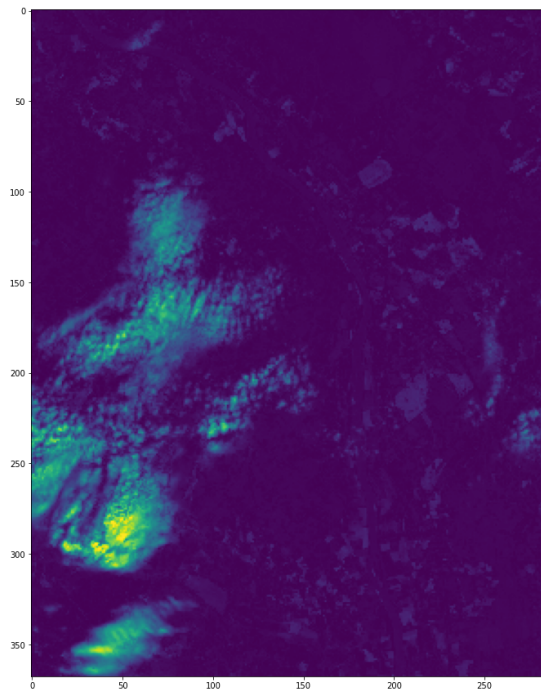
Časová prodleva mezi pořízením těchto snímků je cca 60 dní.



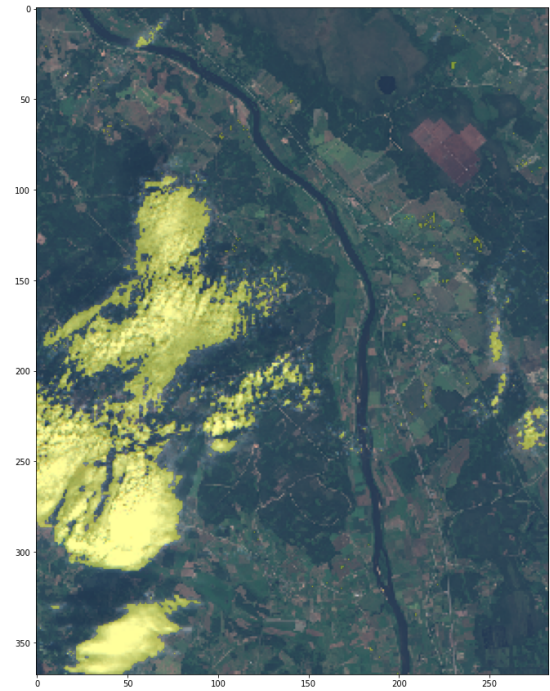
Obrázek 4.9: Řezekne rozdíl B2



Obrázek 4.10: Řezekne rozdíl B10



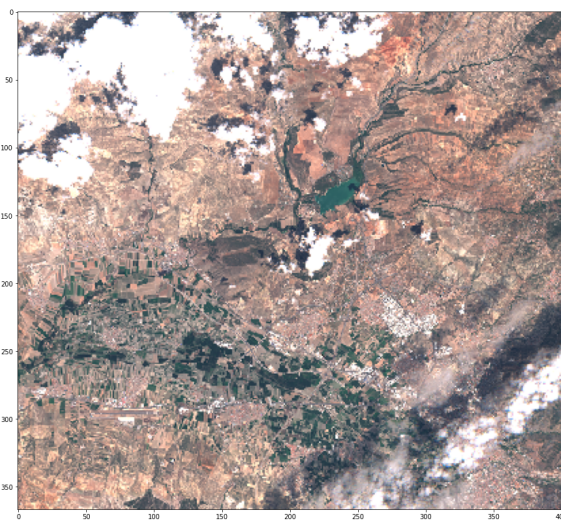
Obrázek 4.11: Rězekne součet rozdílů B2 a B10



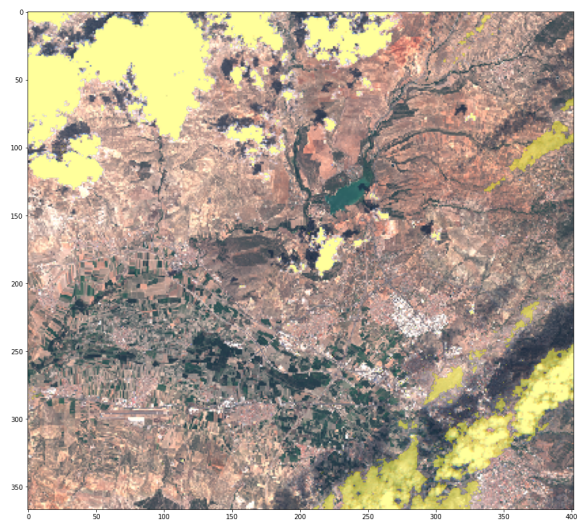
Obrázek 4.12: Rězekne s naší finální maskou oblačnosti

I přes větší časovou prodlevu mezi pořízením snímků jsme tímto naším postupem dokázali vytvořit velice slušnou mrakovou masku.

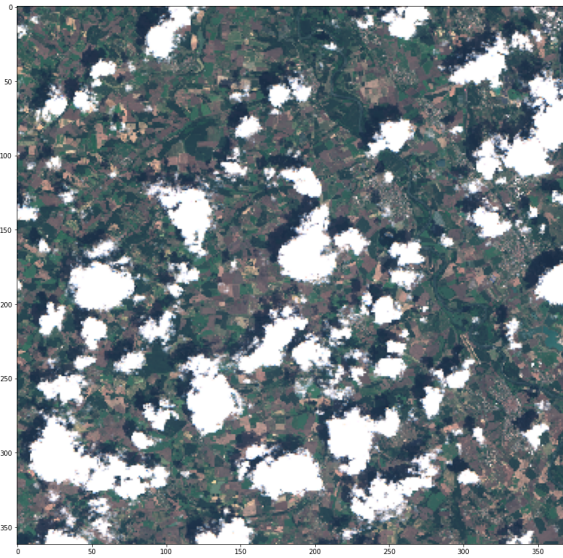
Vytvořili jsme celkem 13 masek, které můžete vidět na následujících obrázcích 4.13 až 4.58



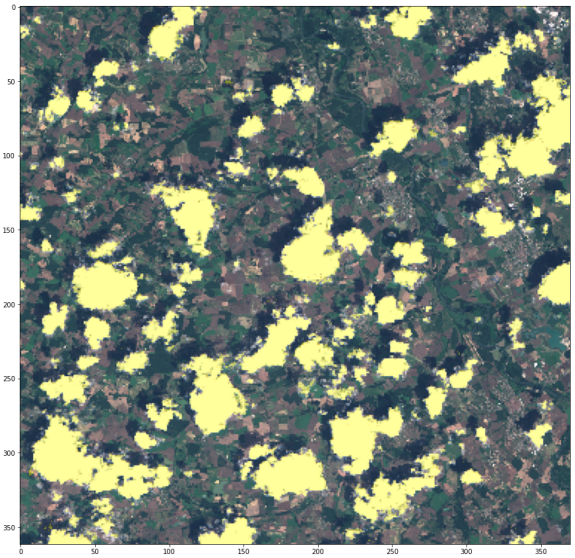
Obrázek 4.13: Španělsko, Granada



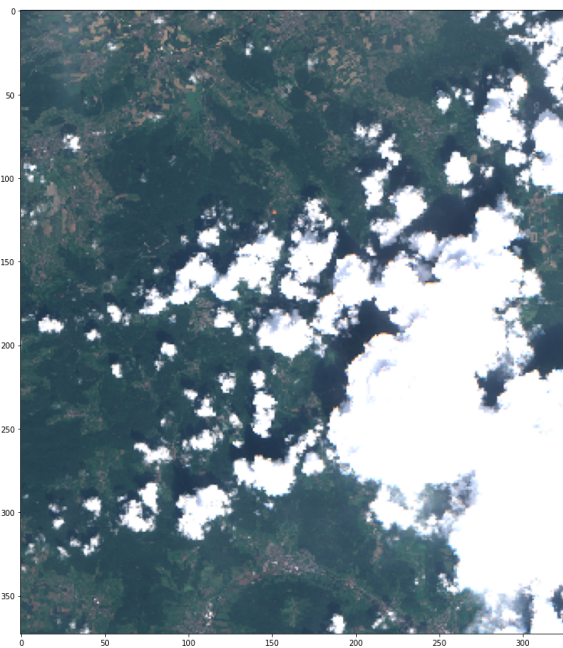
Obrázek 4.14: Španělsko, Granada s naší maskou



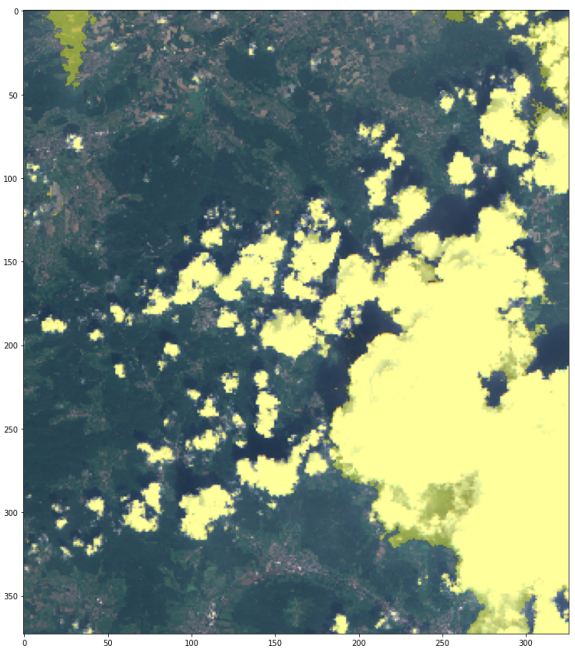
Obrázek 4.15: Francie, Toulouse



Obrázek 4.16: Francie, Toulouse s naší maskou



Obrázek 4.17: Německo, Stuttgart



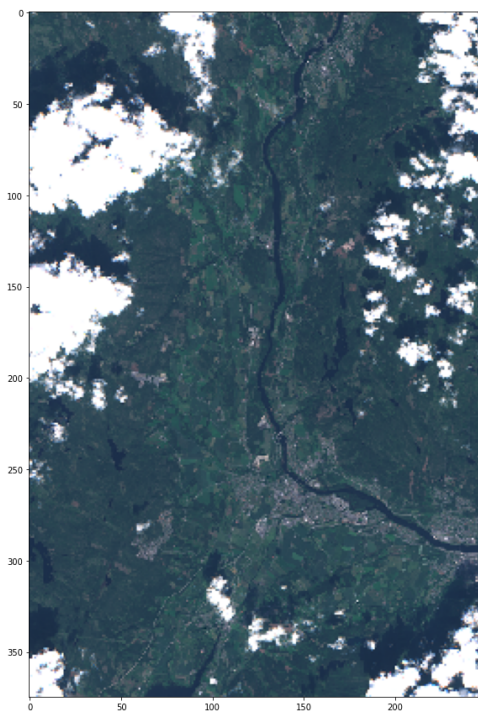
Obrázek 4.18: Německo, Stuttgart s naší maskou



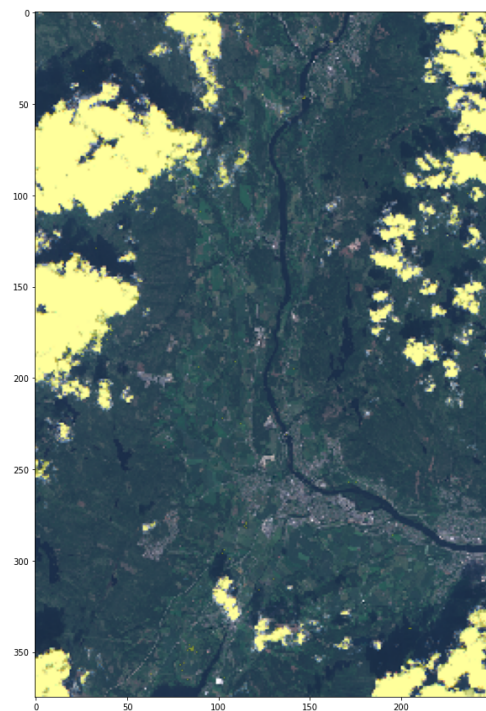
Obrázek 4.19: Německo, Berlín



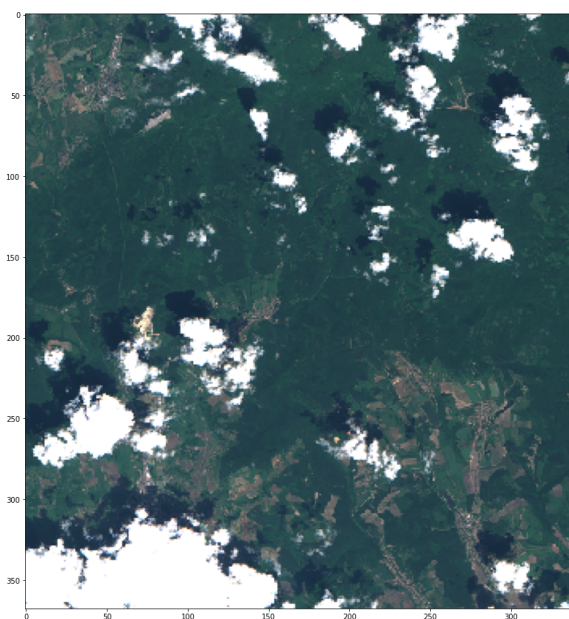
Obrázek 4.20: Německo, Berlín s naší maskou



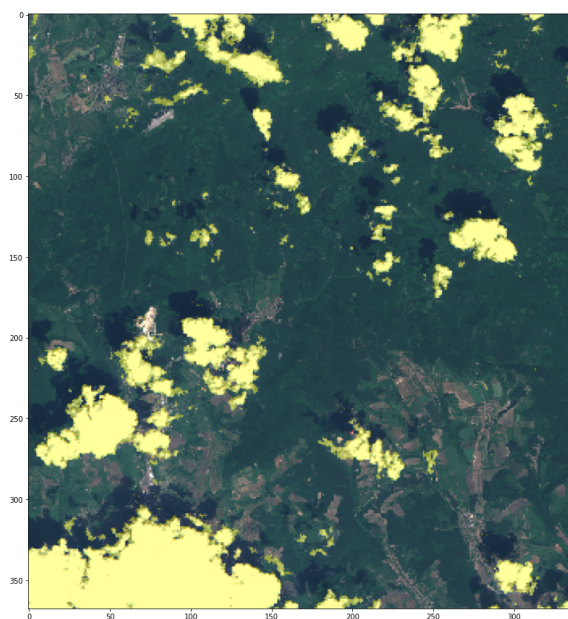
Obrázek 4.21: Norsko, Oslo



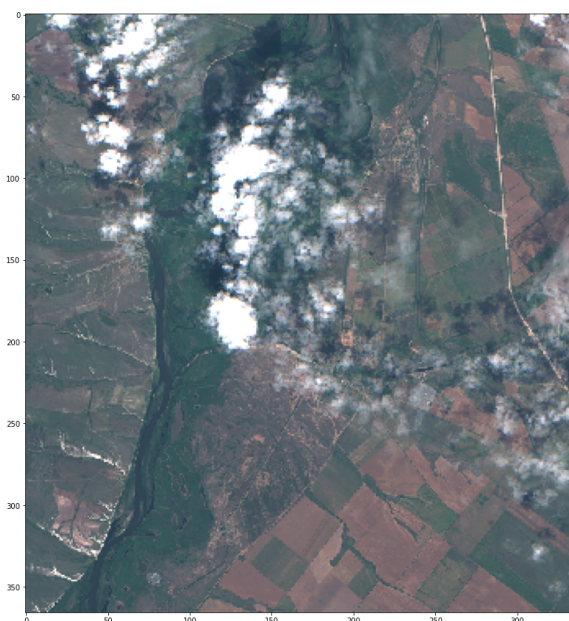
Obrázek 4.22: Norsko, Oslo s naší maskou



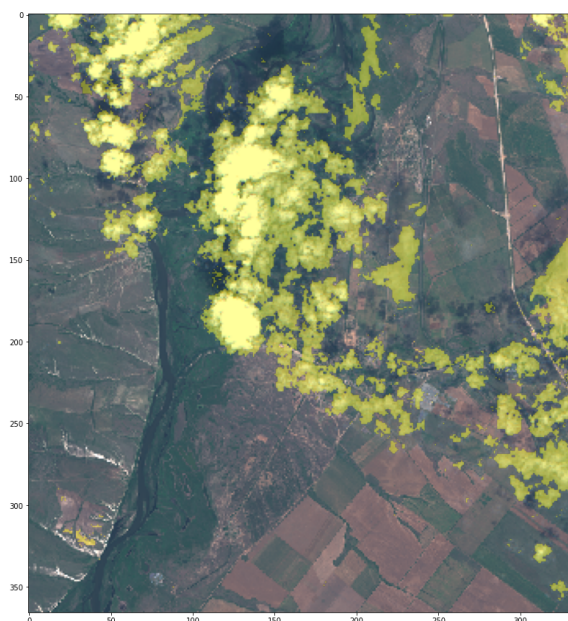
Obrázek 4.23: Maďarsko, Eger



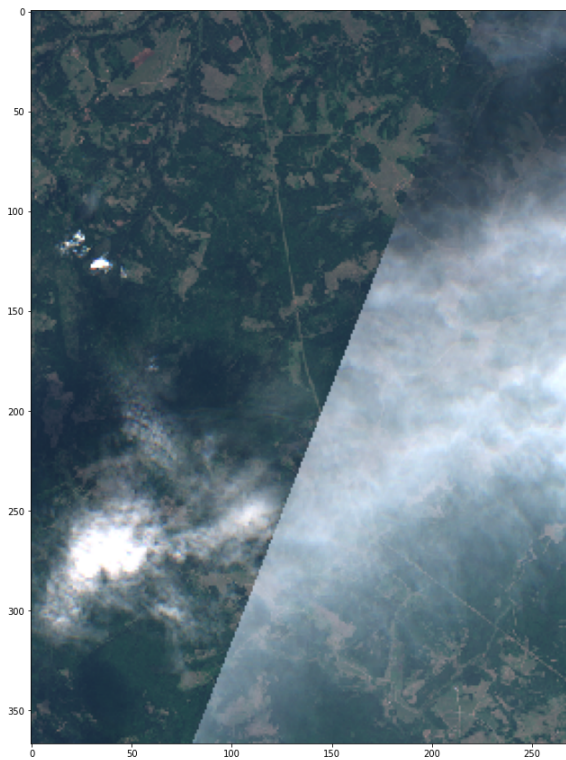
Obrázek 4.24: Maďarsko, Eger s naší maskou



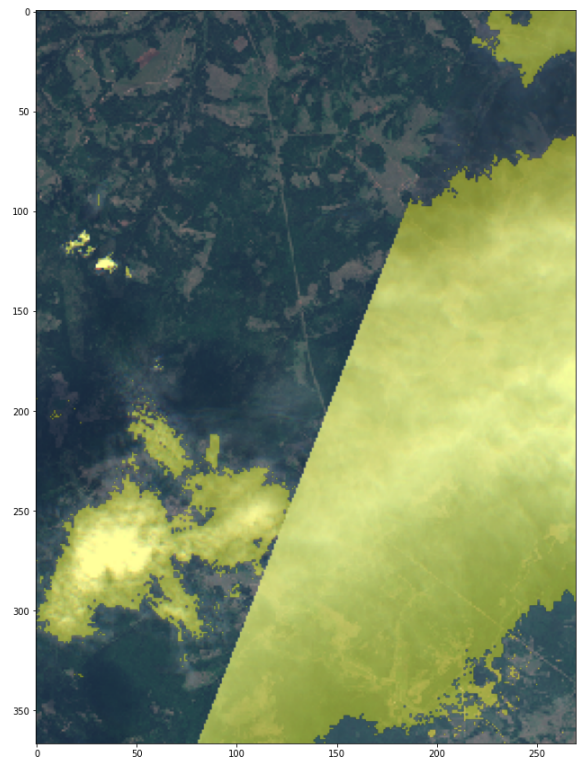
Obrázek 4.25: Rusko, Volgograd



Obrázek 4.26: Rusko, Volgograd s naší maskou



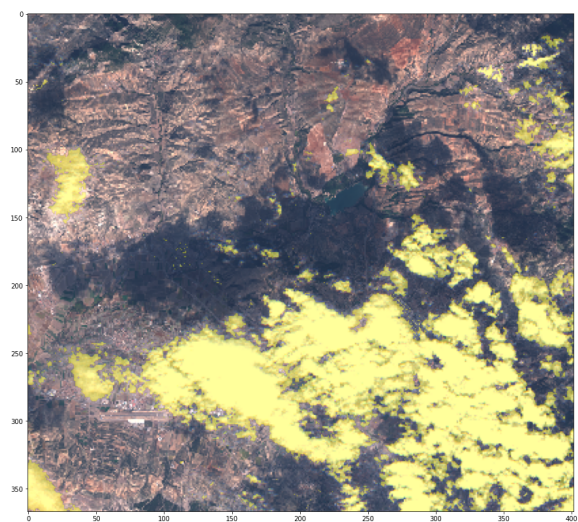
Obrázek 4.27: Rusko, Perm



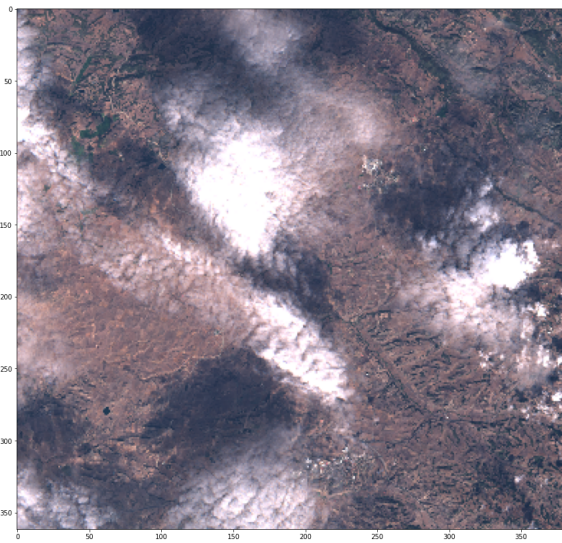
Obrázek 4.28: Rusko, Perm s naší maskou



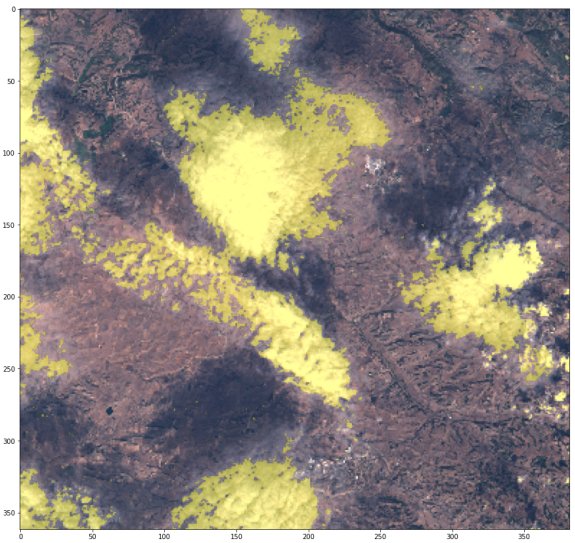
Obrázek 4.29: Španělsko, Granada



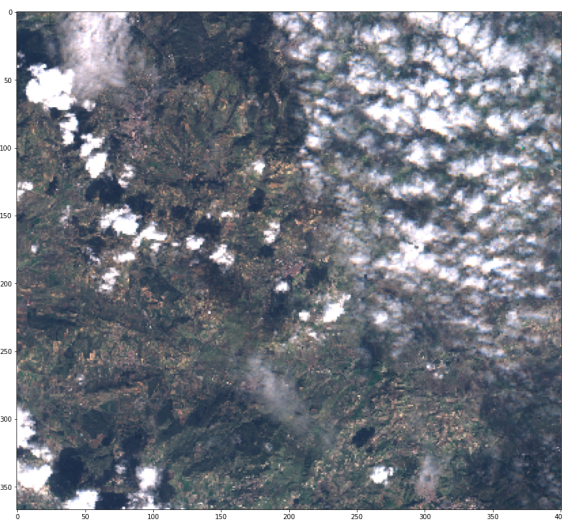
Obrázek 4.30: Španělsko, Granada s naší maskou



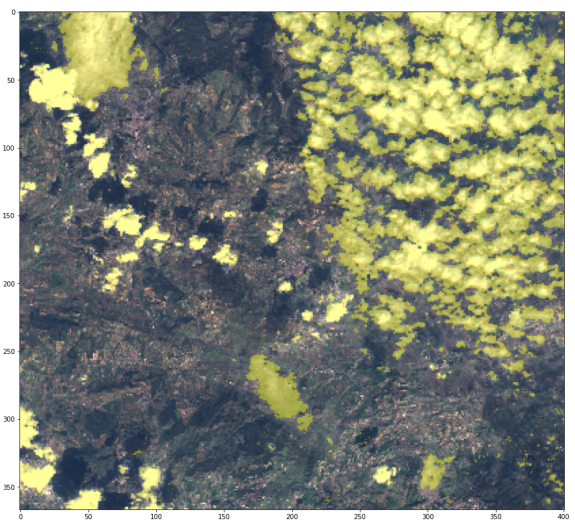
Obrázek 4.31: Španělsko, Barcelona



Obrázek 4.32: Španělsko, Barcelona s naší maskou



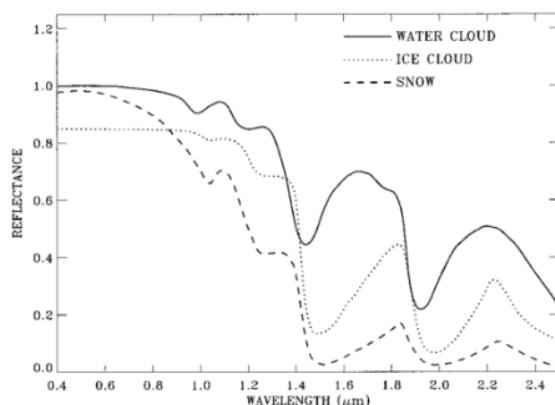
Obrázek 4.33: Itálie, Sicílie



Obrázek 4.34: Itálie, Sicílie s naší maskou

4.1.2 Snímky se sněhem

Jak jsme již zmínili, tak přístup, který poměrně dobře fungoval na snímcích bez sněhové pokrývky pro snímky se sněhem nefungoval. Časové rozdíly doposud fungovali dobře, protože na snímcích nebylo příliš mnoho pixelů, které by měly podobnou odrazivost, jako oblačnost. Sníh a oblačnost mají velice podobnou odrazivost, jak můžeme vidět na obrázku 4.35.

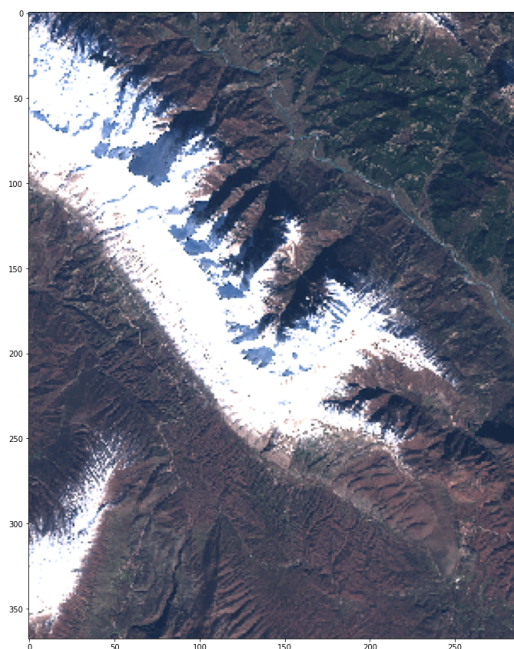


Obrázek 4.35: Odrazivost sněhu a oblačnosti [27]

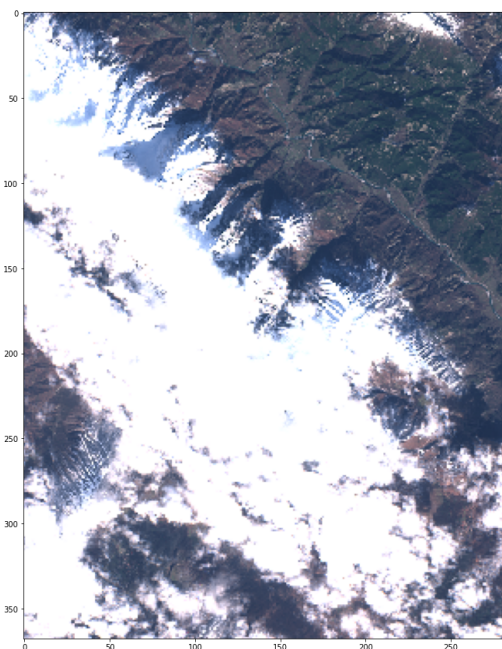
Navíc potřebujeme, aby snímek s oblačností a snímek bez oblačnosti byli pořízeny s co nejmenším časovým rozstupem. Změna tvaru sněhové pokrývky bude mít totiž významný vliv na kvalitu naší masky. Pokud je v jednom místě na jednom ze snímků sníh a na druhém ve stejném místě nikoliv, pak při rozdílu těchto dvou snímků v daném místě získáme významný rozdíl v odrazivostech, který v předchozím případě bez sněhu indikoval přítomnost mraku. Nicméně se nám ne vždy podařilo najít snímky, kde by tento problém nenastal. Tento problém se nám částečně podařilo potlačit přidáním NDSI 2.1.

Nyní demonstrujeme výrobu masky oblačnosti pro snímky, které obsahovali sníh.

Prvním krokem bylo získání dvou snímků dané lokality s, co nejkratším časovým rozstupem.



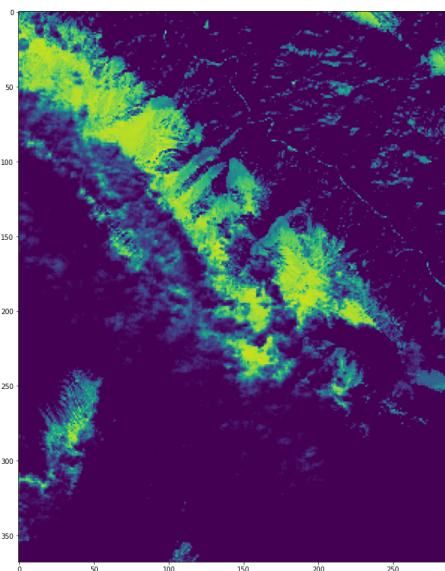
Obrázek 4.36: Řecko, Konitsa bez oblačnosti



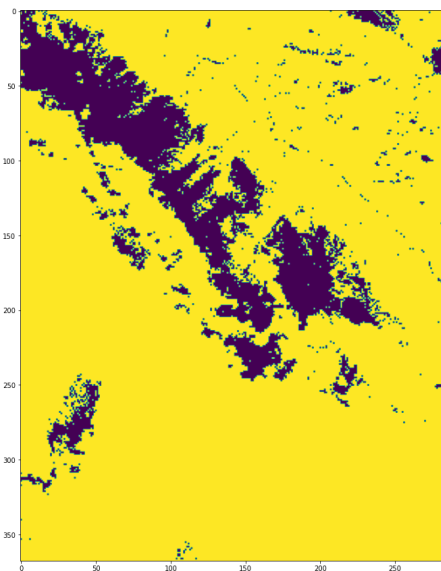
Obrázek 4.37: Řecko, Konitsa s oblačností

Mezi snímky 4.36 a 4.37, je časový rozstup 5 dní. I přes takto krátký rozstup se sněhová pokrývka na několika místech změnila.

Druhým krokem bylo spočtení NDSI na snímku s oblačností a následná prahová filtrace, která je popsána na obrázku 2.2 pod Snow mask branch. Poslední krok této prahové filtrace označený jako Snow boundaries jsme vynechali. Došli jsme k sněhové masce, která je vidět na obrázku 4.39.

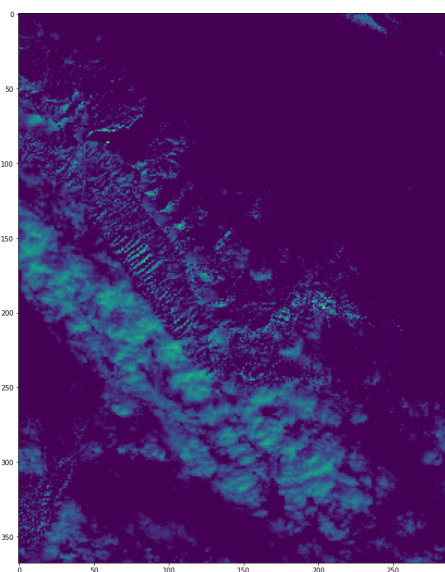


Obrázek 4.38: Řecko, Konitsa NDSI

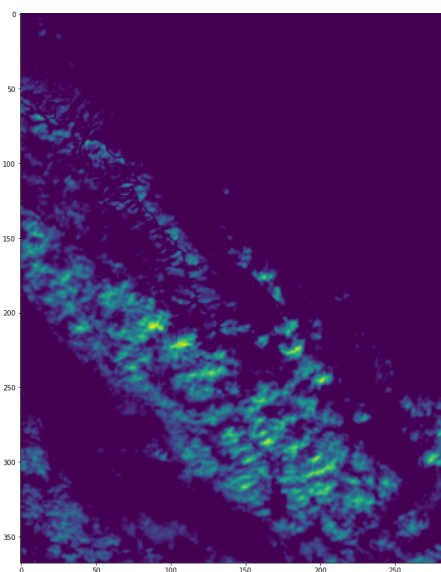


Obrázek 4.39: Řecko, Konitsa NDSI po prahové filtraci

Dalším krokem bylo odečtení obou snímků od sebe pro jednotlivé pixely pro pásmo B10 (jeden z nich je přenásoben konstantou K 4.5) a následná normalizace tohoto rozdílu. To samé jsme provedli pro pásmo B2, avšak ne u všech snímků. Pro některé se ukázalo lepší použít pouze pásmo B10. Nicméně pro tento konkrétní případ používáme obě pásma. Výsledek vidíme na obrázku 4.40 a 4.41.

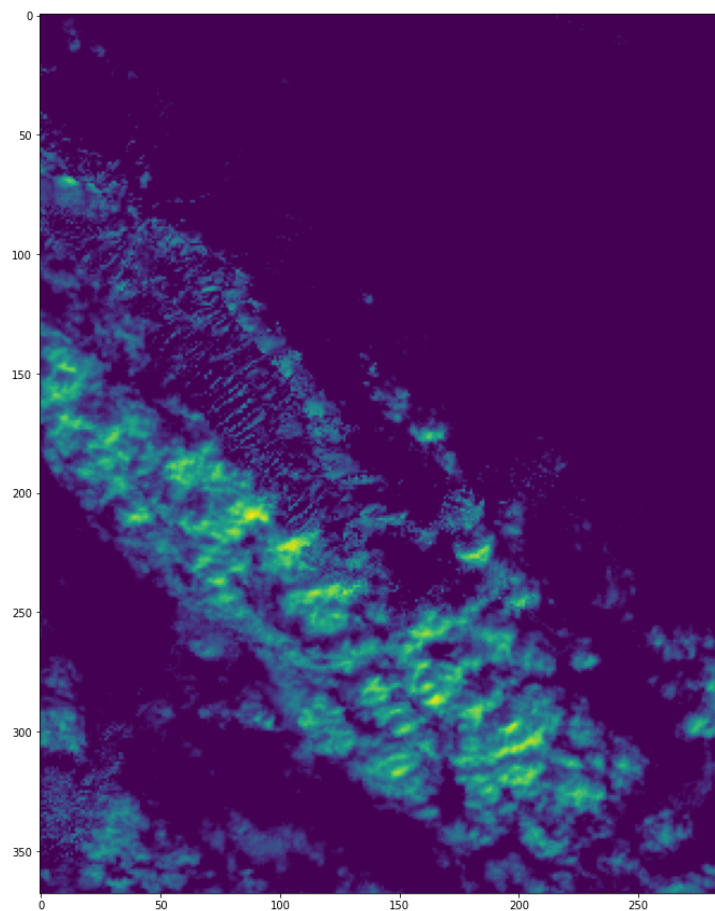


Obrázek 4.40: Řecko, Konitsa rozdíl B2



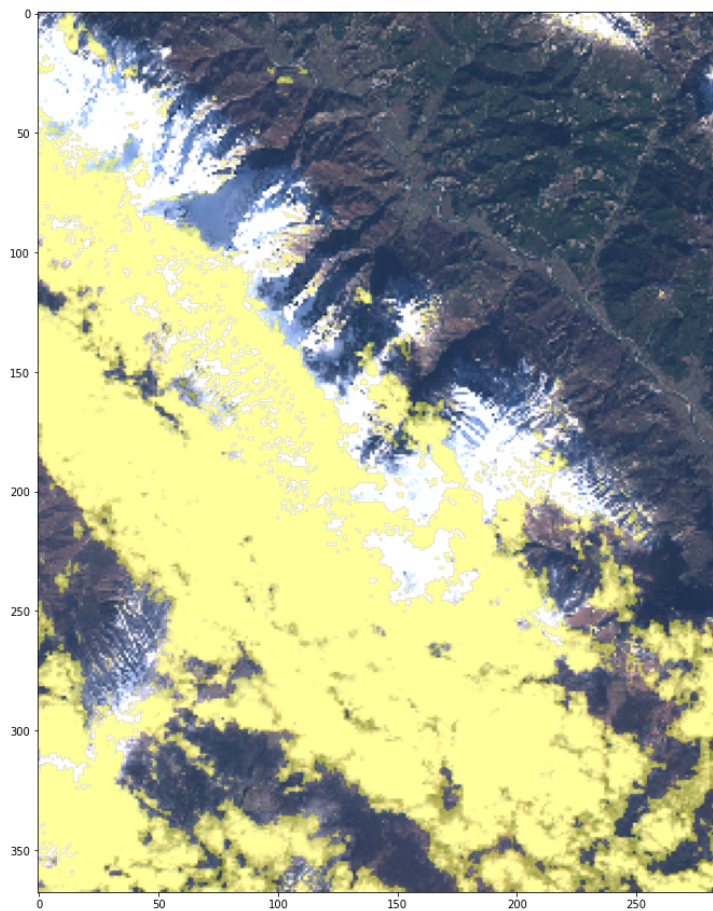
Obrázek 4.41: Řecko, Konitsa rozdíl B10

Na rozdílů B2 4.40 vidíme i nějaké sněhové plochy. Čtvrtým krokem bude sečtení hodnot NDSI po prahové filtraci 4.39, hodnot rozdílů pásma B2 4.40 a hodnot rozdílů pásma B10 4.41. Sečtením těchto hodnot se pokusíme potlačit jejich jednotlivé nedostatky. Očekáváme, že přičtením indexu NDSI částečně potlačíme pixely, které obsahují sníh a neobsahují oblačnost. Výsledek tohoto součtu můžeme vidět vizualizovaný na obrázku 4.5.



Obrázek 4.42: Řecko, Konitsa součet rozdílů pásem B2 a B10 a NDSI po prahové filtraci

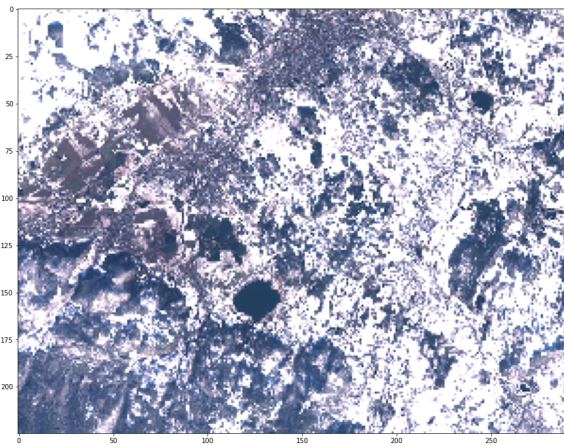
Následným prahováním výsledných hodnot 4.42 získáme naši finální masku. Můžeme ji vidět na obrázku 4.43. Při prahování jsme postupovali stejným způsobem, jako pro snímky bez sněhu.



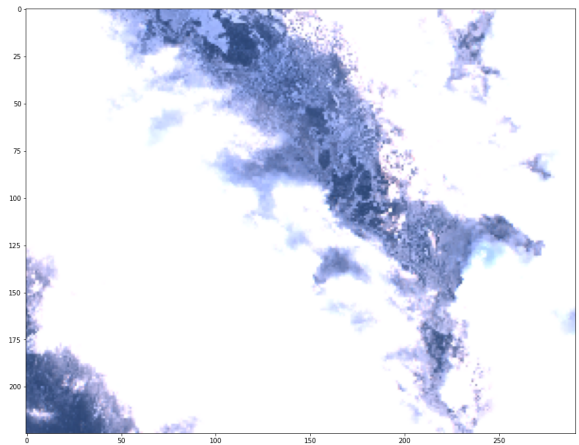
Obrázek 4.43: Řecko, Konitsa s naší finální maskou oblačnosti

Pro snímky bez sněhu jsme získávali pěkné masky u kterých dalo práci nalézt nedokonalosti, bohužel pro snímky se sněhem je tento úkol daleko jednodušší. Na mnoha místech vidíme, že sněhové pixely byly označeny jako oblačnost. Bohužel se nám nepodařilo získat lepší masky a proto budeme pracovat s těmito.

Podívejme se nyní na další příklad tvorby naší masky pro snímky se sněhem, který využívá pouze rozdílů pásma B10 v lokalitě polské Jelenie Góry.

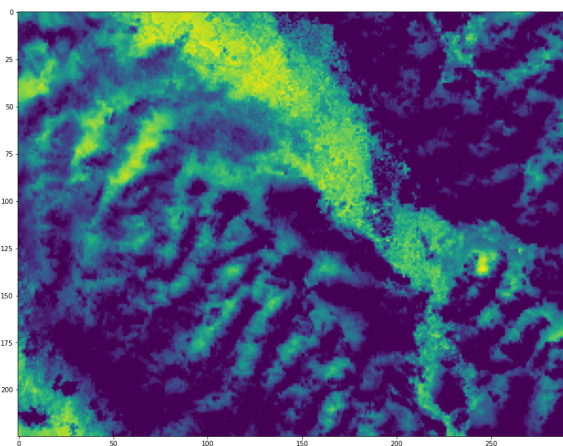


Obrázek 4.44: Polsko, Jelenia Góra bez oblačnosti

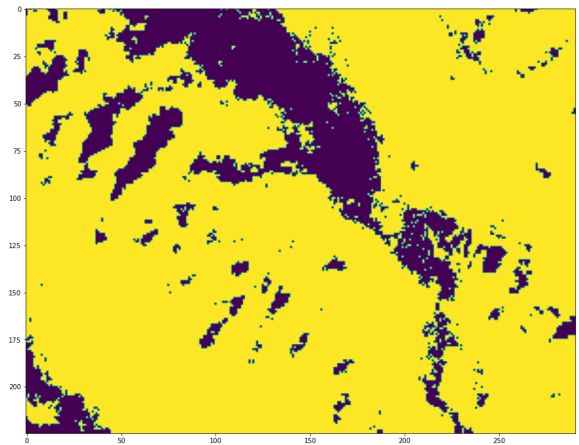


Obrázek 4.45: Polsko, Jelenia Góra s oblačností

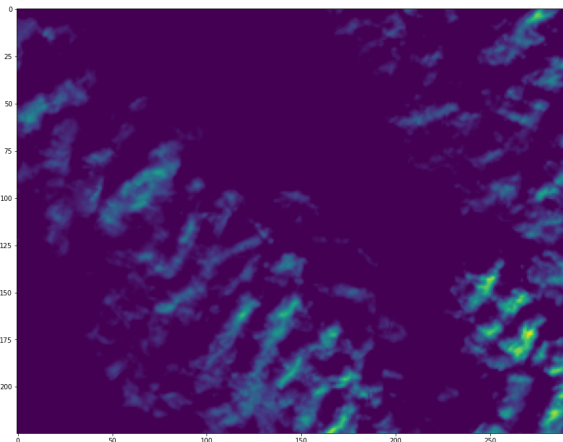
Časová prodleva mezi pořízením těchto snímků je cca 20 dní.



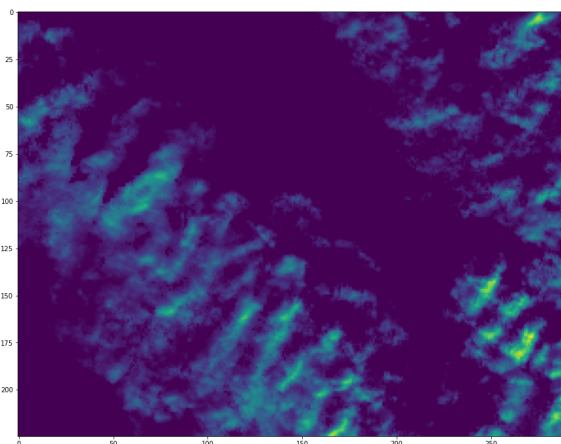
Obrázek 4.46: Polsko, Jelenia Góra NDSI



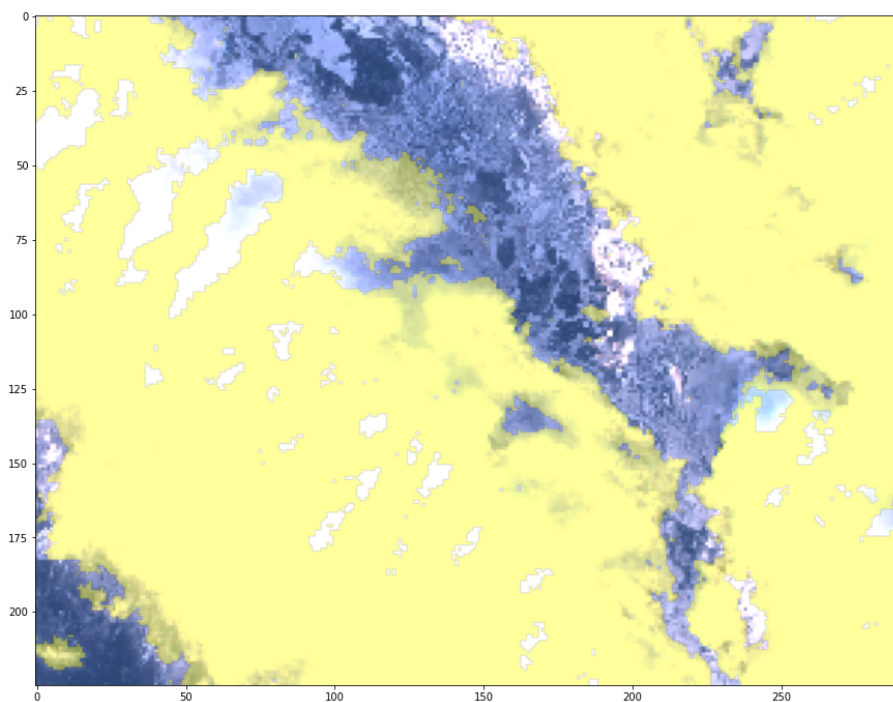
Obrázek 4.47: Polsko, Jelenia Góra NDSI po prahové filtraci



Obrázek 4.48: Polsko, Jelenia Góra rozdíl B10



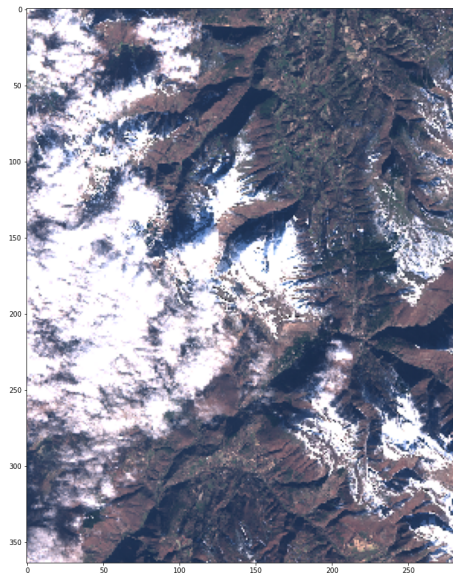
Obrázek 4.49: Polsko, Jelenia Góra součet rozdílů B10 a NDSI po prahové filtraci



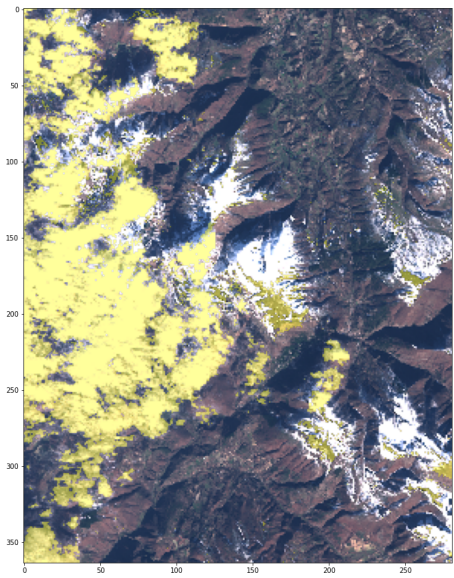
Obrázek 4.50: Polsko, Jelenia Góra s naší finální maskou

Pouhým okem najdeme spoustu míst, kde je námi vytvořená maska nevyhovující, ale je to ta nejlepší které jsme byli schopni dosáhnout.

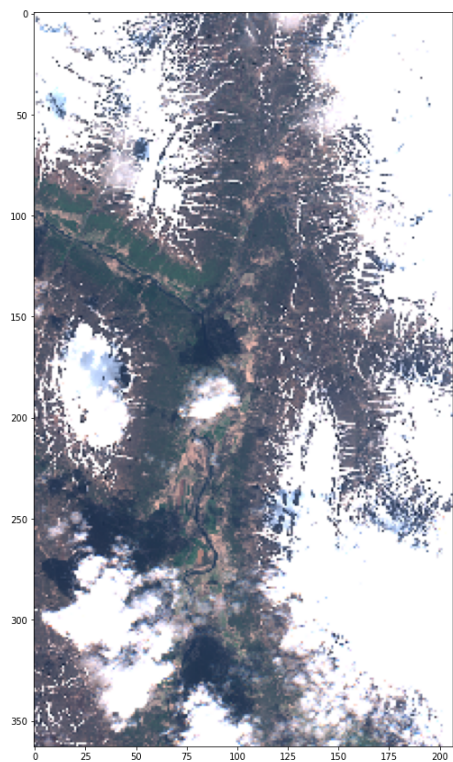
Vytvořili jsme celkem 6 masek pro snímky s oblačností. Pojd' me se nyní podívat na zbylé 4.



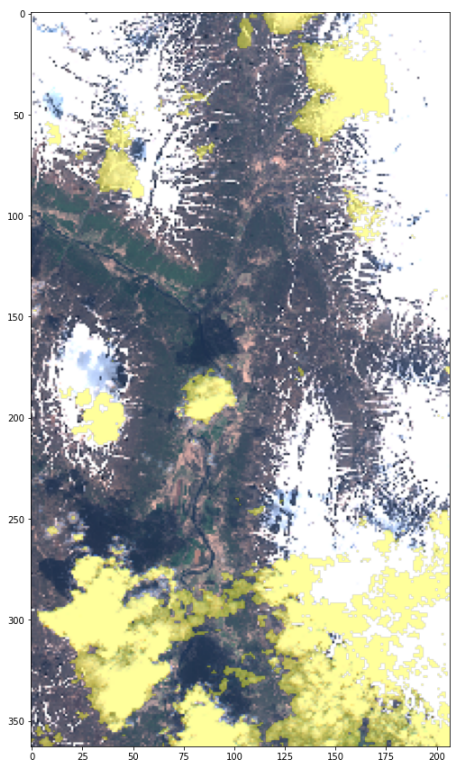
Obrázek 4.51: Itálie, Vary



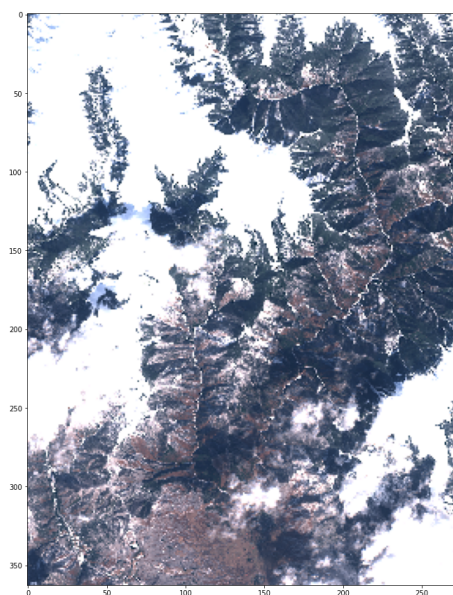
Obrázek 4.52: Itálie, Vary s naší maskou



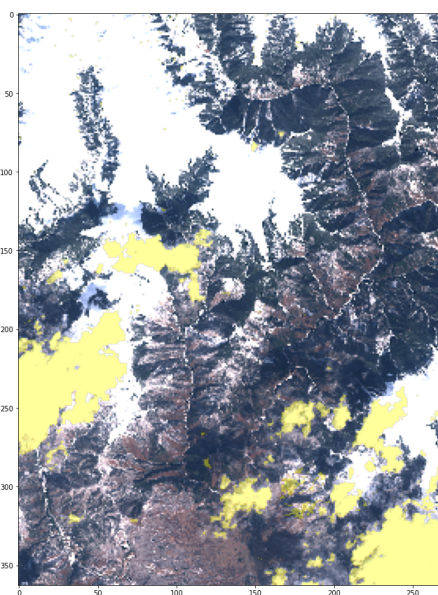
Obrázek 4.53: Island



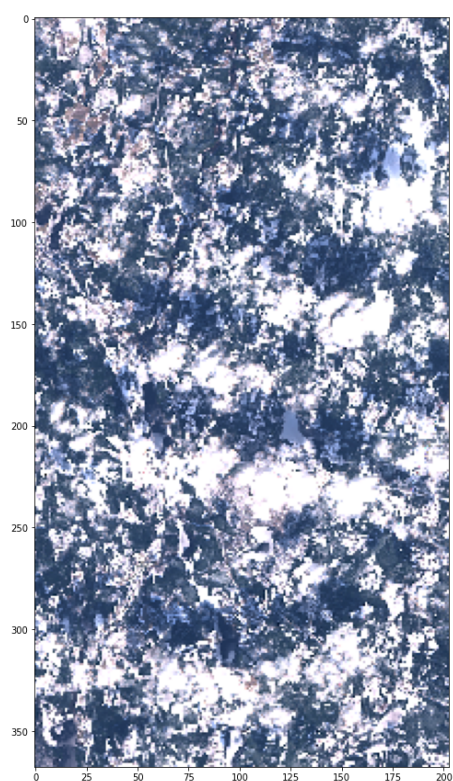
Obrázek 4.54: Island s naší maskou



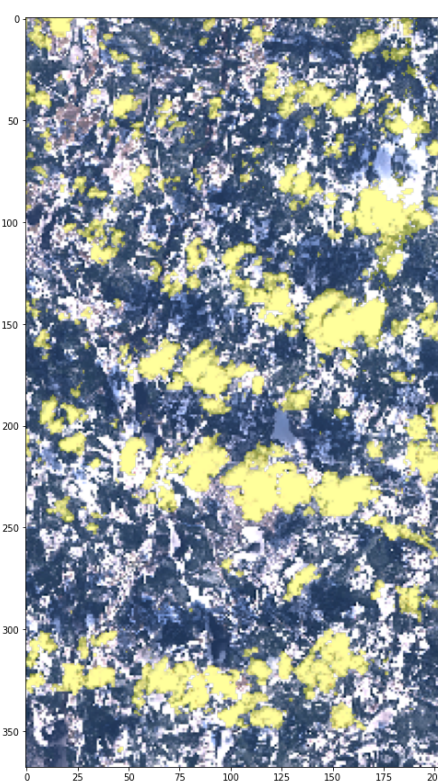
Obrázek 4.55: Rumunsko, Slatina



Obrázek 4.56: Rumunsko, Slatina s naší maskou



Obrázek 4.57: Švédsko, Mullsjö



Obrázek 4.58: Švédsko, Mullsjö s naší maskou

Snímky jsme opatřili pomocí <https://www.sentinel-hub.com/>. Sentinel-hub nám po specifikaci místa vyhledávání a časového intervalu umožňuje stáhnout potřebné snímky. Umožňuje také vyhledávat

podle procentuálního zastoupení oblačnosti v snímku. Nyní je navíc možné stáhnout mrakovou masku vygenerovanou algoritmem s2cloudless společně se snímky.

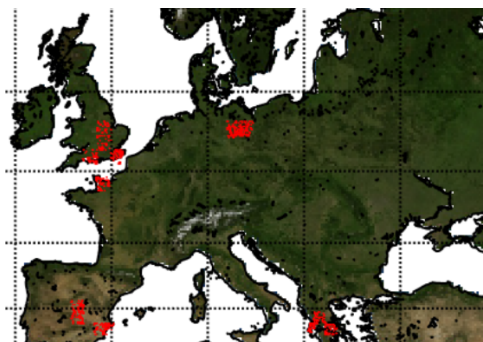
Doposud jsme demonstrovali tvorbu našich masek a ukázali výsledné masky. Čtenář se sám může přesvědčit, že při získávání trénovacího datasetu jsme se snažili obsáhnout všechny možné biomy nacházející se v Evropě. Také jsme se snažili obsáhnout všechny různé druhy oblačnosti či snůh.

4.2 Další použitá data

Poté co budeme mít natrénovaný vlastní model budeme potřebovat ověřit, zda opravdu funguje. Ideální by byl reálný dataset označený člověkem, který je dostatečně velký s co největším zastoupením různých biomů. Z povahy věci je takovýchto datasetů velice málo. Nicméně se nám podařilo najít dataset, který se svými spolupracovníky vytvořil André Hollstein.

4.2.1 Single-pixel dataset vytvořený Hollsteinem a spol.

Tento dataset se skládá z 5 647 725 ručně označených pixelů rozprostřených po celé Zeměkouli [5]. Tato data budou ideální pro validaci výsledného algoritmu. Nevýhodou tohoto datasetu je, že obsahuje pouze jednotlivé pixely a my si tak nebudeme moci výsledek našeho algoritmu zobrazit a opticky srovnat. Budeme také používat pouze data z Evropy, protože náš model budeme trénovat pouze na datech z Evropy. Na obrázku 4.59 můžeme vidět z jakých částí Evropy data pocházejí.



Obrázek 4.59: distribuce datasetu vytvořeného Hollsteinem a spol.

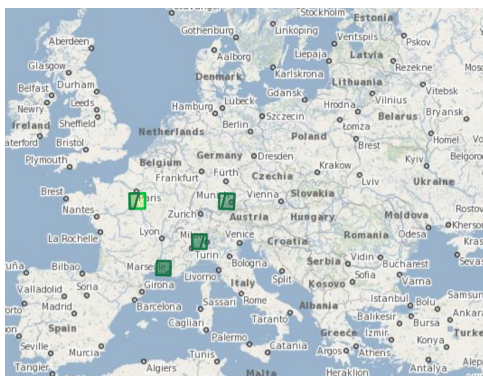
Dalším datasetem, který se nám podařilo najít byl vytvořen Louisem Baetensem a jeho spolupracovníky.

4.2.2 Dataset vytvořený Baetensem a spol.

Baetens a spol. vytvořili software nazvaný Active Learning Cloud Detection (ALCD), který byl vyvinut pro vytváření reference cloud mask. Metody aktivního učení s učitelem umožňují snížit počet nutných vzorků pro trénování iterativním výběrem těch, u kterých je důvěra klasifikačního algoritmu v předchozích iteracích nízká. ALCD metoda byla navržena tak, aby minimalizovala čas operátora díky ručně řízené metodě aktivního učení s učitelem. Natrénovaný klasifikátor používá kombinaci spektrálních a multi-temporálních informací jako vstupní data a produkuje plně klasifikované obrázky. ALCD metoda byla ověřena pomocí vizuálních kritérií, konzistenčních zkoušek a porovnána s jinými ručně generovanými maskami oblačnosti vytvořenými Hollsteinem a spol. s celkovou přesností nad 98 %. ALCD bylo použito pro vytvoření 32 referenčních masek oblačnosti na 10 různých místech s různými ročními

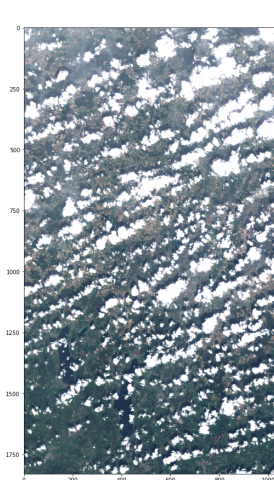
obdobími a různými druhy oblačnosti. Tyto masky použili Baetens a spol. k ověření masek oblačnosti generovanými modely MAJA, Sen2Cor a Fmask. Výsledky ukazují, že MAJA a FMask se chovají podobně s celkovou přesností kolem 90 % (91 % pro MAJA, 90 % pro FMask), zatímco celková přesnost Sen2Cor je 84 % [16].

Z datasetu vytvořeného Baetensem a spol. jsme použili snímky Evropy. Na obrázku 4.60 můžeme vidět z jakých částí Evropy data pocházejí.

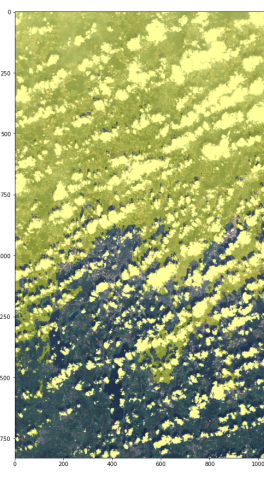


Obrázek 4.60: distribuce datasetu vytvořeného Baetensem a spol.

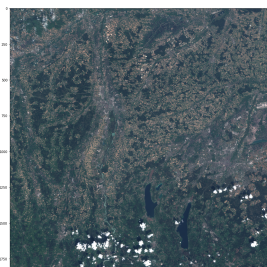
Jednotlivé snímky pak můžeme vidět na obrázcích 4.64 až 4.65.



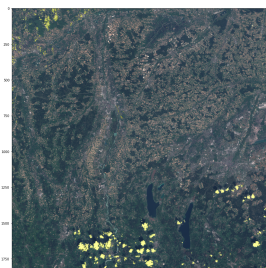
(a) Německo, Mnichov



(b) Německo, Mnichov s maskou

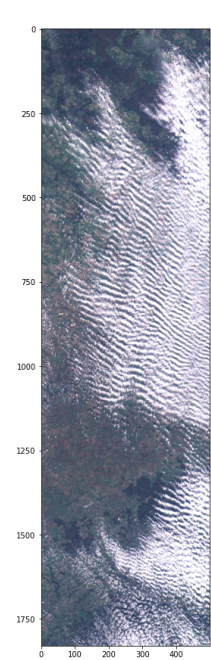


(c) Německo, Mnichov

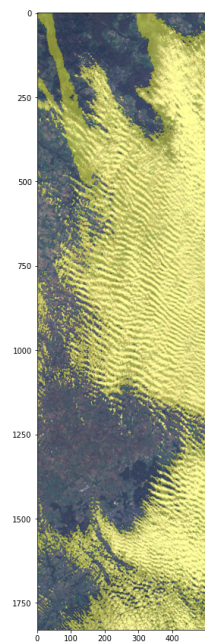


(d) Německo, Mnichov s maskou

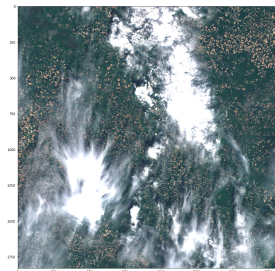
Obrázek 4.61: Masky vytvořené pomocí ALCD



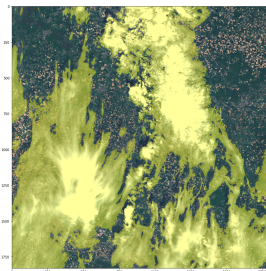
(a) Francie, Orleans



(b) Francie, Orleans s maskou

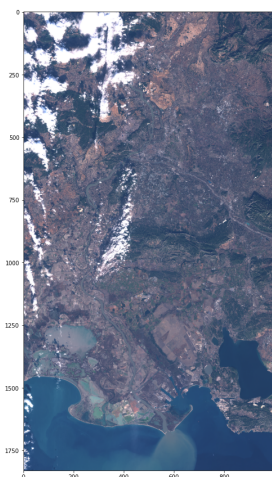


(c) Francie, Orleans

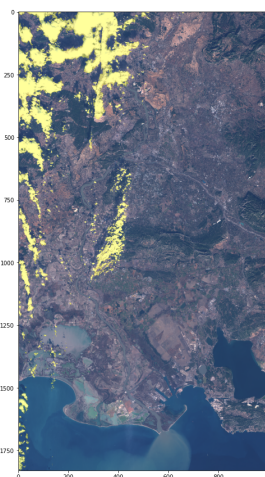


(d) Francie, Orleans s maskou

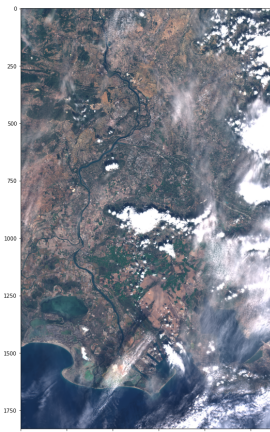
Obrázek 4.62: Masky vytvořené pomocí ALCD



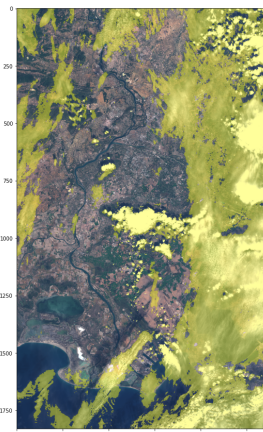
(a) Francie, Arles



(b) Francie, Arles s maskou

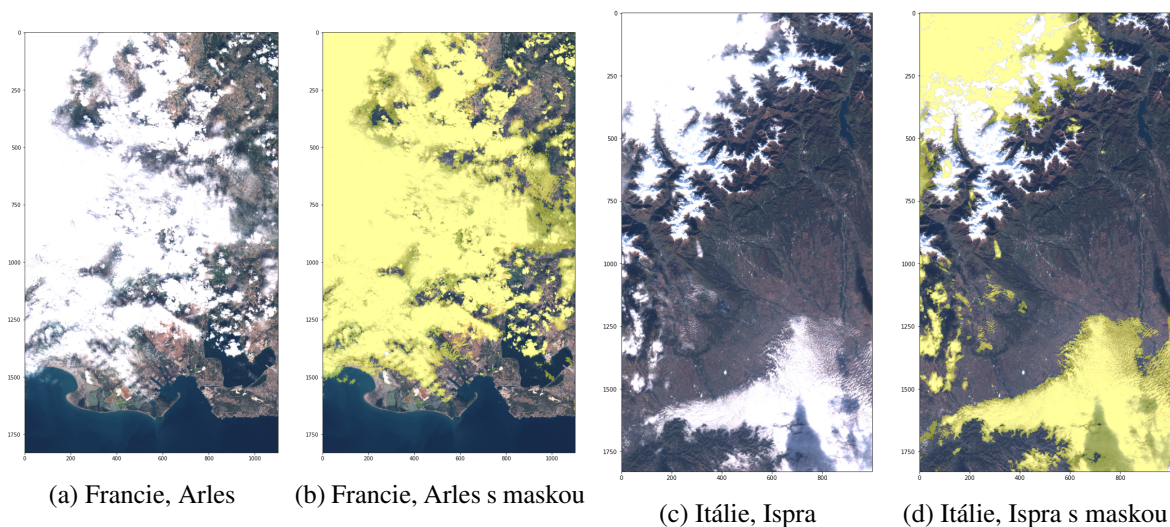


(c) Francie, Arles

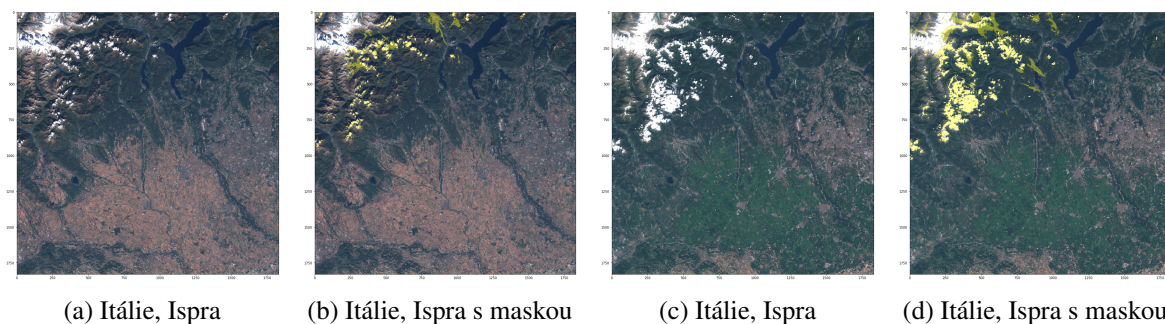


(d) Francie, Arles s maskou

Obrázek 4.63: Masky vytvořené pomocí ALCD



Obrázek 4.64: Masky vytvořené pomocí ALCD

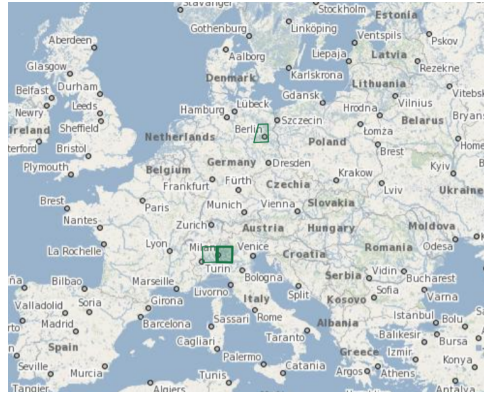


Obrázek 4.65: Masky vytvořené pomocí ALCD

Čtenář se může přesvědčit, že tyto snímky obsahují všechny druhy oblačnosti a i sněhovou pokrývku, která je na rozdíl od námi vytvořeného datasetu velice dobře označená. Protože nejsme plně spokojeni s podobou našeho datasetu, pokusíme se použít i dataset vytvořený Baetensem k natrénování našeho algoritmu. Možnou nevýhodou, je že se jedná o snímky z pouze 4 evropských lokací.

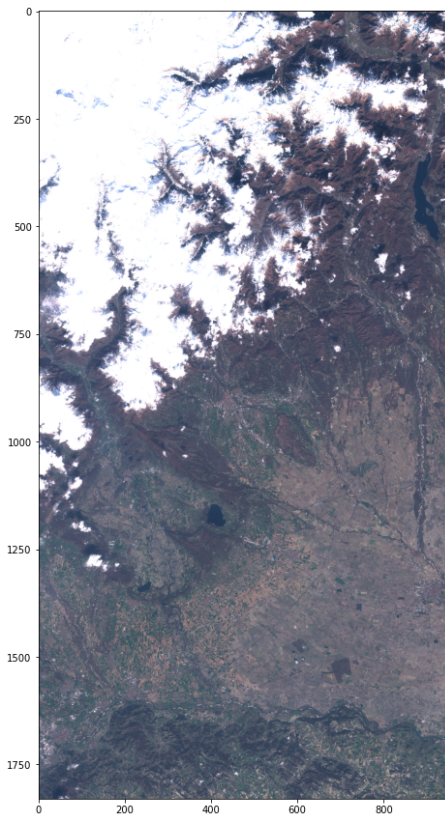
4.2.3 Validační snímky vytvořené Hollsteinem a spol.

Kromě masek vytvořených pomocí ALCD získali i masky celých snímků vytvořených a ručně označených Hollsteinem a spol. na kterých Baetens a spol. ALCD validovali. Tyto ručně označené snímky budou ideální k validaci a posouzení funkčnosti našeho modelu. Jedná se o jiná data než ta jejichž distribuci můžeme vidět na obrázku 4.59. Distribuci těchto snímků můžeme vidět na obrázku 4.66.

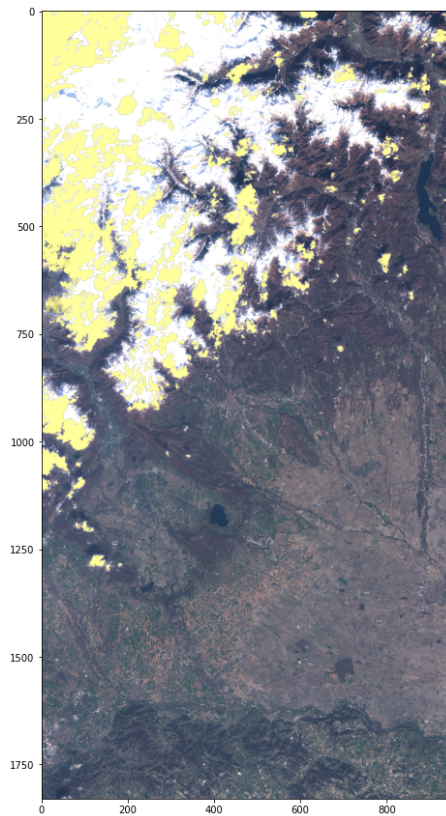


Obrázek 4.66: lokace snímků označených Hollsteinem a spol.

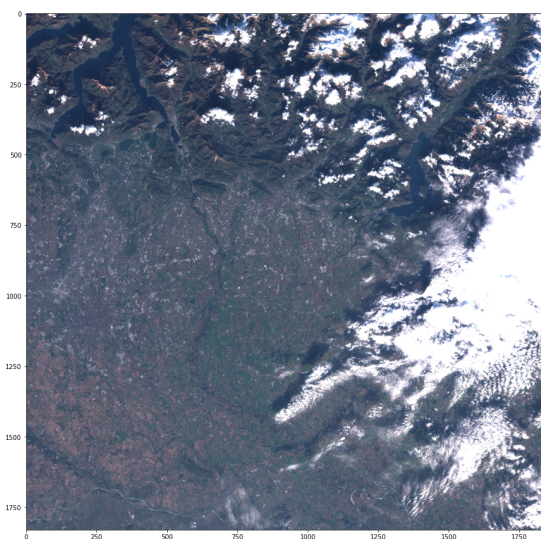
Snímky můžeme vidět na obrázcích 4.67 až 4.72.



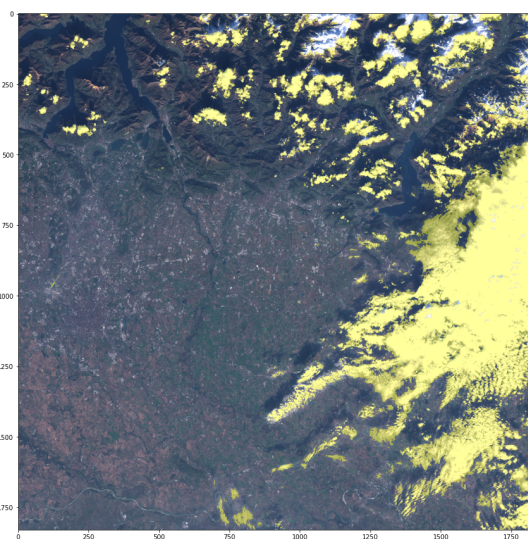
Obrázek 4.67: Itálie, Ispra



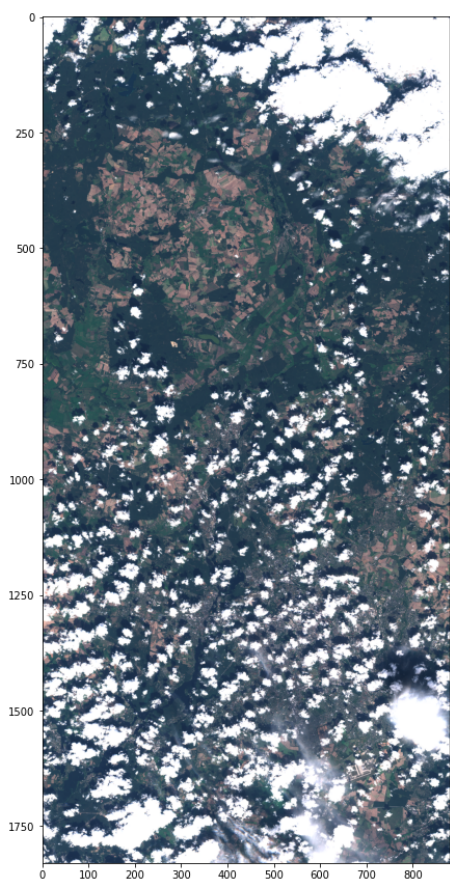
Obrázek 4.68: Itálie, Ispra s maskou



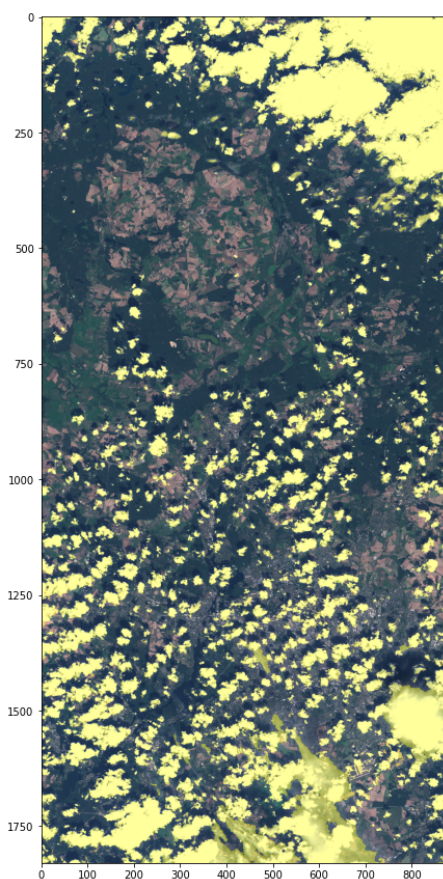
Obrázek 4.69: Itálie, Bergamo



Obrázek 4.70: Itálie, Bergamo s maskou



Obrázek 4.71: Německo, Berlín



Obrázek 4.72: Německo, Berlín s maskou

Kapitola 5

Srovnání úspěšnosti variant algoritmů Náhodných lesů

Cílem naší práce je navrhnout vlastní algoritmus, který bude detekovat oblačnost a srovnat jeho výsledky s nejlepšími dostupnými algoritmy, které se v současnosti používají. Budeme využívat 2 přístupy. Prvním bude detekce oblačnosti na základě informace z jednoho pixelu a druhým bude detekce oblačnosti s využitím závislosti mezi sousedními pixely. To znamená, že budeme detekovat oblačnost v jednom pixelu za použití informace z jednoho pixelu a jeho 8 sousedních pixelů. Tyto dva přístupy budeme nadále označovat jako single-pixel model a multi-pixel.

5.1 Single-pixel náhodné lesy

Náhodné lesy jsou velice oblíbeným algoritmem strojového učení. Tento algoritmus můžeme nalézt v knihovně Scikit-learn pro Python. Kompletní dokumentaci lze najít na stránce: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

Tento algoritmus má verzi regresoru jehož výsledkem jsou spojitá data i verzi klasifikátoru jehož výsledkem jsou binární data. My budeme pracovat pouze s klasifikátorem. Tomuto klasifikátoru musíme dodat dvoje vstupní data. Jednak binární vektor y , který nám sděluje, zda je v daném místě mrak ($y_i = 1$) či nikoliv ($y_i = 0$). Ten může vypadat například takto:

$$y = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \quad y \in \mathbf{R}^n. \quad (5.1)$$

Tento vektor vyrobíme z námi vytvořených masek oblačnosti. Jednotlivé hodnoty tedy korespondují s konkrétním pixelem v obrázku. Těchto pixelů je n .

Druhým druhem dat, který dodáme klasifikátoru Náhodné lesy je tzv. prediktor M (feature vector). Ten může vypadat následovně:

$$M_{n,p} = [a \quad b \quad \cdots \quad p], \quad a, b, \dots, z \in \mathbf{R}^n. \quad (5.2)$$

a, b, \dots, p jsou vektory o n prvcích, které představují jednotlivé pásma. Těchto vektorů je p . Například vektor a může obsahovat data z pásma B1 atd. Počet pásem, které budeme používat, bude 10 a budou to konkrétně pásma B01, B02, B04, B05, B08, B8A, B09, B10, B11, B12. Stejná pásma využívá algoritmus

s2cloudless a to je důvod proč je budeme používat i my. MSI Sentinelu-2 nám poskytuje 13 spektrálních pásem a počet všech kombinací různé délky, které bychom museli vyzkoušet abychom zjistili, které fungují nejlépe je 8191. A to nebereme do úvahy všechny možné poměry jednotlivých pásem, které bychom mohli také prozkoumat. Abychom tedy ušetřili velké množství času rozhodli jsme využít stejná pásma jako právě s2cloudless, který je součástí sentinel-hubu. Obecně počet pásem budeme značit p , nicméně $p = 10$ bude pro single-pixel model platit vždy.

V dokumentaci lze najít, že Náhodné lesy mají vícero různých parametrů, které můžeme manuálně nastavit či ponechat jejich výchozí nastavení. Pro nás budou důležité parametry `n_estimators`, kterým stanovujeme počet stromů v náhodném lesu a `max_depth`, který stanovuje maximální hloubku stromu.

Časová náročnost tohoto algoritmus roste s počtem stromů v lesu, zvyšující se maximální hloubkou, počtem pásem p , ale i s velikostí trénovací množiny. Budeme muset tedy dělat kompromis mezi množstvím informací, které náhodným lesům dodáme a mezi výpočetním časem.

Abychom našli ideální kombinaci parametrů `n_estimators` a `max_depth` provedeme hledání v síti (gridsearch). K tomu nám poslouží funkce `GridSearchCV`, která lze opět nalézt v knihovně Scikit-learn pro python na stránce: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

V předchozí kapitole jsme si představili různé datasety, které jsme vytvořili nebo převzali z jiných prací. Nyní si pojďme říct na kterých budeme trénovat náš model. Zkusíme model natrénovat na různých kombinacích těchto datasetů. Budou to následující kombinace společně s jejich označeními, které budeme nadále používat.

- Naše data bez sněhu, ozn. `nosnow`.
- Naše data se sněhem, ozn. `snow`.
- Naše data bez sněhu i se sněhem, ozn. `snow_nosnow`
- Data vytvořená Baetensem a spol., ozn. `baetens`
- Data vytvořená Baetensem a naše data bez sněhu, ozn. `nosnow_baetens`
- Data vytvořená Baetensem a naše data se sněhem, ozn. `snow_baetens`
- Data vytvořená Baetensem a naše data bez sněhu i se sněhem, ozn. `snow_nosnow_baetens`

Poměr mezi našimi daty a daty od Baetense a spol. bude vždy 50:50. Velikost každé trénovací množiny bude 750 000 pixelů jejichž rozlišení je 60m. Testovací množina bude obsahovat 250 000 pixelů. Dodržíme tak obecné doporučení rozdělit trénovací a testovací data v poměru 75:25. Tyto pixely jsme vybrali z celého datasetu náhodně podle uniformního rozdělení. V případech kdy model trénujeme na kombinaci našich dat a dat od Baetense a spol., tzn. `nosnow_baetens`, `snow_baetens`, `snow_nosnow_baetens`, pak bereme 500 000 z našich dat a 500 000 z dat Baetense a spol.

Naši trénovací množinu budeme muset standardizovat, tedy posunout na střední hodnotu 0 a jednotkový rozptyl. Tuto operaci musíme provést, protože naše validační množina od Hollsteina má jinou škálu než data od Sentinelu 2. Pro tento proces použijeme balíček `sci-kit learn`, který jsme zmiňovali výše. Konkrétní funkce se nazývá `sklearn.preprocessing.scale`.

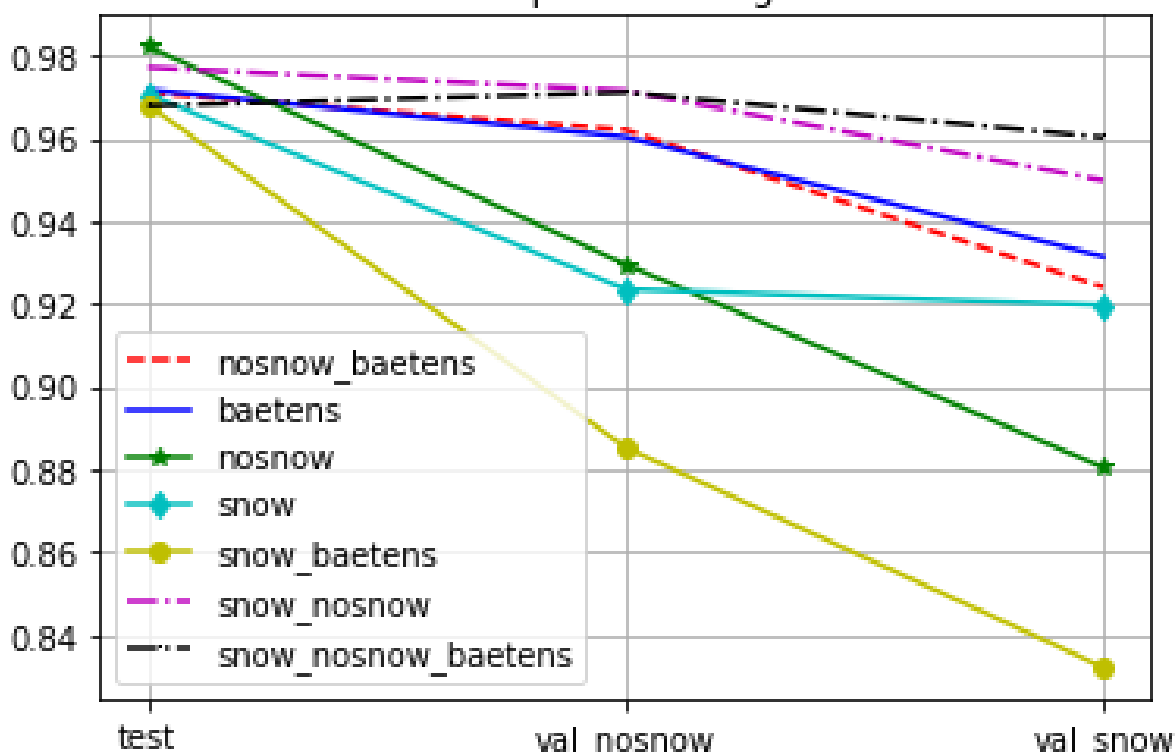
5.1.1 Výsledky single-pixel modelů

Na obrázku 5.1 můžeme vidět srovnání výsledků jednotlivých modelů trénovaných na různých kombinacích trénovacích dat. Vidíme na něm výsledky na testovacích datech, poté výsledky na single-pixel

validačních datech od Hollsteina ze kterých jsme vyřadili všechny pixely obsahující sněž a poté výsledky na kompletních single pixel validačních datech od Hollsteina.

Již jsme zmiňovali, že sněž a oblačnost mají velice podobnou odrazivost a z toho důvodu je jejich rozlišení největší překážkou úspěšné detekce oblačnosti. Pokud by náš algoritmus nedokázal správně rozlišit sněž a oblačnost a označoval všechny sněžové pixely za mraky budeme to chtít vědět. Takový algoritmus si také najde své využití a to buď pro detekci mraků v místech, kde se sněž nevyskytuje. Další užití by našel pro ty, kteří pracují s daty ze satelitních snímků a jsou pro ně nežádoucí, jak mraky tak sněž například při klasifikaci plodin na polích atd. Proto jsme se rozhodli rozdělit naši single-pixel validační množinu na množinu bez sněžu a množinu se sněžem.

Srovnání přesnosti algoritmů



Obrázek 5.1: Srovnání přesností single-pixel modelů

Můžeme si všimnout, že výsledky na testovacích datech jsou pro všechny modely velice podobné. Nejlepších výsledků dosahuje model nosnow, dále pak snow_nosnow. Nejhorší je algoritmus snow_nosnow_baetens. Nicméně podle výsledků na testovacích datech nedokážeme nijak výrazně posoudit kvalitu jednotlivých modelů.

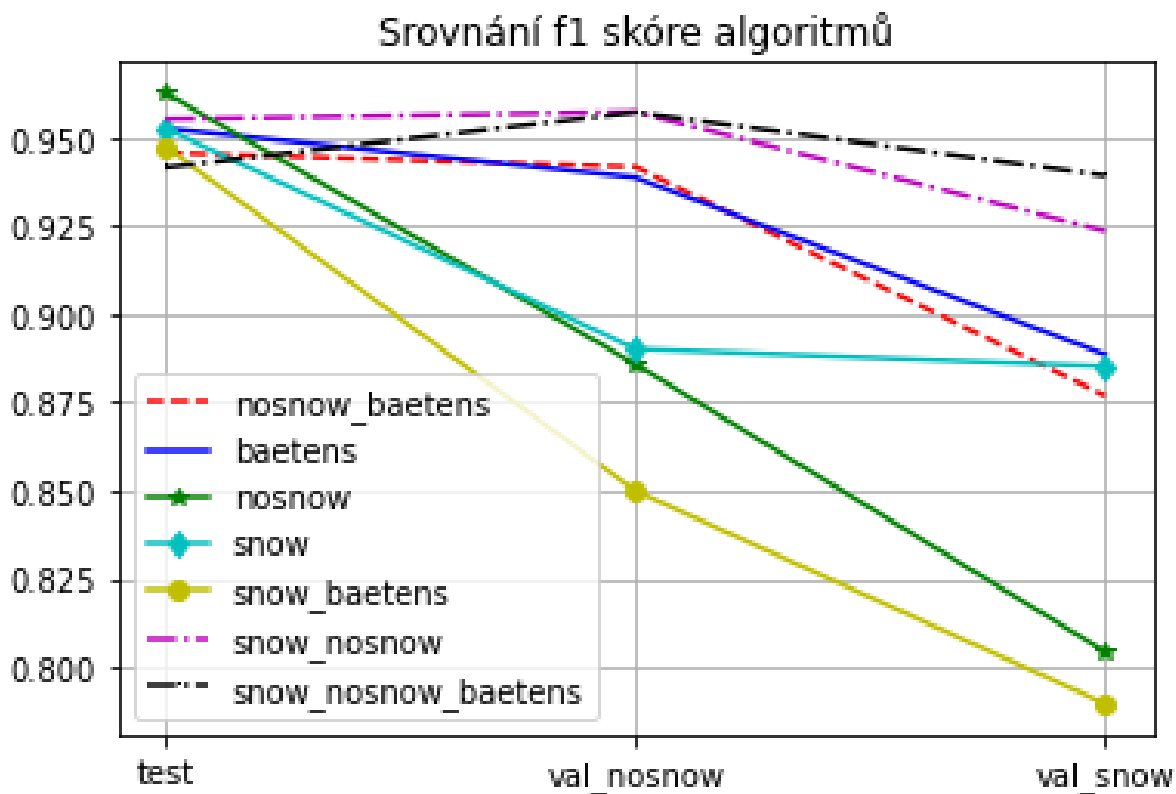
Na validačních single-pixel datech od Hollsteina bez sněžu dosáhli nejlepších přesností algoritmy snow_nosnow a snow_nosnow_baetens a to okolo 97 %. V druhé řadě jsou algoritmy baetens a no_snow baetens, které mají přesnost o procento nižší, což jsou stále velmi dobré výsledky. Horších výsledků dosáhly modely snow a nosnow a nejhorších snow_baetens. Kromě modelu snow_nosnow_baetens mají všechny modely horší výsledky než na testovacích datech, což je přirozené. Můžeme si všimnout, že na těchto reálných datech bez sněžu došlo k největšímu poklesu mezi těmito daty a testovacími daty u algoritmů snow_baetens, snow a nosnow. Již dříve jsme zmiňovali, že námi vytvořený dataset obsahující

sníh nemá takovou kvalitu, jako bychom si představovali a zde je první náznak, že tento nedostatek se překlopil do modelu natrénovaném na těchto datech.

Na validačních single-pixel datech od Hollsteina a spol. se sněhem dosáhli nejlepších přesností opět modely snow_nosnow a snow_nosnow_baetens a to 96 % a 95 %. Oproti validaci bez sněhu si můžeme všimnout, že přesnosti klesly pouze o 1 a 2 procentní body. Zde již máme nižší přesnost než na testovacích datech pro model snow_nosnow_baetens. Další jsou algoritmy baetens a nosnow_baetens, které dosáhli přesnosti 93 % a 92,5 %. Zde došlo k většímu propadu přesnosti o cca 3 % u obou.

Vidíme zajímavé chování pro modely snow a nosnow. U algoritmu nosnow došlo k více než 4% propadu v přesnosti na 88 %, zatímco u algoritmu snow nedošlo téměř k žádnému propadu. Z toho bychom mohli usuzovat, že algoritmus snow si poměrně dobře poradí s pixely obsahující sníh, které přibýli v této validaci, ale celkově nedetekuje oblačnost tak dobře jako jiné modely. Naproti tomu přítomnost sněhových pixelů snížila přesnost algoritmu, který při trénování nezískal informaci o rozdílu mezi sněhovými a mrakovými pixely. Nejhorším modelem je opět model snow_baetens.

Nyní se podívejme na srovnání pro nás důležitější metriky a tou je f1-skóre. To můžete vidět na obrázku 5.2.



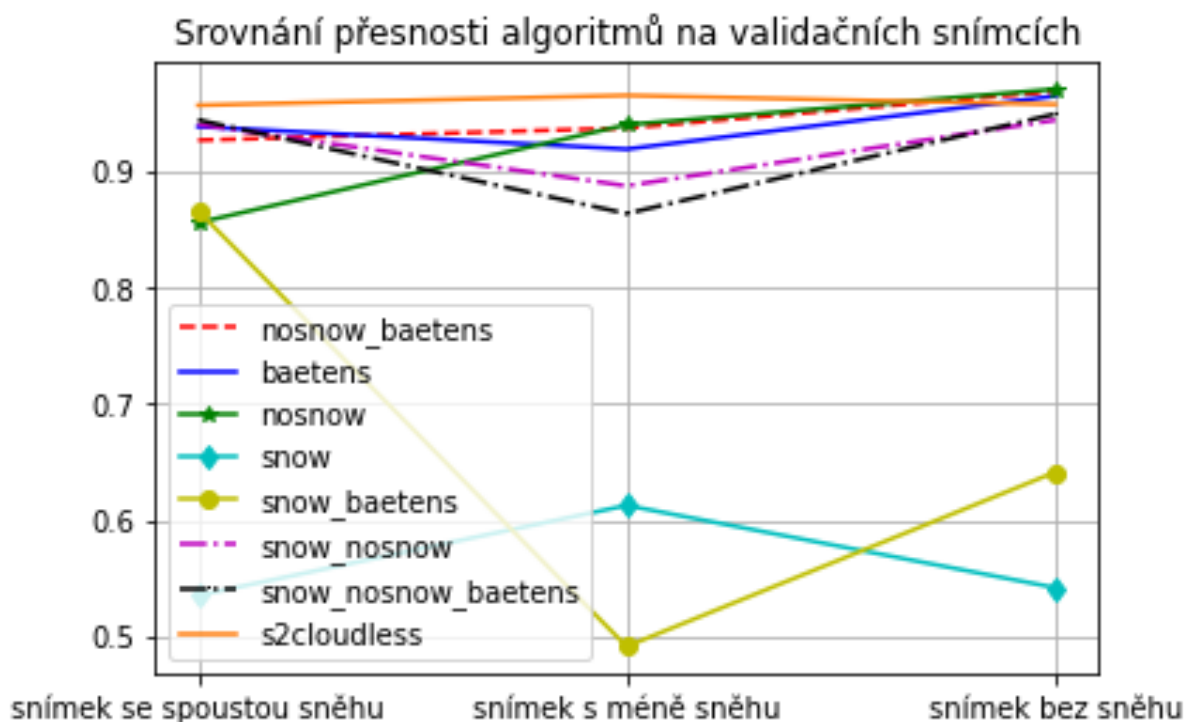
Obrázek 5.2: Srovnání f1-skóre single-pixel modelů

Stejně jako u metriky přesnost vidíme velmi podobné výsledky na testovacích datech pro všechny modely. Pojd' me tedy na výsledky na validačních datech bez sněhu. Zde vidíme podobný obrázek, jako u přesnosti, nejlepší jsou snow_nosnow a snow_nosnow_baetens následované baetens a nosnow_baetens. Výrazně horší jsou pak opět snow, nosnow a nejhorší snow_baetens. Opět zde došlo k zvláštnímu jevu, kdy má snow_nosnow a snow_nosnow_baetens lepší f1-skóre na validačních datech než na testovacích. Je ale třeba brát do úvahy, že data jsme uměle očistili o sníh.

Na validačních single-pixel datech od Hollsteina a spol. se sněhem dosáhl nejlepšího f1-skóre téměř 94 % algoritmus snow_nosnow_baetens následovaný modelem snow_nosnow s 92,5 %. Podobných hodnot f1-skóre na těchto datech dosáhli algoritmy snow, nosnow_baetens a baetens a nejhorších pak snow_baetens a nosnow.

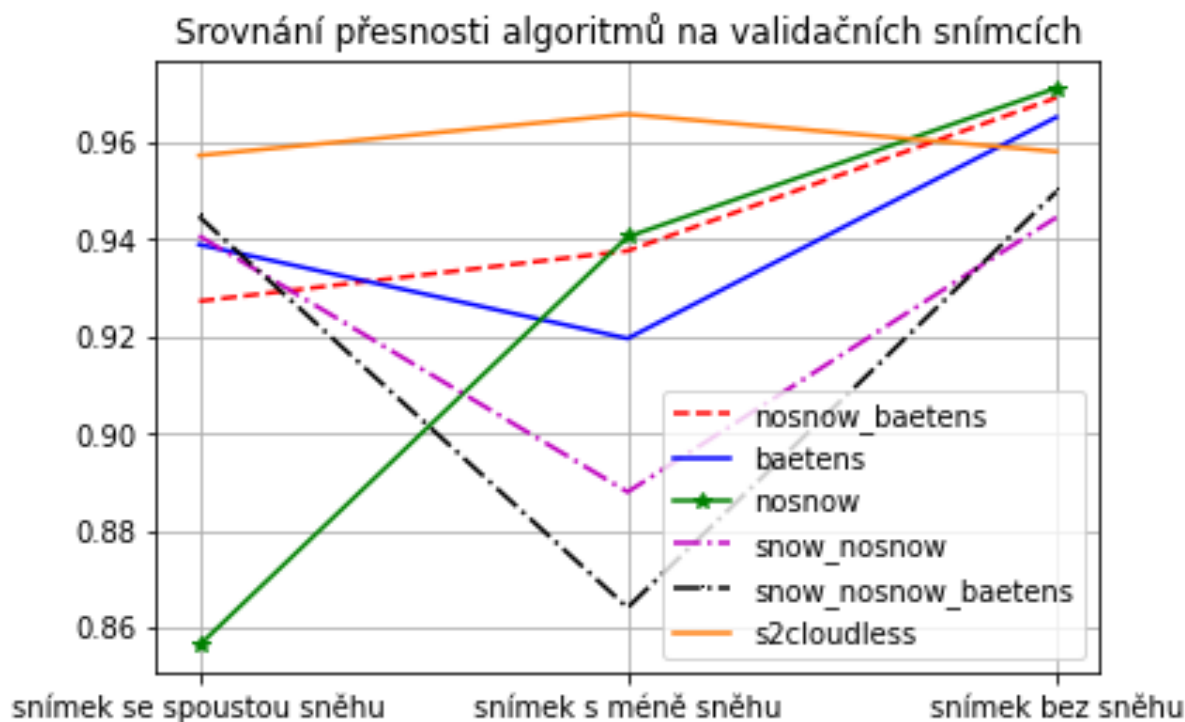
Nevýhodou validace na single-pixel datech je, že si detekované masky nemůžeme vizuálně ověřit a ani nevíme, jak data vypadají. Nebudeme tedy prozatím z těchto výsledků vyvozovat žádné závěry a počkáme si na výsledky našich modelů na celých snímcích od Hollsteina a spol. kde získáme, jak přesnost a f1-skóre, tak i možnost vizuální kontroly a budeme moci říct o jednotlivých algoritmech více.

Pojďme se tedy podívat na přesnost na validačních snímcích od Hollsteina a spol. Toto srovnání je podle nás nejvíce vypovídající protože se jedná o reálné snímky s člověkem vytvořenou maskou. Navíc na těchto snímcích můžeme použít algoritmus s2cloudless a srovnat jeho přesnost s přesností našich algoritmů. Srovnání jednotlivých algoritmů můžeme vidět na obrázku 5.3. Na ose X máme 3 různé snímky. První snímek označený jako snímek se spoustou sněhu můžeme vidět na obrázku 4.67, snímek s méně sněhu je na obrázku 4.69 a snímek bez sněhu je na obrázku 4.71.



Obrázek 5.3: Srovnání přesností single-pixel modelů na validačních snímcích

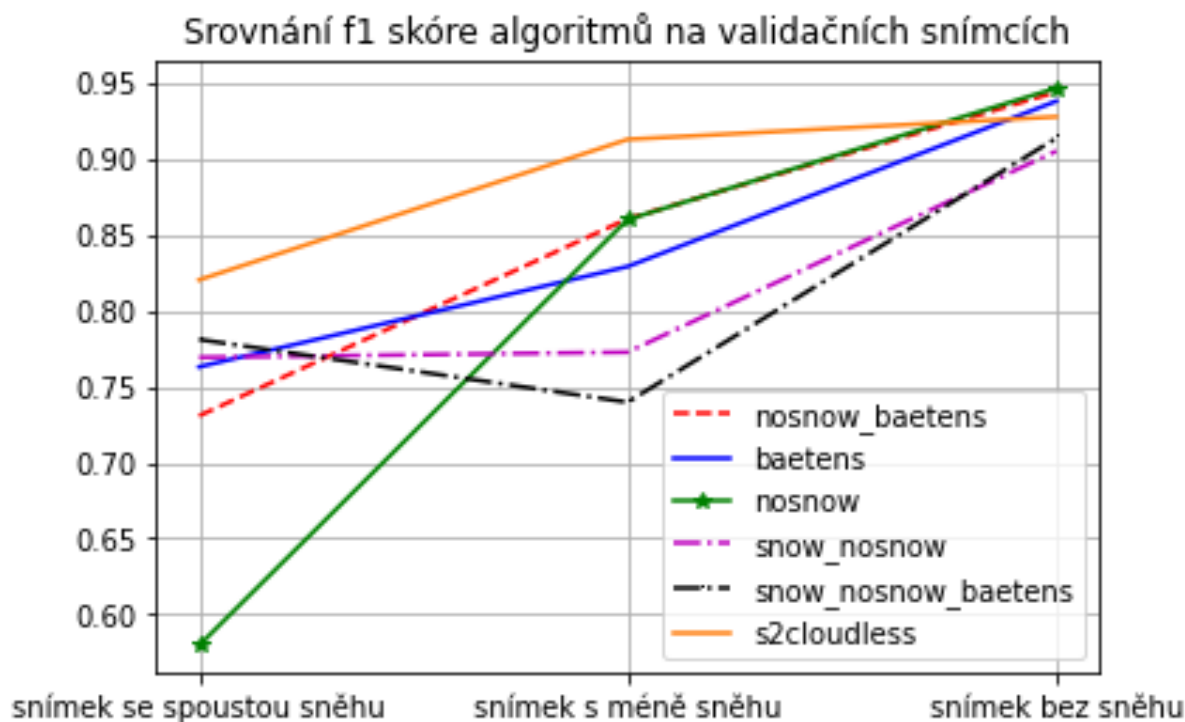
Jako první vidíme, že modely snow a snow_baetens nedosáhli vůbec dobrých výsledků a proto je ze srovnávání vyřadíme a budeme se věnovat lepším modelům. Abychom lépe viděli rozdíly mezi modely, které dosáhli lepších výsledků, vytvoříme si nový plot bez modelů snow a snow_baetens. Vidět ho můžeme na obrázku 5.4.



Obrázek 5.4: Srovnání přesností single-pixel modelů na validačních snímcích

Můžeme si všimnout, že detektor s2cloudless byl na snímcích obsahujících sněh o cca 2 % přesnější než jakýkoliv z našich modelů. Na snímku bez sněhu, ale několik našich modelů dosáhlo lepších výsledků. Nejdříve se podívejme na model nosnow. Ten dosáhl nejhorší přesnosti, tam kde bylo nejvíce sněhu. Na snímku s méně sněhu dosáhl druhé nejlepší přesnosti a na snímku bez sněhu nejlepší přesnosti. Model nosnow_baetens dosáhl podobných přesností na snímku s méně sněhu a bez sněhu, ale také dobrého výsledku na snímku s více sněhu. Byl na tomto snímku pouze o procento méně přesný než zbylé modely baetens, snow_nosnow_baetens a snow_nosnow. Tyto modely mají dobrý výsledek na snímku se spoustou sněhu nicméně na ostatních zaostávají.

Na obrázku 5.5 můžeme vidět srovnání f1-skóre pro jednotlivé modely na validačních snímcích.



Obrázek 5.5: Srovnání f1-skóre single-pixel modelů na validačních snímcích

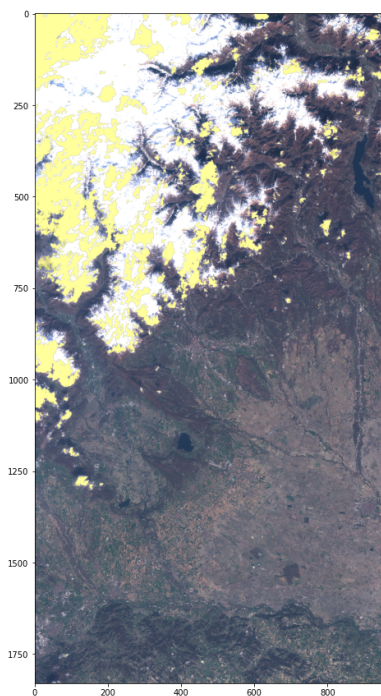
Na srovnání f1-skóre vidíme podobný obrázek, jaku u srovnávání přesnosti. S2cloudless je opět na snímcích se sněhovou pokrývkou lepší než jakýkoliv náš model. Na snímku bez sněhu, ale dosahují naše modely nosnow_baetens, baetens a nosnow o procento až dvě lepších výsledků. Nosnow_baetens dosahuje na snímku s méně sněhu nejlepších výsledků ze všech ostatních našich modelů. Nosnow model na tomto snímku dosahuje téměř stejných výsledků, ale na snímku s nejvíce sněhu je nejhorší. Algoritmus baetens má f1-skóre na snímku se spoustou sněhu o 3 % lepší než model nosnow_baetens, ale na snímku s menším množstvím je o 3 % horší. Modely trénované na námi vytvořené trénovací množině obsahující sněž, tedy snow_nosnow a snow_nosnow_baetens dopadli nejlépe na snímku obsahující největší množství sněhu. Na ostatních snímcích jsou nicméně nejhorší.

5.1.2 Vizuální srovnání single-pixel modelů

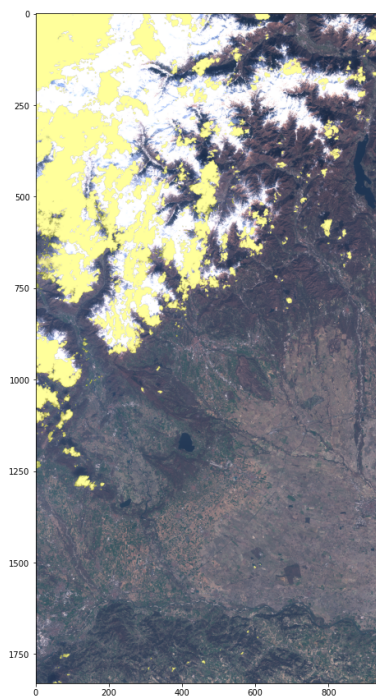
Díky tomu, že jsme získali ručně označené snímky vytvořené Hollsteinem a spol. tak máme možnost vizuálně porovnat, jak si vedou naše algoritmy při detekování mraku oproti referenčnímu snímku a také v porovnání s algoritmem s2cloudless.

Na obrázku 5.6 můžeme vidět srovnání reálné masky snímku se spoustou sněhu a výsledku algoritmu s2cloudless. Tento algoritmus si vede velmi dobře jedině nedostatky lze nalézt při rozlišování sněhových ploch od oblačnosti, která je nad těmito plochami.

Na obrázcích 5.7 a 5.8d jsou výsledky našich modelů. Ty vypadají všechny víceméně podobně. Oproti algoritmu s2cloudless můžeme vidět, že naše modely mají větší problémy s rozlišením sněhu a oblačnosti. Označují větší část sněhu jako oblačnost nicméně neoznačují všechny sněž na snímku jako oblačnost. Z toho soudíme, že určitou rozlišovací schopnost mají. Výjimkou je algoritmus nosnow, který nebyl trénován na žádných datech, které by obsahovali sněž a očekávaně označuje všechny sněhové plochy jako oblačnost.

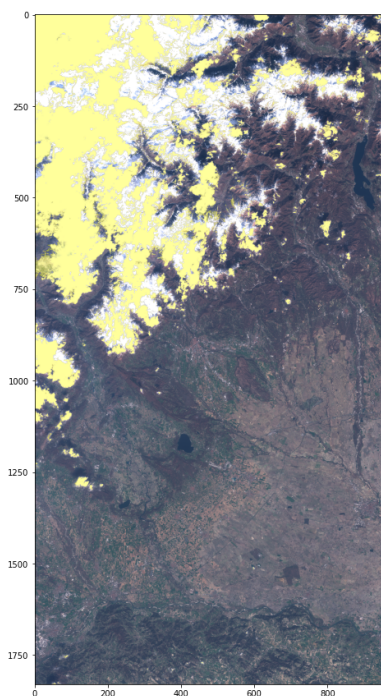


(a) Referenční snímek

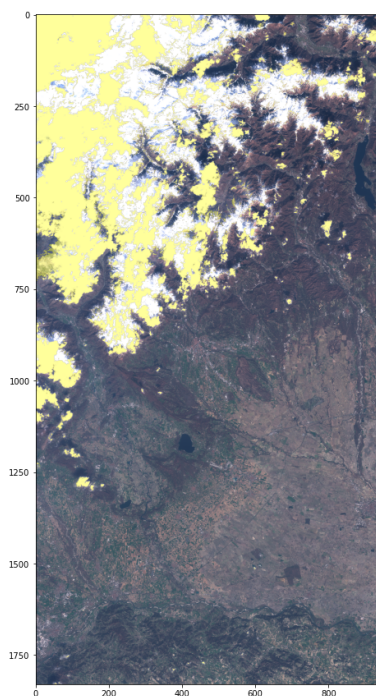


(b) s2cloudless

Obrázek 5.6: Výsledek s2cloudless na snímku se spoustou sněhu

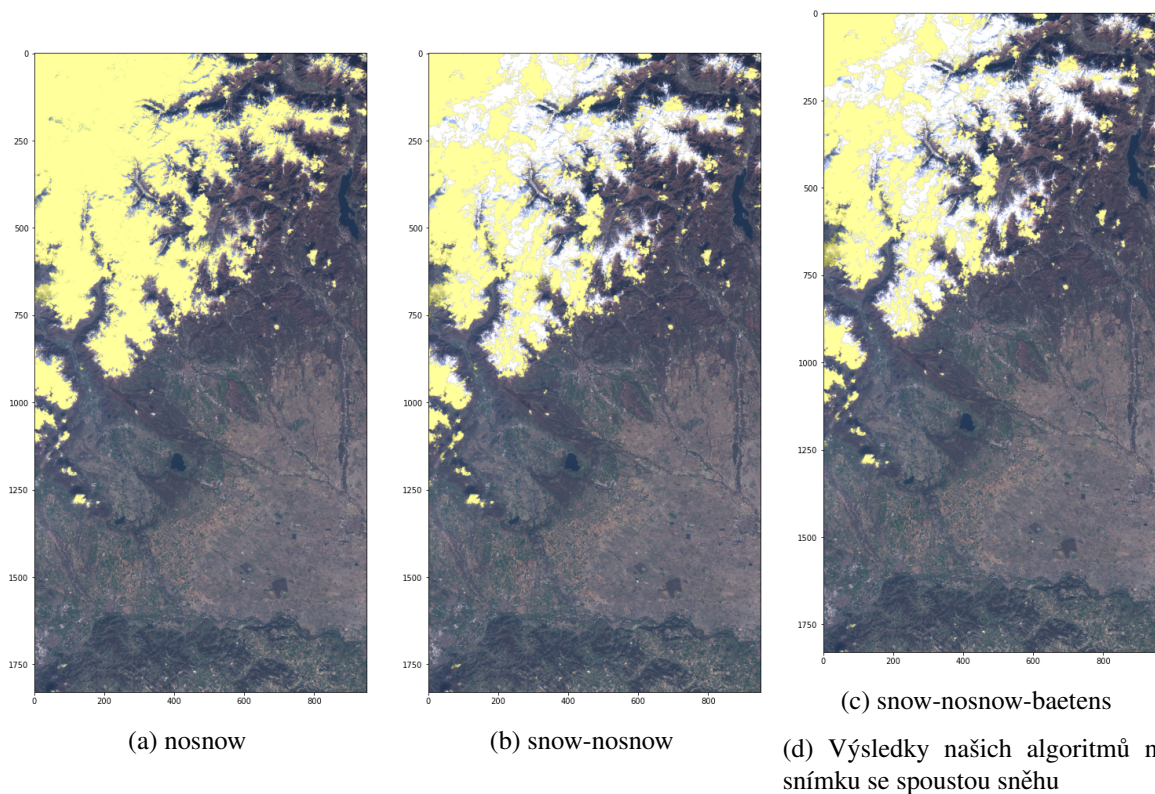


(a) nosnow-baetens



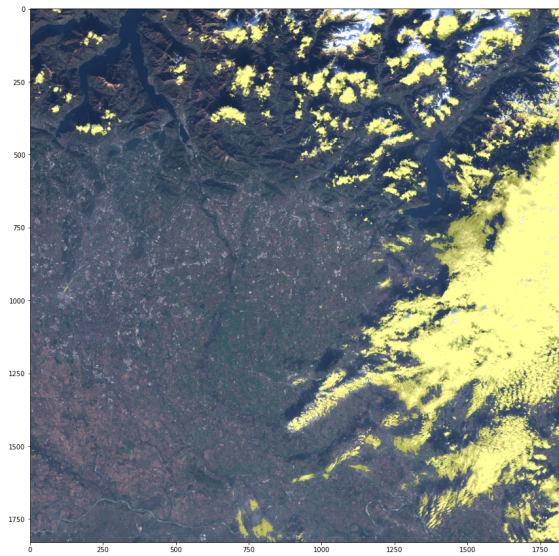
(b) baetens

Obrázek 5.7: Výsledky našich algoritmů na snímku se spoustou sněhu

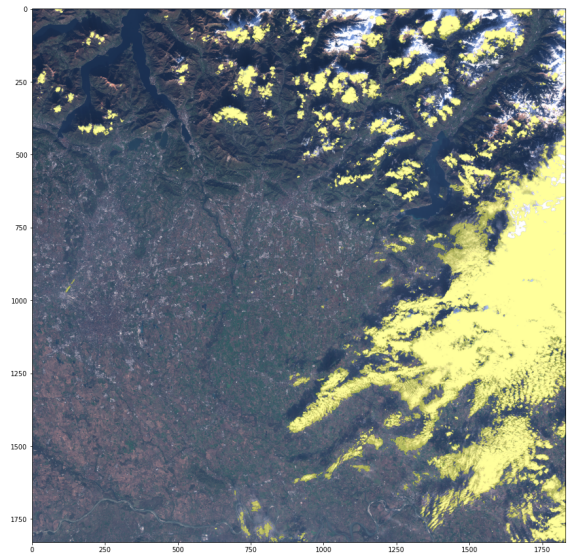


Na obrázku 5.9 vidíme referenční snímek s méně sněhu a predikce modelu `s2cloudless` pro tento snímek. Zde lze nalézt dvě drobné chyby tohoto modelu. První je opět rozlišení mezi sněhem a oblačností a druhou je označení několika ploch s oblačností jako místa bez výskytu mraků.

Naše výsledky můžeme vidět na obrázku 5.10. Všechny naše algoritmy měli opět problém s rozlišováním sněhu a oblačností, ale některé sněhové plochy označili správně. Algoritmus `nosnow` opět označil všechny sněhové plochy jako oblačnost a v jednom případě neoznačil jemnou oblačnost v jižní části snímku. Problém měli naše algoritmy i s rozeznáváním jasných ploch nacházejících se ve městech. Často tyto plochy označili jako oblačnost. Nejhůře na tom byli algoritmy `snow_nosnow` a `snow_nosnow_baetens`. Trochu lépe si vedl model `baetens` a nejlépe při klasifikaci oblačnosti na tomto obrázku dopadl `nosnow_baetens`. Tento model trpí tímto nedostatkem také, nicméně v daleko omezenější míře.

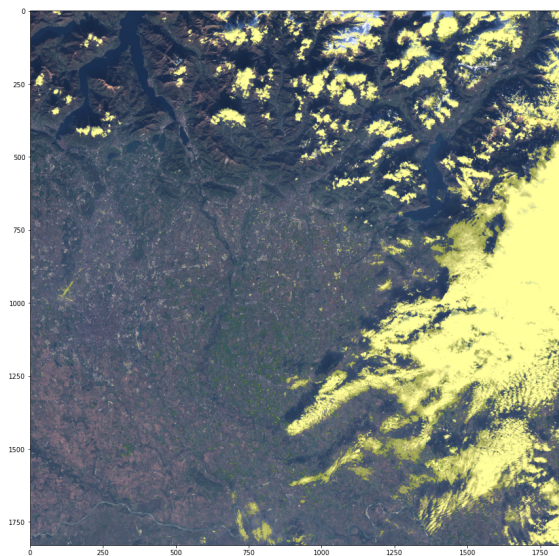


(a) Referenční snímek

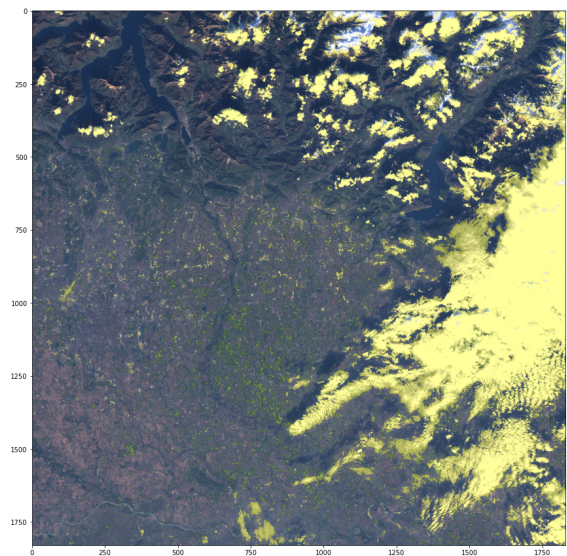


(b) s2cloudless

Obrázek 5.9: Výsledek s2cloudless na snímku s méně sněhu

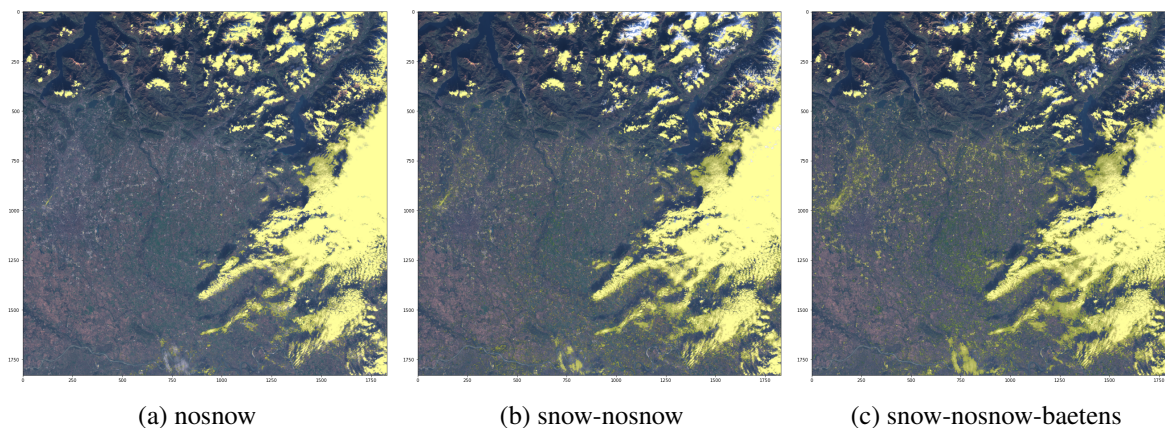


(a) nosnow-baetens



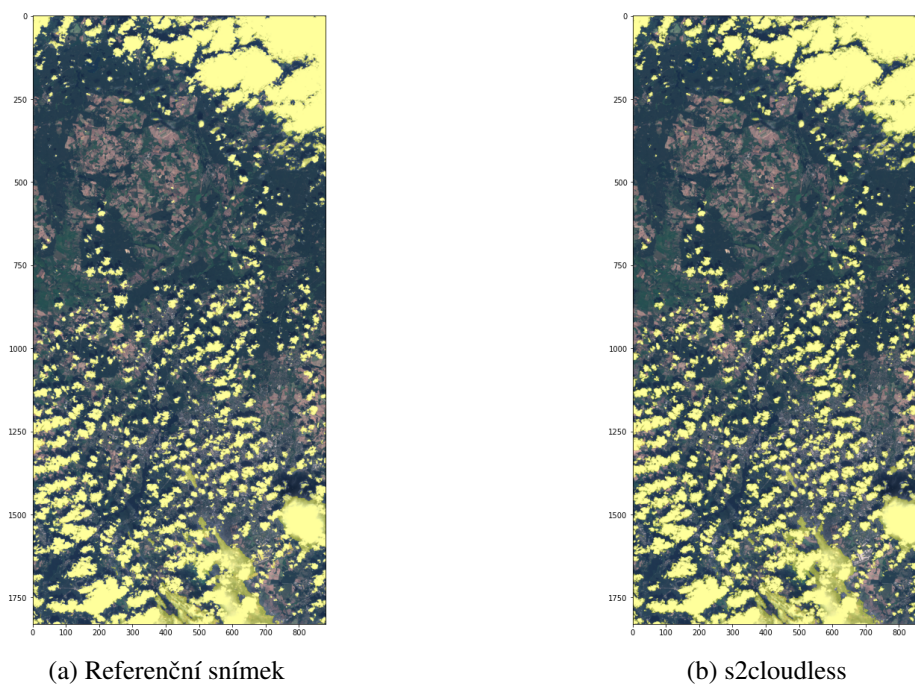
(b) baetens

Obrázek 5.10: Výsledky našich algoritmů na snímku s méně sněhu

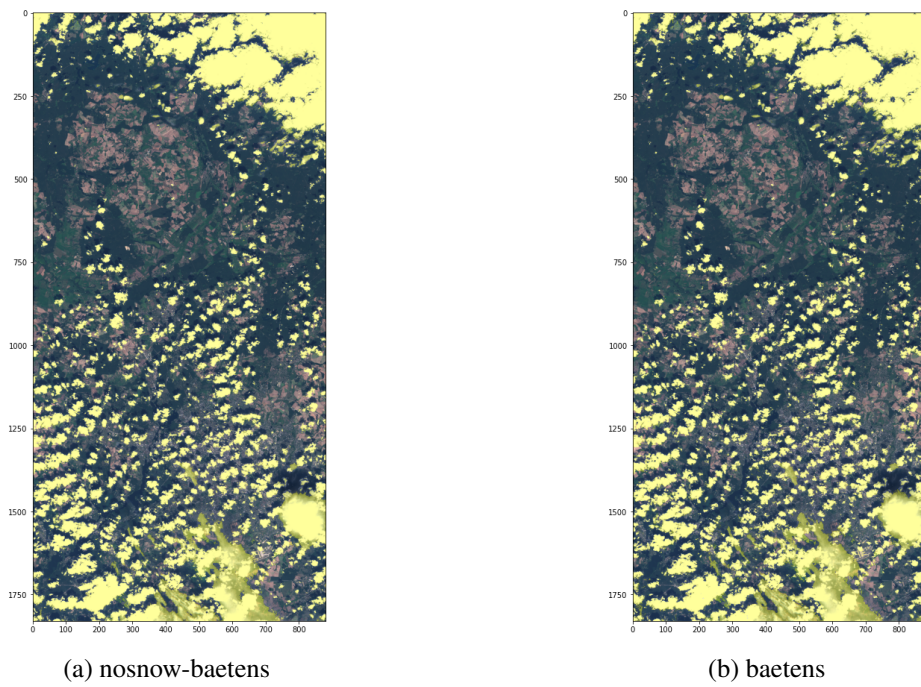


Obrázek 5.11: Výsledky našich algoritmů na snímku s méně sněhu

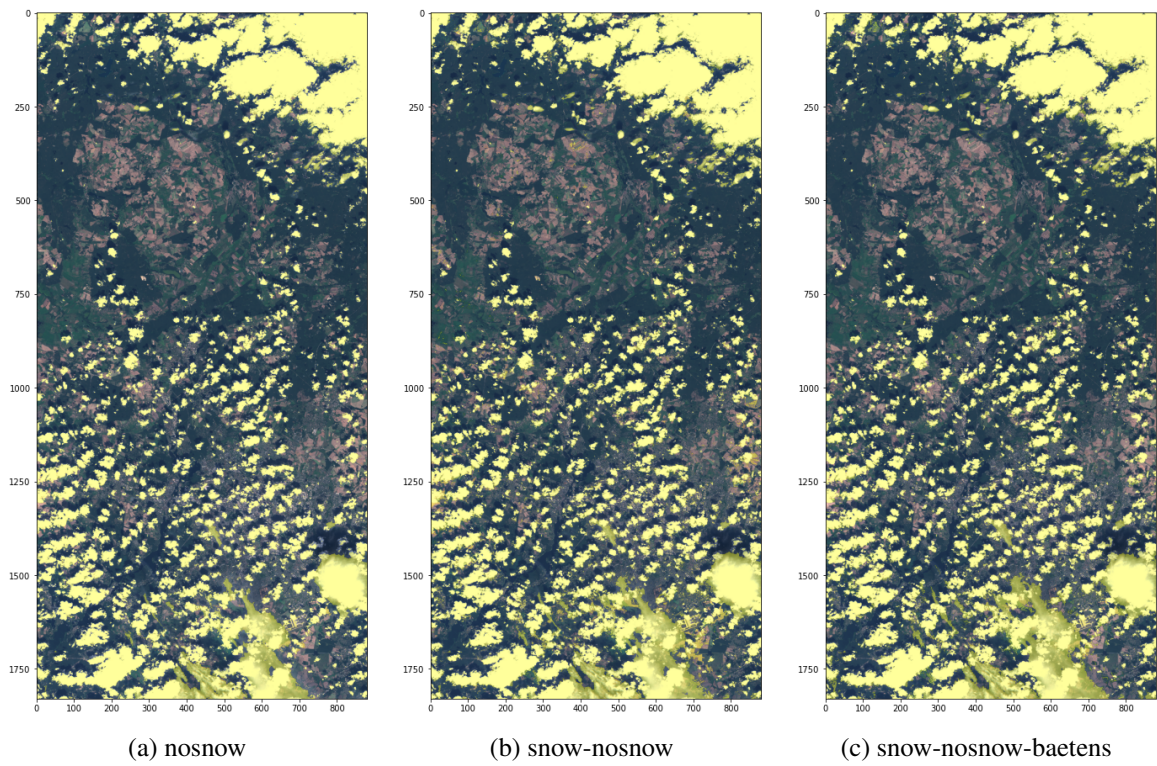
Na obrázcích 5.12 až 5.14 vidíme výsledky pro referenční snímek bez sněhu. Zde performovaly všechny naše modely i s2cloudless výborně a vizuálně lze nalézt rozdíly mezi jednotlivými maskami jen stěží.



Obrázek 5.12: Výsledek s2cloudless na snímku bez sněhu



Obrázek 5.13: Výsledky našich algoritmů na snímku bez sněhu



Obrázek 5.14: Výsledky našich algoritmů na snímku bez sněhu

Na základě námi porovnávaných metrik a vizuálních výsledků nám přijde, že nejlépe z námi zkouma-

ných algoritmů performoval algoritmus nosnow_baetens, který dosáhl na reálných člověkem označených datech uspokojivých výsledků. Na dvou snímcích byl nejlepší a na dalším dosáhl velice podobných výsledků, jako ostatní modely. Při vizuálním porovnání byl z námi srovnávaných modelů také nejlepší. Z těchto důvodů budeme nadále zkoumat pouze tuto kombinaci trénovacích dat. Pokusíme se srovnat, jakého zlepšení dosáhne náš model po přidání informace o okolních pixelech.

5.2 Multi-pixel Náhodné lesy

Naším dalším cílem bylo prozkoumat, jak selepší či zhorší námi navržený algoritmus, při využití závislosti mezi sousedními pixely. Trénovací data tedy upravíme tak, aby každé jednotlivé y_i mělo k dispozici $x_{i,1}$ až $x_{i,9}$ $i \in n$. $x_{i,1}$ je tedy hodnota v určitého spektrálního pásma na stejném místě jako y_i a $x_{i,2}$ až $x_{i,9}$ jsou hodnoty určitého spektrálního pásma v místech, které sousedí s pozicí $x_{i,1}$. Náš prediktor M (feature vector) bude vypadat následovně:

$$M_{n,p} = [a \quad b \quad \cdots \quad p], \quad a, b, \dots, p \in \mathbf{R}^{n \times 9}, \quad (5.3)$$

kde

$$a_i = [a_{i,1} \quad a_{i,2} \quad \cdots \quad a_{i,9}] \quad \forall i \in n. \quad (5.4)$$

Budeme používat stejná pásma jako v předchozím případě a tudíž i zde bude $p = 10$. Opět jsme provedli hledání v síti (gridsearch) pro nalezení ideální kombinace parametrů `n_estimators` a `max_depth`.

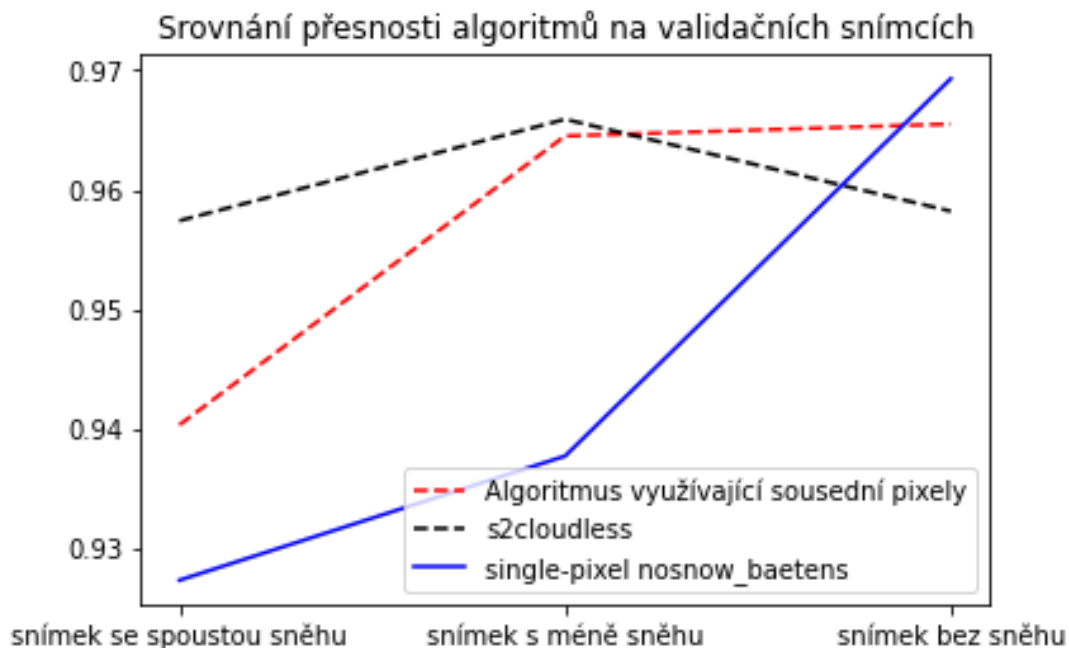
Tento algoritmus budeme trénovat na datech vytvořených Baetensem a spol. a našich datech bez sněhu. Poměr mezi těmito daty bude 1:1 a velikost trénovací množiny bude 750 000 datových bodů jejichž rozlišení bude 60m. Testovací množina bude obsahovat 250 000 datových bodů a tudíž dodržíme poměr trénovacích a testovacích dat 75:25. Jednotlivé datové body byli vybrány náhodně podle uniformního rozdělení. Každý datový bod bude obsahovat 9 pixelů s informací z 10 spektrálních pásem. Tyto data opět standardizujeme pomocí `sklearn.preprocessing.scale`.

5.2.1 Výsledky multi-pixel modelu

Na testovacích datech dosáhl náš model přesnosti 97,32 % a f1-skóre 94,47 %. Jeho přesnost tak byla o 0,2 % vyšší než u single-pixel modelu nosnow_baetens a jeho f1-skóre bylo o 0,1 % nižší než u single-pixel modelu nosnow_baetens. Na testovacích datech jsme tedy přidáním informace o sousedních pixelech nedosáhli výrazného zlepšení.

V předchozím případě jsme modely validovali na single-pixel datech od Hollsteina a spol., ale to v případě multi-pixel modelu nebude možné. Validovat jej tedy budeme na snímcích od Hollsteina a spol.

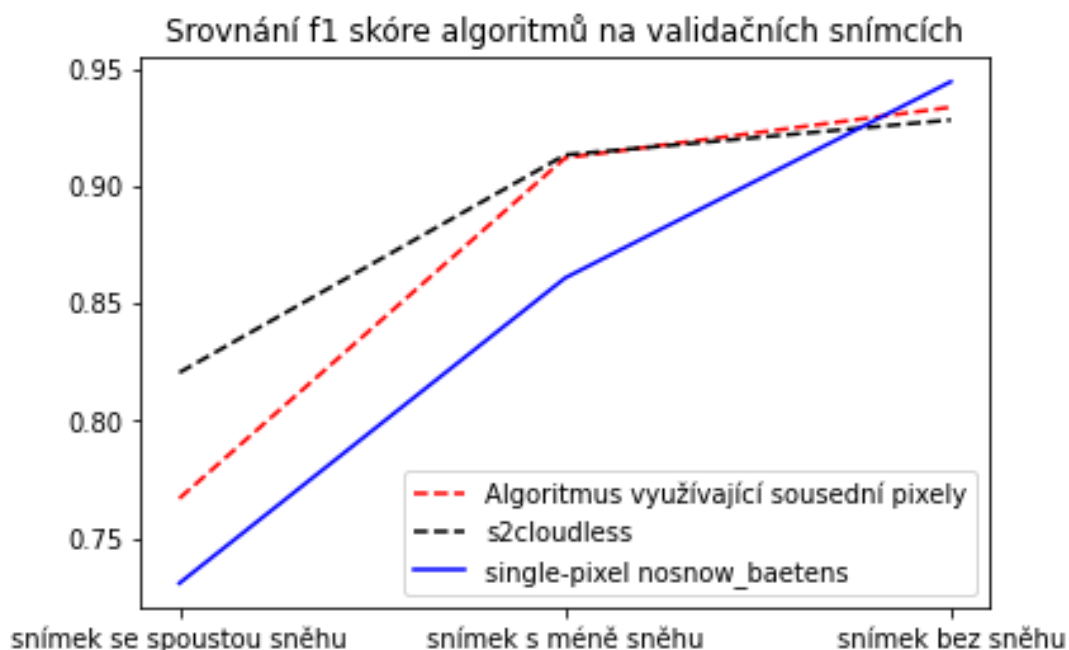
Na obrázku 5.15 můžeme vidět srovnání výsledků multi-pixel modelu s algoritmem `s2cloudless` a se single-pixel modelem `nosnow_baetens`.



Obrázek 5.15: Srovnání přesností multi-pixel modelu na validačních snímcích

Algoritmus využívající sousední pixely se na obou snímcích obsahujících sněh zlepšil poměrně výrazně. Na snímku se spoustou sněhu je o cca procento lepší a na snímku s méně sněhu dosáhl téměř stejného výsledku jako s2cloudless. Zlepšil se tak o více než 2 procenta oproti single-pixel modelu. Na snímku bez sněhu je jeho přesnost mírně horší než single pixel model, ale stále je lepší než s2cloudless.

Podívejme se nyní na srovnání f1-skóre na obrázku 5.16.

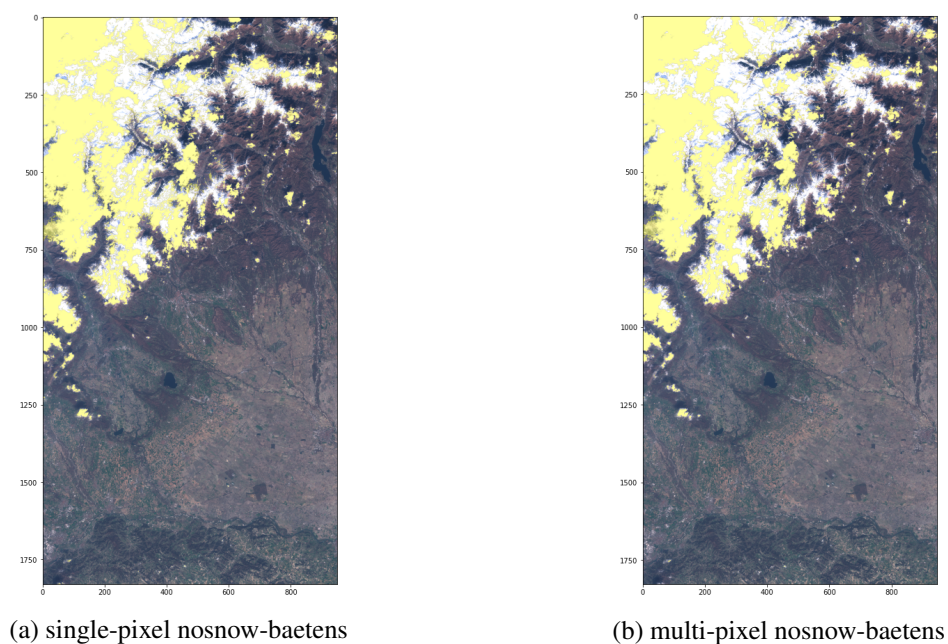


Obrázek 5.16: Srovnání f1-skóre multi-pixel modelu na validačních snímcích

Vidíme stejné chování jako u přesnosti. Na snímku s méně sněhu multi-pixel model dosáhl více méně stejného výsledku jako s2cloudless, na snímku bez sněhu lepšího než s2cloudless a na snímku se spoustou sněhu se zlepšil o cca 3,5 % oproti single-pixel modelu. Porovnejme nyní výsledky vizuálně.

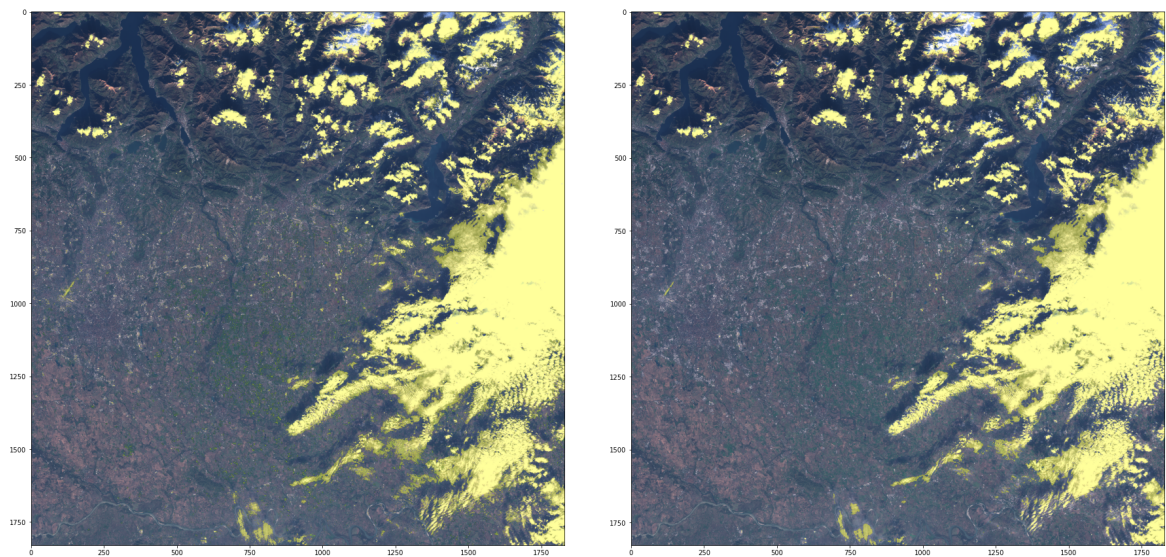
5.2.2 Vizuální srovnání multi-pixel modelu

Na obrázku 5.17 vidíme srovnání multi-pixel modelu se single-pixel modelem. I multi-pixel model má problémy s rozlišováním sněhu a oblačnosti nicméně si lze všimnout, že na několika málo místech ponechal více sněhu než single-pixel model.



Obrázek 5.17: Výsledky našich algoritmů na snímku se spoustou sněhu

Na obrázku 5.18 vidíme nejvýraznější rozdíl mezi single-pixel modelem a multi-pixel modelem. Multi-pixel model daleko méně často chybuje při klasifikaci pixelů z městských oblastí s vysokou odrazivostí.

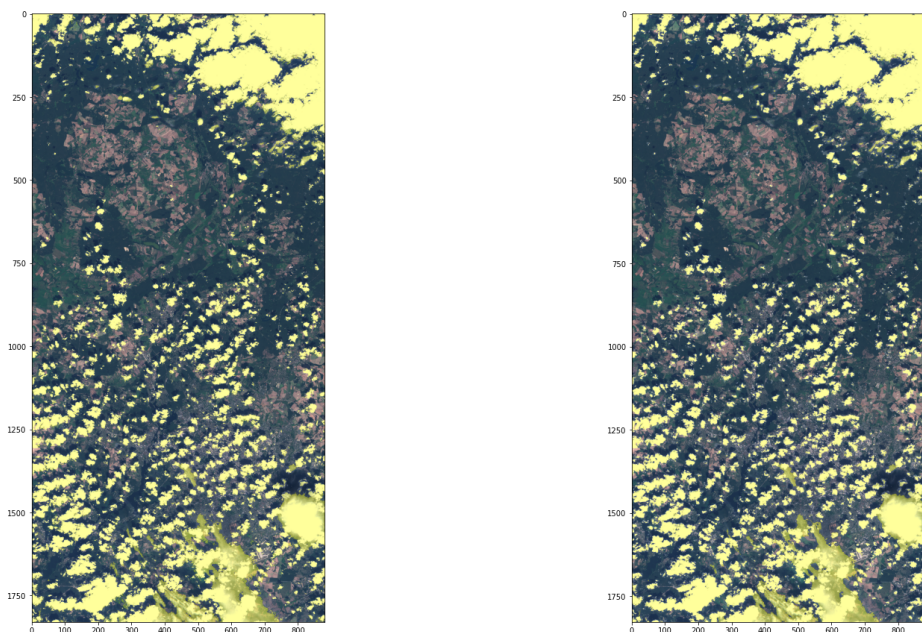


(a) single-pixel nosnow-baetens

(b) multi-pixel nosnow-baetens

Obrázek 5.18: Výsledky našich algoritmů na snímku s méně sněhu

Na obrázku 5.19 téměř nelze najít rozdíl mezi těmito dvěma modely.



(a) single-pixel nosnow-baetens

(b) multi-pixel nosnow_baetens

Obrázek 5.19: Výsledky našich algoritmů na snímku bez sněhu

Na základě srovnání single-pixel algoritmu a multi-pixel algoritmu můžeme tvrdit, že využití závislosti mezi sousedními pixely našemu algoritmu zlepšilo, jak přesnost tak f1-skóre a vizuální srovnání na nám dostupných snímcích toto tvrzení potvrdilo. Přidáním této závislosti se nám na dvou snímcích podařilo dosáhnout téměř stejných přesností a f1-skóre jako algoritmus s2cloudless, který je využíván sentinelhubem a na dalším se mu přiblížit.

5.3 Srovnání single-pixel modelu, multi-pixel modelu a s2cloudless

Na závěr zde uvedeme srovnání nejlepšího single-pixel modelu, multi-pixel modelu a algoritmu s2cloudless. Uvedeme průměrnou přesnost a průměrné f1-skóre, které jednotlivé modely získali na validačních snímcích. Nebude průměrovat přes všechny pixely, místo toho sečteme výsledky jednotlivých modelů na jednotlivých snímcích a podělíme je celkovým počtem snímků. Tím se vyhneme zkreslení, kterého bychom dosáhli například tím, že by nějaký algoritmus byl skvělý na snímku, který má více pixelů než ostatní snímky, ale horší na snímku s méně pixely.

Tyto průměry můžeme vidět v tabulce 5.3

	single-pixel nosnow-baetens	multi-pixel nosnow-baetens	s2cloudless
průměrná přesnost	0.945	0.957	0.96
průměrné f1-skóre	0.845	0.87	0.887

V tabulce 5.3 můžeme vidět že průměrná přesnost nejlepšího single-pixel modelu je cca o 1,5% nižší než u algoritmu s2cloudless. F1-skóre má náš nejlepší single-pixel model o cca 4% nižší. Multi-pixel model je srovnatelnější s modelem s2cloudless. Jeho průměrná přesnost je nižší pouze o 0,3 % a průměrné skóre je nižší o cca 1.7 %.

Závěr

Seznámili jsme se se specifikací družice Sentinel-2, s přístupem k jejím datům a základy strojového učení. Také jsme se seznámili s řadou algoritmů na detekci oblačnosti v satelitních snímcích a popsali jejich fungování.

Pomocí časových rozdílů a zkoumání vlivu oblačnosti na různá spektrální pásma jsme vytvořili trénovací množinu satelitních snímků s příslušnou segmentací s přihlédnutím k celoroční použitelnosti na území EU. Vytvořili jsme dva přístupy, jeden pro snímky bez sněhu a druhý pro snímky se sněhovými plochami. Pro snímky bez sněhu tento přístup fungoval velmi dobře. Pro snímky se sněhem měl své limity a pro finální algoritmus jsme tyto snímky nevyužili. Dále jsme si opatřili další trénovací data vytvořená Louistem Baetensem a jeho týmem. Díky těmto datům jsme získali lepší trénovací data obsahující sněh než ty námi vytvořené. Opatřili jsme si také dva druhy validačních dat, prvním byli jednotlivě označené pixely a druhým byli celé snímky. Tyto data vytvořil André Hollstein se svým týmem.

Navrhli jsme a vytvořili vlastní algoritmus typu Náhodné lesy, který detekuje oblačnost v satelitních snímcích Sentinelu-2. Dále jsme experimentovali s různými kombinacemi trénovacích dat. Zjistili jsme že pro algoritmus, který detekoval oblačnost na základě jednoho pixelu, fungovala nejlépe kombinace našich trénovacích dat bez sněhu a dat vytvořených Baetensem a spol. Na třech reálných snímcích, které byli označené člověkem jsme dosáhli průměrné přesnosti 94,5 % a f1-skóre 84,5 %. Jeden z nejlepších detektorů současnosti používaný Sentinel Hubem, enginem pro zpracování petabajtů satelitních dat, s2cloudless dosáhl průměrné přesnosti 96 % a f1-skóre 88,7 %. Největší slabinou našeho algoritmu bylo rozlišování sněhu a oblačnosti. Tato slabina způsobila největší část nepřesností. Další slabina byla chybná klasifikace velmi jasných pixelů ve městech. Tuto slabinu se nám podařilo potlačit dalším algoritmem, který jsme vytvořili.

Pro kombinaci našich trénovacích dat a dat vytvořených Baetensem jsme vytvořili další algoritmus, který detekuje oblačnost v pixelu za použití daného pixelu a sousedních pixelů. Tento algoritmus dosáhl průměrné přesnosti 95,7 % a průměrného f1-skóre 87 %. Chybná klasifikace velmi jasných pixelů u tohoto modelu byla potlačena nicméně problém s rozlišováním sněhu a oblačnosti přetrvával.

Pro vylepšení našeho detektoru oblačnosti bychom museli vyřešit problém se záměnou sněhu a oblačnosti. Nejpřímočařejším řešením tohoto problému by bylo dodáním většího množství různorodějších trénovacích dat obsahujících sněh a oblačnost. V budoucí práci by bylo možné si takovýto dataset vytvořit například pomocí nástroje ALCD [16].

Literatura

- [1] R.E. Huschke, *Glossary of Meteorology*, American Meteorological Society, Boston, MA, 1970
- [2] F. Hu, Q. Hao, *Intelligent Sensor Networks: The Integration of Sensor Networks*, Signal Processing and Machine Learning (1st ed.). CRC Press, 2013
- [3] P. Cabena, P. Hadjinian, R. Stadler, J. Verhees, and A. Zanasi. *Discovering Data Mining: From Concept to Implementation*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1998
- [4] L. Breiman, *Random forests*. Machine Learning 45, 5–32, 2001
- [5] A. Hollstein, K. Segl, L. Guanter, M. Brell, M. Enesco *Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in Sentinel-2 MSI images* Remote Sens., Vol. 8 (8) p. 666, ISSN 2072-4292, 2016
- [6] D. P. Roy, J. Ju, K. Kline, P. L. Scaramuzza, V. Kovalsky, M. Hansen, T. R. Loveland, E. Vermote, Ch. Zhang, *Web-enabled Landsat Data (WELD): Landsat ETM+ composited mosaics of the conterminous United States*, Remote Sensing of Environment, Volume 114, Issue 1, Pages 35-49, ISSN 0034-4257, 2010
- [7] E. Vermote, N. Saleous, Ch. Justice, *Atmospheric correction of MODIS data in the visible to middle infrared: first results*, Remote Sensing of Environment, Volume 83, Issues 1–2, Pages 97-111, ISSN 0034-4257, 2002
- [8] A. Huete, K. Didan, T. Miura, E.P. Rodriguez, X. Gao, L.G. Ferreira, *Overview of the radiometric and biophysical performance of the MODIS vegetation indices*, Remote Sensing of Environment, Volume 83, Issues 1–2, Pages 195-213, ISSN 0034-4257, 2002
- [9] Y. Zhang, B. Guindon, J. Cihlar, *An image transform to characterize and compensate for spatial variations in thin cloud contamination of Landsat images*, Remote Sensing of Environment, Volume 82, Issues 2–3, Pages 173-187, ISSN 0034-4257, 2002
- [10] K. Tarrío, X. Tang, J. G. Masek, M. Claverie, J. Ju, S. Qiu, Z. Zhu, C. E. Woodcock, *Comparison of cloud detection algorithms for Sentinel-2 imagery*, Science of Remote Sensing, Volume 2, 100010, ISSN 2666-0172, 2020
- [11] Z. Zhu, C. E. Woodcock, *Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change*, Remote Sensing of Environment, Volume 152, Pages 217-234, ISSN 0034-4257, 2014
- [12] S. Skakun, E. Vermote, J.-C. Roger and Ch. Justice, *Multispectral Misregistration of Sentinel-2A Images: Analysis and Implications for Potential Applications*, in IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 12, pp. 2408-2412, 2017

- [13] C.-C. Liu, Y.-C. Zhang, P.-Y. Chen, C.-C. Lai, Y.-H. Chen, J.-H. Cheng, M.-H. Ko, *Clouds Classification from Sentinel-2 Imagery with Deep Residual Learning and Semantic Image Segmentation*, Remote Sensing vol. 11, no. 2, ISSN 2072-4292, 2019
- [14] Y. Shendryk, Y. Rist, C. Ticehurst, P. Thorburn, *Deep learning for multi-modal classification of cloud, shadow and land cover scenes in PlanetScope and Sentinel-2 imagery*, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 157, Pages 124-136, ISSN 0924-2716, 2019
- [15] A. Zupanc, *Improving Cloud Detection with Machine Learning*. 2019. Available online: <https://medium.com/sentinel-hub/improving-cloud-detection-with-machine-learning-c09dc5d7cf13> (accessed on 29 November 2019)
- [16] L. Baetens, C. Desjardins, O. Hagolle, *Validation of copernicus Sentinel-2 cloud masks obtained from MAJA, Sen2Cor, and FMask processors using reference cloud masks generated with a supervised active learning procedure*. Rem. Sens. 11, 433, 2019
- [17] S. Skakun, J. Wevers, C. Brockmann, G. Doxani, M. Aleksandrov, M. Batič, D. Frantz, F. Gascon, L. Gómez-Chova, O. Hagolle, D. López-Puigdollers, J. Louis, M. Lubej, G. Mateo-García, J. Osman, D. Peressutti, B. Pflug, J. Puc, R. Richter, J.-C. Roger, P. Scaramuzza, E. Vermote, N. Vesel, A. Zupanc, L. Žust, *Cloud Mask Intercomparison eXercise (CMIX): An evaluation of cloud masking algorithms for Landsat 8 and Sentinel-2*, Remote Sensing of Environment, Volume 274, 112990, ISSN 0034-4257, 2022
- [18] D.H. Wolpert, W.G. Macready, *No free lunch theorems for optimization*. IEEE Trans. Evol. Comput. 1, 67–82, 1997
- [19] Z. Zhu, C.E. Woodcock, *Object-based cloud and cloud shadow detection in LANDSAT imagery*. Remote Sens. Environ. 118, 83–94, 2012
- [20] Z. Zhu, S. Wang, C.E. Woodcock, *Improvement and expansion of the FMask algorithm: Cloud, cloud shadow, and snow detection for LANDSATs 4–7, 8, and Sentinel 2 images*. Remote Sens. Environ. 159, 269–277, 2015
- [21] O. Hagolle, M. Huc, D.V. Pascual, G. Dedieu, *A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENS, LANDSAT and SENTINEL-2 images*. Remote Sens. Environ. 114 (8), 1747–1755, 2010
- [22] A. Lyapustin, Y. Wang, R. Frey, *An automatic cloud mask algorithm based on time series of MODIS measurements*. Journal of Geophysical Research, 113, D16207, 2008
- [23] S. Skakun, E.F. Vermote, J.-C. Roger, C.O. Justice, J.G. Masek, *Validation of the LaSRC cloud detection algorithm for Landsat 8 images*. IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 12 (7), 2439–2446, 2019
- [24] J.H. Jeppesen, R.H. Jacobsen, F. Inceoglu, T.S. Toftegaard, *A cloud detection algorithm for satellite imagery based on deep learning*. Remote Sens. Environ. 229, 2019 247–259
- [25] M.J. Hughes, D.J. Hayes, *Automated detection of cloud and cloud shadow in single-date Landsat imagery using neural networks and spatial post-processing*. Remote Sens. 6 (6), 4907–4926, 2014

- [26] J. Puc, L. Zust, *On cloud detection with multi-temporal data*. <https://medium.com/sentinel-hub/on-cloud-detection-with-multi-temporal-data-f64f9b8d59e5>, 2019
- [27] B. C. Gao, W. Han, S. C. Tsay, and N. F. Larsen, *Cloud detection over the arctic region using airborne imaging spectrometer data during the daytime*. *Journal of Applied Meteorology*, 37, 11, 1421-1429, 1998
- [28] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. *Data clustering: a review*. *ACM Comput. Surv.* 31, 3, 1999, 264–323.
- [29] W. J. Blackwell, *A neural-network technique for the retrieval of atmospheric temperature and moisture profiles from high spectral resolution sounding data*, in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 11, pp. 2535-2546, Nov. 2005