

Mendelova univerzita v Brně  
Provozně ekonomická fakulta

---

# **Analýza pokrytí území metodami dálkového průzkumu země**

**Diplomová práce**

Vedoucí práce:  
Ing. David Procházka, Ph.D.

Bc. Martin Chodúr

Brno 2015



Tato práce by nikdy nevznikla bez podpory mých rodičů jak při její tvorbě, tak během celého studia. Tímto bych jim rád poděkoval za jejich trpělivost, čas a ochotu. Mé díky také patří Davidovi Procházkovi, který byl mým vedoucím a ochotně mi vždy poradil a ukázal správný směr kudy se vydat. V neposlední řadě také Jiřímu Fejfarovi za odborné konzultace a rady.





### **Čestné prohlášení**

Prohlašuji, že jsem tuto práci: **Analýza pokrytí území metodami dálkového průzkumu země**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 30. prosince 2015

.....



**Abstract**

Chodúr, M. Land use analysis using remote sensing methods. Brno, 2015.

This thesis deals with the use of remote sensing methods for the land use analysis. On the evaluation of already used methods and procedures, own solution that simplifies the whole process at the expense of accuracy of classification is designed and tested. The proposed method offers a different view on the issue and possibilities for its further development.

**Keywords**

Remote sensing, image processing, classification, machine learning, geographic information system, land use.

**Abstrakt**

Chodúr, M. Analýza pokrytí území metodami dálkového průzkumu země. Brno, 2015.

Tato práce se zabývá využitím metod dálkového průzkumu země pro analýzu pokrytí území. Na základě hodnocení využívaných metod a postupů je navrženo a otestováno vlastní řešení zjednodušující celý proces na úkor přesnosti klasifikace. Navržený postup nabízí odlišný pohled na danou problematiku a možnosti jeho dalšího rozvoje.

**Klíčová slova**

Dálkový průzkum země, zpracování obrazu, strojové učení, geografické informační systémy, pokrytí území.



## Obsah

<b>1</b>	<b>Úvod práce</b>	<b>13</b>
<b>2</b>	<b>Cíl práce</b>	<b>14</b>
<b>3</b>	<b>Rešerše</b>	<b>15</b>
3.1	Dálkový průzkum Země a jeho výstupy . . . . .	15
3.2	Metody analýzy pokrytí území . . . . .	17
3.3	Architektury geografických informačních systémů . . . . .	18
3.4	Dostupná softwarová řešení pro metody DPZ . . . . .	19
3.5	Další software pro práci s geografickými daty . . . . .	27
3.6	Projekty zabývající se analýzou pokrytí území . . . . .	28
<b>4</b>	<b>Metodika práce</b>	<b>30</b>
4.1	Stanovení požadavků na použití a výstup . . . . .	30
4.2	Volba architektury systému . . . . .	30
4.3	Návrh uživatelského rozhraní . . . . .	30
4.4	Výběr zdrojových dat . . . . .	31
4.5	Volba klasifikačního algoritmu . . . . .	31
4.6	Výběr technologií pro samotnou implementaci . . . . .	31
4.7	Implementace samotného řešení . . . . .	32
<b>5</b>	<b>Vlastní práce</b>	<b>33</b>
5.1	Volba architektury systému . . . . .	33
5.2	Návrh uživatelského rozhraní . . . . .	34
5.3	Výběr zdrojových dat . . . . .	34
5.4	Volba klasifikačního algoritmu . . . . .	36
5.5	Výběr technologií pro samotnou implementaci . . . . .	38
5.6	Implementace samotného řešení . . . . .	41
<b>6</b>	<b>Diskuze</b>	<b>54</b>
6.1	Zhodnocení výstupů klasifikace . . . . .	54
6.2	Zhodnocení postprocessingu výstupních dat . . . . .	58
6.3	Zhodnocení vhodnosti návrhu . . . . .	60
6.4	Ekonomická zhodnocení navrženého řešení . . . . .	60
<b>7</b>	<b>Závěr</b>	<b>62</b>
<b>8</b>	<b>Reference</b>	<b>63</b>
<b>9</b>	<b>Algoritmus odstranění mostů</b>	<b>67</b>
<b>10</b>	<b>CD</b>	<b>69</b>

## Seznam obrázků

Obrázek 1: Rozdělení elektromagnetického spektra	15
Obrázek 2: Ukázka snímku satelitu Landsat 7 v různých pásmech	16
Obrázek 3: Ilustrace sekvence hodnot pro daný pixel v jednotlivých pásmech snímku.	16
Obrázek 4: Architektura GIS pro oddělení prezentační, aplikační a datové vrstvy	19
Obrázek 5: Ukázka výsledků klasifikace s aplikací nástrojů pro čištění výsledků pomocí <i>ENVI</i>	20
Obrázek 6: Ukázka výsledků klasifikace s aplikací nástrojů pro čištění výsledků pomocí <i>EDRAS</i>	22
Obrázek 7: Ukázka výsledků klasifikace pomocí aplikace <i>QGIS</i> využívající modulů <i>SAGA</i>	23
Obrázek 8: Ukázka výsledků klasifikace pomocí aplikace <i>QGIS</i> využívající modulů <i>SAGA</i>	25
Obrázek 9: Ukázka výsledků klasifikace pomocí <i>GIS GRASS</i>	26
Obrázek 10: Ukázka komunikace ve webové aplikaci	33
Obrázek 11: Ukázka uživatelského rozhraní <i>GIS GRASS</i>	34
Obrázek 12: Diagram funkce <i>WMS</i> serveru	35
Obrázek 13: Ukázka mapových dat služby <i>WMS</i> ČUZK	36
Obrázek 14: Vývojový diagram celého procesu provedení analýzy.	42
Obrázek 15: Ukázka vytvořeného uživatelského rozhraní	43
Obrázek 16: Ukázka rozdílu zeleného pásma (vlevo) a <i>GLI</i> (vpravo) na vodní ploše.	47
Obrázek 17: Ukázka histogramu zeleného pásma	49
Obrázek 18: Ukázka vlivu mediánového filtru na odstranění šumu	50

---

Obrázek 19: Ukázka možných průniků polygonu s bufferem.	52
Obrázek 20: Ukázka vlivu algoritmu pro odstranění mostů	53
Obrázek 21: Ukázka výsledků klasifikace zalesněné oblasti s vodní plochou.	55
Obrázek 22: Ukázka výsledků klasifikace větší oblasti se zemědělskou plochou, vodní nádrží a obydlými oblastmi.	56
Obrázek 23: Ukázka výsledků klasifikace větší oblasti se zemědělskou plochou v různých stavech obdělání a rozsáhlým zalesněným územím.	57
Obrázek 24: Ukázka funkce algoritmu pro odstranění malých polygonů	58
Obrázek 25: Ukázka funkce algoritmu pro rozdělení mostů	59
Obrázek 26: Ukázka funkce algoritmu pro rozdělení mostů	59

## Seznam tabulek

Tabulka 1: Zhodnocení vhodnosti dostupných řešení.	27
Tabulka 2: Data HTTP požadavku.	46
Tabulka 3: Matice záměn pro oblast obrázku č. 21.	55
Tabulka 4: Matice záměn pro oblast obrázku č. 22.	56
Tabulka 5: Matice záměn pro oblast obrázku č. 23.	57





## 1 Úvod práce

Technologie v dnešní době umožňují sběr prostorových dat pokročilými metodami využívajícími zařízení pohybující se ve velkých výškách nad povrchem Země. Obecně se tato oblast získávání dat označuje dálkový průzkum země (DPZ). Analýzou výstupů těchto metod jsme schopni získat užitečné informace o pozorovaných objektech, plochách či jevech, bez nutnosti přímé interakce (Lillesand, T., Kiefer, R. W., Chipman, J., 2008).

Jelikož se většinou jedná o velké objemy dat, je snaha jejich zpracování automatizovat. Výzkum v této oblasti už značně pokročil, i přes to je ale většina postupů velmi specializovaných a nelze tyto metody zcela zobecnit. Jsou tedy závislé na konkrétním cíli dané analýzy.

Velkou část těchto úloh tvoří analýza pokrytí území. Cílem těchto postupů je z nasnímaných dat zjistit co se na nich nachází. Jako příklad může posloužit satelitní snímek, na kterém algoritmus automaticky rozezná různé druhy povrchu například pole, les, vodní plochu a další. Tyto informace mohou sloužit k tvorbě mapových podkladů, výměrám rozloh jednotlivých oblastí nebo studiu jejich změn.

Obvyklým důvodem klasifikace je vytvoření mapových podkladů pro analytickou činnost. Je samozřejmé, že v těchto případech je snaha dosáhnout co nejvyšší přesnosti, protože přesnost klasifikace přímo ovlivňuje kvalitu následných analýz.

Nicméně na celý problém klasifikace se dá podívat ještě z jiného, netradičního, pohledu. Řadu let se řeší detekce objektů v obraze (obvykle fotografiích). Důvody mohou být semi-automatické vymazání nechtěných objektů z fotografií nebo třeba výběr určitého objektu pro interakci (Theodoridis, S., Koutroumbas, K., 2009).

Obdobně se lze podívat na klasifikaci snímků z DPZ. Pro řadu aplikací by mohlo být zajímavé umožnit uživateli vybrat z fotografie jezero, les nebo jeho louku. Snadno by tak o těchto objektech mohl zjistit další užitečné informace — jaké mají rozměry, plochu, obvod atp.

## 2 Cíl práce

Postupů a metod v této oblasti je celá řada a každý se hodí pro jiný druh dat nebo požadovaný výstup. Nedá se tedy zcela jednoznačně říct, že některý z nich je optimální. Zároveň jsou tyto úlohy velmi náročné a pro běžného uživatele těžko použitelné. Jedná se spíše o nástroje vědeckých výzkumů.

Samo obstarání potřebných dat je velký problém. Metody sběru těchto dat jsou velmi nákladné. Většinou se jedná o využití satelitů, letadel nebo v poslední době dronů. Následně je třeba data předzpracovat, než dojde k jejich samotné publikaci. Právě tento aspekt se stal hlavní motivací práce.

Cílem práce je zhodnotit dostupné metody analýzy pokrytí území na základě dat z DPZ. Na základě tohoto průzkumu navrhnout metodu, která bude jednoduše dostupná běžnému uživateli bez nutnosti náročného shánění, a přípravy dat a starostí ohledně samotné analýzy.

Prvním krokem bude rešerše existujících metod a řešení pro klasifikaci obrazových dat pořízených metodami DPZ. V rámci existujících řešení bude kladen důraz na využití svobodného softwaru.

Následně bude provedeno srovnání těchto možností pro řešení daného problému na základě stanovených kritérií. Nejvhodnější z nich bude vybrána pro implementaci vlastního řešení.

Hlavní částí bude implementace, tedy vytvoření metody umožňující automatizovaný proces analýzy pokrytí, za využití parametrizace uživatelem se zaměřením na praktické využití a selekci požadovaných objektů.

Posledním krokem bude zhodnocení této metody a její využitelnosti.

## 3 Rešerše

### 3.1 Dálkový průzkum Země a jeho výstupy

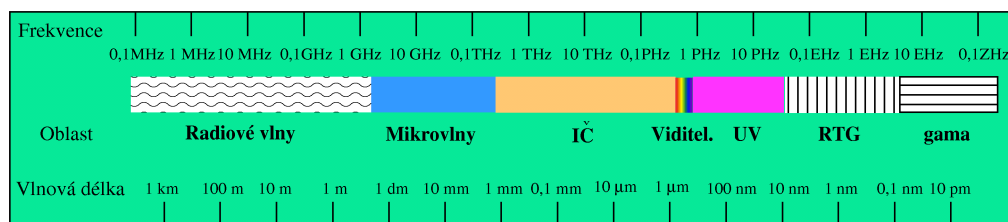
Lillesand definuje Dálkový průzkum Země jako vědu i umění získávat užitečné informace o objektech, plochách či jevech prostřednictvím dat měřených na zařízení, která s těmito zkoumanými objekty, plochami či jevy nejsou v přímém kontaktu (Lillesand, T., Kiefer, R W., Chipman, J., 2008).

Z této definice vyplývá, že do DPZ spadají i samotné analýzy nad získanými daty. V této části se ale budeme zabývat výstupy samotného snímání. Ty budou představovat vstup pro dále hodnocené metody.

Jak již bylo zmíněno výše, pořizování dat DPZ je prováděno nejčastěji pomocí satelitů nebo letadel. Ty jsou speciálně uzpůsobeny pro tuto činnost jak dráhou svého letu, tak technickým vybavením.

Dráha jejich letu musí být přesně definována, aby bylo zaručené pokrytí oblastí snímačem a následné poskládání obrazové mozaiky. Důležitá je také výška, ze které jsou data pořizována, jelikož čím výše je snímač, tím více je výsledek ovlivněn atmosférou. Ta pohlcuje část snímaných dat a snižuje jejich kvalitu. Zároveň jsou data z větší výšky postižena nižším prostorovým rozlišením (Lillesand, T., Kiefer, R W., Chipman, J., 2008).

Technické vybavení těchto zařízení pro DPZ zahrnuje hlavně snímače, pomocí kterých jsou data pořizována. Naprostá většina snímačů pracuje na bázi elektromagnetického vlnění. Jejich hlavní rozdělení je na pasivní a aktivní. Pasivní snímače detekují vlnění vyzářené jiným zdrojem. Aktivní využívají vlastního zdroje, u kterého detekují jeho odraz od snímaných oblastí. Snímačů se často využívá více pro pokrytí většího množství pásme. Nesnímá se tedy pouze viditelné spektrum, ale často se využívá například infračervené a blízké infračervené pásmo nebo mikrovlnné pásmo (Lillesand, T., Kiefer, R W., Chipman, J., 2008).

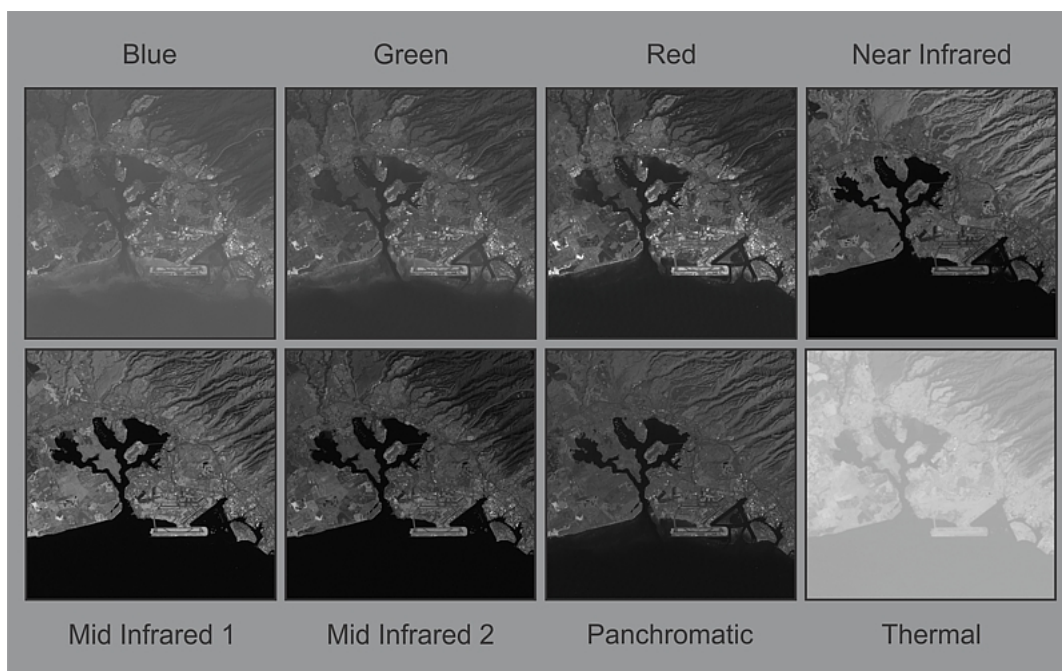


Obrázek 1: Rozdělení elektromagnetického spektra. Zdroj: <https://commons.wikimedia.org/wiki/File:ElmgSpektrum.png>

Pomocí tohoto snímání získáváme rastry pro každé pásmo zvlášť. Jednotlivé pixely nesou hodnotu reprezentující intenzitu vlnění v dané vlnové délce. Oblast, kterou pixel pokrývá, se nazývá prostorové rozlišení a definuje největší detail, který můžeme rozeznat na daném snímku.

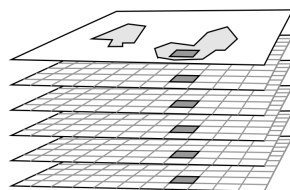
Tyto rastry můžeme vizualizovat pomocí barevné syntézy využívající červené, zelené a modré. Další možností je vytvořit pseudobarevný obraz. Ten se tvoří zamě-

něním některých viditelných pásem za pásma lidským okem neviditelných. To lze využít například k detekci rostlin a vodních ploch pokud nahradíme červené pásmo infračerveným.



Obrázek 2: Ukázka snímku satelitu Landsat 7 v různých pásmech. Zdroj: [http://www.paulillsley.com/Images\\_of\\_the\\_Earth/Composites/Bands.jpg](http://www.paulillsley.com/Images_of_the_Earth/Composites/Bands.jpg)

Vizualizace už je ovšem krok, který pro automatizaci není nezbytně třeba. Tato data nám poskytují počet hodnot pro každý pixel daný počtem pásem, ve kterých byl snímán. Tato datová struktura a cíle dané analýzy se nabízí řešit metodami strojového učení spadajících do oblasti umělé inteligence. Nejsou však jedinou možností. Lze využít mnoho jiných postupů jako například statistické analýzy, objektových klasifikací a dalších (Tibshirani, R., Hastie, T., Friedman, J., 2009).



Obrázek 3: Ilustrace sekvence hodnot pro daný pixel v jednotlivých pásmech snímku.

## 3.2 Metody analýzy pokrytí území

Problém analýzy pokrytí území lze velmi dobře přirovnat ke klasifikačním metodám ve strojovém učení. Jedná se o úlohu rozdělení množiny prvků do stejnorodých skupin na základě jejich příznaků. Pokud toto aplikujeme na rastrová data získaná pomocí DPZ popsaná výše, můžeme prvky reprezentovat jednotlivými pixely a jejich příznaky jednotlivými hodnotami pro daná pásma. To nám umožňuje úlohu řešit pomocí algoritmů specializujících se na tento problém. Ty se rozdělují na tři hlavní oblasti.

### Klasifikace bez učitele

Metody klasifikace bez učitele nepřijímají od uživatele žádná vstupní data kromě počtu tříd, do kterých chce uživatel prvky rozdělit. Fungují na bázi porovnávání vzájemné podobnosti příznaků jednotlivých prvků, na jejichž základě rozhodují, do které třídy prvek patří (Tibshirani, R., Hastie, T., Friedman, J., 2009).

Jejich výstupem není rozdělení dat do konkrétních skupin s daným významem ve vztahu ke vstupním datům. Jedná se pouze o rozdělení dat na daný počet shluků, ve kterých jsou si prvky více podobny než s prvky ostatních shluků. Metoda nám však neřekne, zda se jedná o les, nebo vodní plochu. Jednotlivé algoritmy se liší hlavně v metodách porovnávání podobnosti prvků jednotlivým shlukům.

- Neuronové sítě
- K nejbližších sousedů
- Rozhodovací strom

### Klasifikace s učitelem

Klasifikace s učitelem vyžaduje vstup od učitele. Tím je myšleno, že algoritmus obdrží skupiny prvků, o kterých uživatel ví, že patří do dané třídy. Tyto prvky jsou takzvaně ohodnoceny. To znamená, že každý prvek trénovací množiny má příznak definující do které třídy patří. Na základě těchto vstupních dat se algoritmus učí rozlišovat mezi jednotlivými třídami a následně roztrždit i zbytek neohodnocených dat (Tibshirani, R., Hastie, T., Friedman, J., 2009).

Díky specifikaci daných tříd uživatelem jsou výsledná data již rozdělena na třídy s reálným významem definovaným uživatelem. Pokud tedy uživatel definuje část trénovací množiny jako prvky s příznakem popisujícím les, dostane ve výstupu třídu prvků, které reprezentují les.

- Hierarchické
- SVM
- K-means

## Objektová klasifikace

Jedná se o poměrně novou metodu, která neuvažuje každý prvek samostatně, ale je zaměřená i na jeho okolí. Nevytváří shluky napříč všemi daty, ale pouze vytváří souvislé objekty založené na příznakové a prostorové souvislosti. Výstupem této metody je tedy rozdělení obrazu na souvislé části, které ale nereprezentují stejnorodé oblasti z hlediska pouze příznaků (Theodoridis, S., Koutroumbas, K., 2009). Někdy se používá jako předzpracování obrazových dat a následně se klasifikují až tyto objekty .

## 3.3 Architektury geografických informačních systémů

GIS byly dříve monolitické a závislé na platformě stejně jako většina ostatních systémů. Tomu odpovídala forma desktopových aplikací. V devadesátých letech se na scénu dostal *World Wide Web* a *hypertext transfer protocol*. To umožnilo vývoj velkých komplexních GIS dostupných přes Internet (Agrawal, S., Gupta, R. D., 2014). Prvními kroky byly mapové prohlížecké aplikace a postupně se rozrůstala jejich funkcionality na úrovni provádění analýz na straně serveru a dalších.

Tendence nově vyvíjených systémů, je dodržovat specifikované architektury, díky kterým dochází k abstrakci tří základních vrstev systému. Ten se rozdělí na datovou část, aplikační logiku a prezentační část (Rodríguez Luaces, M. A., 2004). Tím, že se tyto části oddělí, je možné je libovolně nahrazovat a měnit bez vlivu na ostatní. Díky tomuto je možné vyvíjet GIS, které mají více prezentačních vrstev ať už desktopových nebo webových.

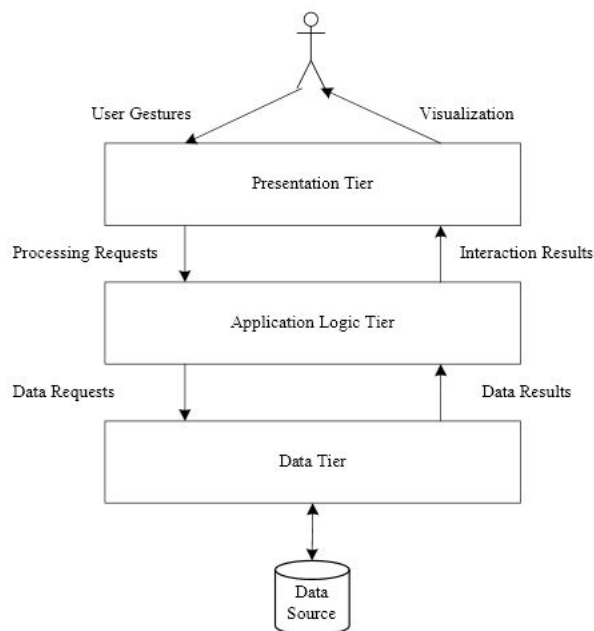
### Desktopové GIS

Desktopové systémy jsou v dnešní době stále komplexnější než většina webových. Jejich zástupce najdeme jak v komerčních řešeních, tak v open source. Mezi omezení patří hlavně závislost na platformě, na které běží, výkon zařízení, na kterém jsou nainstalovány, obtížnější možnosti sdílení dat a další.

Na poli komerční sféry se hlavně jedná o produkty firmy *ESRI*, konkrétně o aplikaci *ArcGIS for Desktop*. *ESRI* už ale také modernizovalo své služby a přešlo převážně na model webových GIS.

*GRASS GIS* je původně projekt americké armády. Nyní se z něj stal velmi obsáhlý nástroj pro široké spektrum analýz a zpracování geografických dat. Je využíván v akademické i komerční sféře (Neteler, Markus a Mitsova, Helena , 2004).

*QGIS* je čistě dobrovolnickým projektem. Jeho největší výhodou je možnost implementace vlastních zásuvných modulů a jejich následná publikace. Umožňuje také využití funkcionality valné většiny open source GIS jako například i *GRASS GIS* (QGIS Development Team, 2014).



Obrázek 4: Architektura GIS pro oddělení prezentační, aplikační a datové vrstvy. Zdroj: <http://lbd.udc.es/Repository/Thesis/432677332J.pdf>

## Webové GIS

V této oblasti je možné využít libovolných webových technologií pro tvorbu webových GIS. Jedním z hlavních směrů je ale využití jazyka *JavaScript*. Tento jazyk je interpretován přímo v prohlížeči na straně klienta.

Pro usnadnění tvorby webových GIS v tomto jazyce vzniklo několik knihoven. Ty umožňují rychlé definování mapových oken, správu vrstev, tvorbu geometrií a mnoho dalších úkonů.

Hlavním zástupcem v této oblasti na poli open source je knihovna *Open Layers*. Jedná se o čistě *JavaScript* knihovnu bez jakékoli serverové části. Momentálně se jedná o pravděpodobně nejvyspělejší open source knihovnu pro webové mapové aplikace. (Santiago, A., 2015)

Další možností z open source sféry je *JavaScript* knihovna *Leaflet*. Je rychle se rozrůstajícím projektem, ale prozatím nedosahuje takových možností jako *Open Layers*. (Wikipedia, 2015)

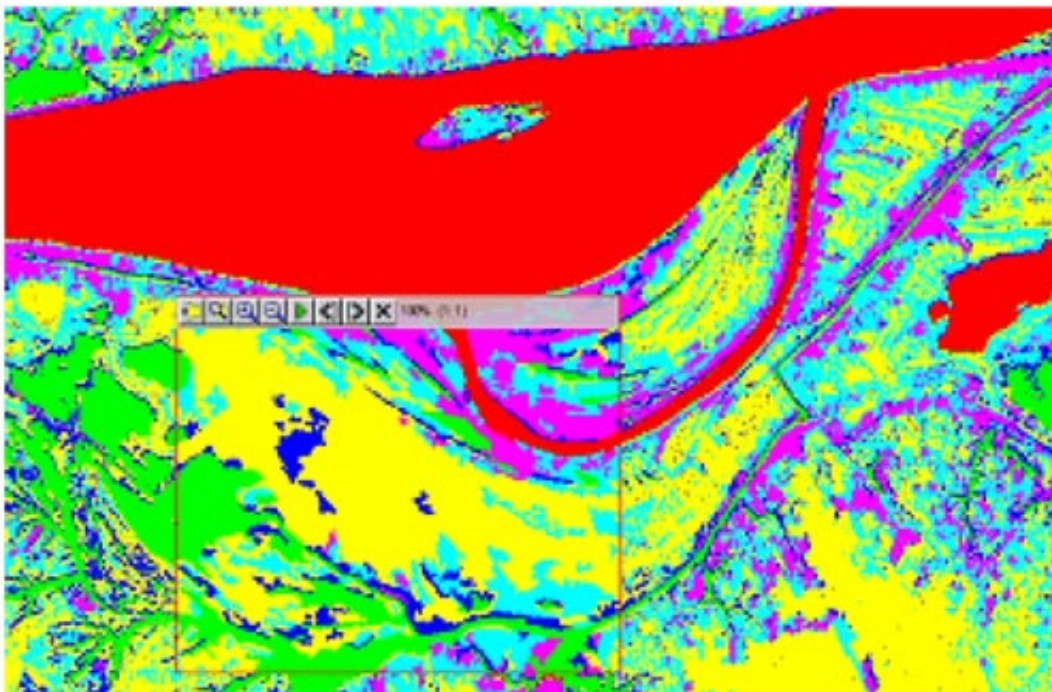
## 3.4 Dostupná softwarová řešení pro metody DPZ

### ENVI

Jedná se o komerční desktopové řešení zaměřující se na celkový proces zpracování obrazových dat DPZ a jejich následnou integraci do geodatabází (Exelis Visual Information Solutions, 2013). Zahrnuje nástroje pro detekci změn, klasifikaci, ortho-

rektifikaci a extrakci objektů. Všechny tyto nástroje je možné provázat a vytvářet postup jejich aplikací a parametrizací. Součástí jsou také nástroje na zpracování výsledků zahrnující tvorbu mapových výstupů a integrace se systémem *ArcGIS*.

Co se týče podporovaných datových formátů, umožňuje *ENVI* import z většiny rozšířených datových formátů jako například *HDF*, *GeoTIFF*, *JPEG* a *JPEG2000*. Zajímavou funkcionalitou je také možnost načítání dat pomocí *OGC* služeb.



Obrázek 5: Ukázka výsledků klasifikace s aplikací nástrojů pro čištění výsledků pomocí *ENVI*. Zdroj: [http://www.exelisvis.com/portals/0/images/wp\\_ENVIGIS\\_img6.jpg](http://www.exelisvis.com/portals/0/images/wp_ENVIGIS_img6.jpg)

V rámci klasifikace umožňuje využití algoritmů pro učení s i bez učitele. Uživateli dává možnost zvolit použitý algoritmus a parametrizovat ho.

Pro klasifikaci s učitelem nabízí tyto algoritmy:

- Parallelepiped
- N Mahalanobis Distance
- Maximum Likelihood
- Minimum Distance
- Spectral Angle Mapper (SAM)
- Spectral Information Divergence (SID)
- Support Vector Machine
- Binary Encoding
- Neural Net
- Winner Takes All



Pro klasifikaci bez učitele jsou dostupné následující algoritmy.

- K-Means
- ISODATA

Dále *ENVI* nabízí další krok kterým je čištění dat po proběhlé klasifikaci. V této části dochází k odstranění „ostrovů“ a dalších nechtěných jevů.

Jednotlivé moduly a nástroje *ENVI* neposkytují žádné programové rozhraní umožňující jejich integraci do vlastních automatizovaných procesů.

## ERDAS IMAGINE

*EDRAS IMAGINE* je velmi komplexní softwarové řešení pro zpracování, analýzu a publikaci geografických dat. Umožňuje zpracování rastrových i vektorových dat. Také má velmi dobrou podporu zpracování 3D dat a jejich vizualizaci. Umožňuje definování jak vlastních posloupností zpracování v rámci aplikace, tak poskytuje programové rozhraní pro využití v uživatelské automatizaci (HEXAGON, 2015).

Pro klasifikaci nabízí algoritmy s učitelem s využitím paralelizace, což má výrazný dopad na rychlost prováděných analýz.

Pro klasifikaci s učitelem nabízí tyto algoritmy.

- Parallelepiped classifier
- Maximum likelihood method
- Minimum distances method

(Harikrishnan, Poongodi, 2015)

Pro klasifikaci bez učitele jsou dostupné následující algoritmy.

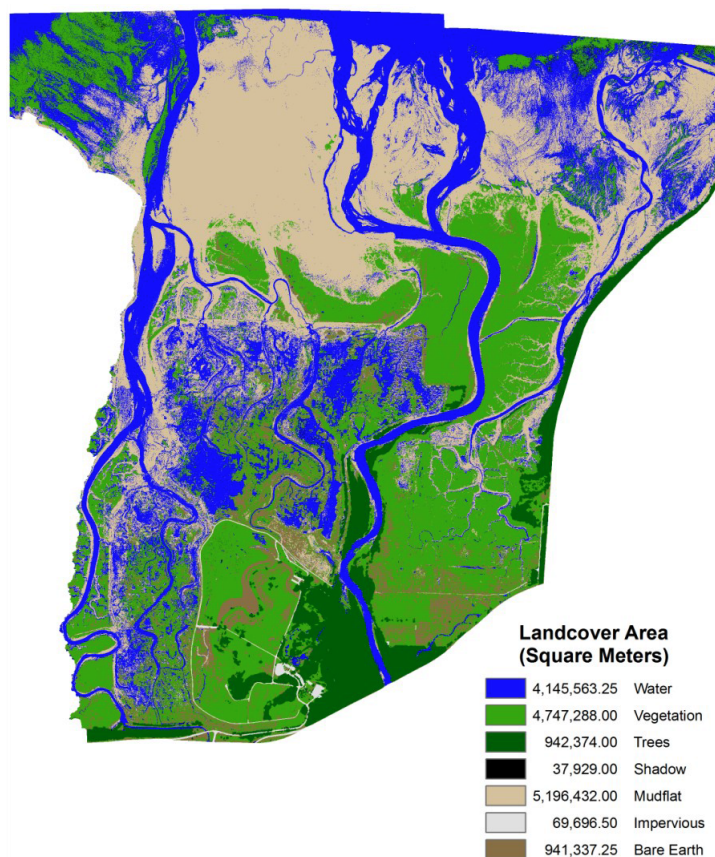
- K-Means
- ISODATA
- Fuzzy

(Harikrishnan, Poongodi, 2015)

U těchto algoritmů je dále možné volit hodnotící kritéria pro klasifikaci a optimalizační metody pro volbu parametrů. Celkové množství možností, jak optimalizovat proces klasifikace, je nepřeberné a umožňuje uživateli maximálně přizpůsobit datům a výstupům.

Dále umožňuje využití velkého množství metod pro vyhodnocení přesnosti a úspěšnosti výstupů, definování tříd po klasifikaci, objektovou klasifikaci a mnoho dalších.

Velkou výhodou je také již zmíněné programové rozhraní zaměřené na jazyk *Python*. Díky tomu je možné využít funkcionality *EDRAS IMAGINE* ve vlastních



Obrázek 6: Ukázka výsledků klasifikace s aplikací nástrojů pro čištění výsledků pomocí EDRAS. Zdroj: [http://www.tidalmarshmonitoring.org/pdf/USGS\\_WERC\\_ERDAS\\_Habitat\\_Classification\\_SOP.pdf](http://www.tidalmarshmonitoring.org/pdf/USGS_WERC_ERDAS_Habitat_Classification_SOP.pdf), s. 13

procesech. Další výhodou je možnost vytvářet serverové řešení přímo jedním z nástrojů. To umožňuje provádění těchto analýz přes síť, bez nutnosti instalace aplikace na straně uživatele a zatížení jeho výpočetních kapacit.

## SAGA GIS

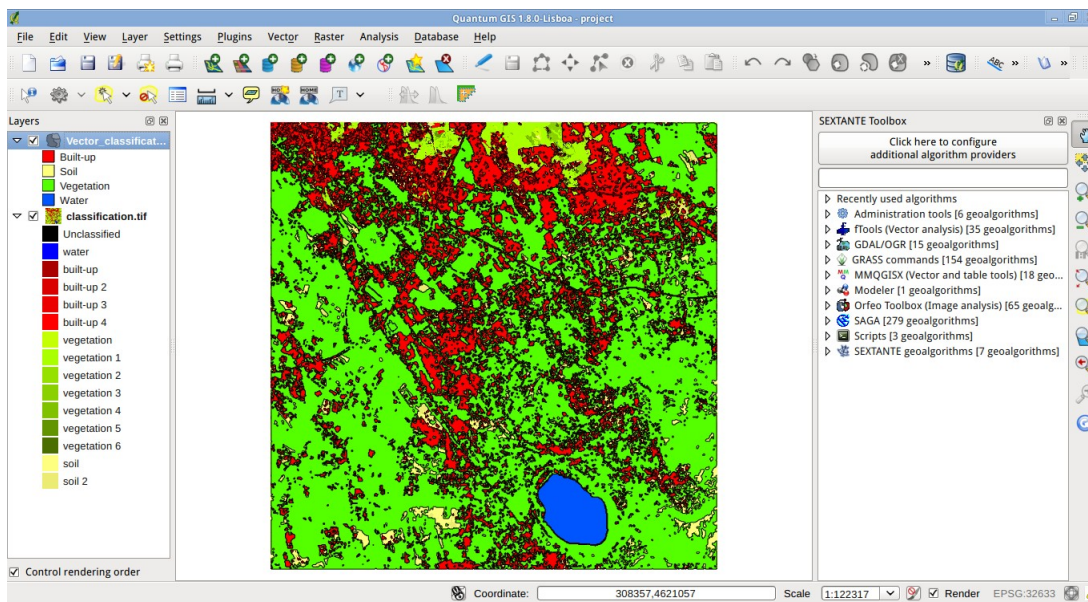
*System for Automated Geoscientific Analyses*<sup>1</sup> (SAGA) byl navržen pro jednoduchou a efektivní implementaci prostorových algoritmů. Slouží tedy jako platforma pro vývoj a implementaci analýz a metod pro zpracování nad geografických dat. (Conrad, O., 2007)

V dnešní době tento software obsahuje přes 600 metod pokrývajících celé spektrum geografických analýz. Je vydán pod licencí *GNU General Public License (GPL)*. Umožňuje zpracování jak rastrových, tak vektorových dat, převody projekcí a práci s topologiemi. Jádro tvoří jednotlivé nástroje, které jsou rozděleny do modulů a knihoven. Veškerá funkcionalita je dostupná pomocí aplikačně programového

<sup>1</sup>Systém pro Automatizaci Geografických Analýz

rozhraní(API). Jedním z modulů je také grafické rozhraní zpřístupňující veškerou funkcionalitu uživateli.

(Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V. a Böhner, J., 2015)



Obrázek 7: Ukázka výsledků klasifikace pomocí aplikace QGIS využívající modulů SAGA. Zdroj: <http://2.bp.blogspot.com/-8ah6hm4KDCo/Ucxx10PuDvI/AAAAAAAAAQo/2Rd-HeCYI0g/s1366/vector.jpg>

Pro účel analýzy dat DPZ nabízí *SAGA* mnoho nástrojů jak pro předzpracování a přípravu dat, tak pro samotnou klasifikaci.

Pro klasifikaci s učitelem nabízí tyto algoritmy.

- Binary Encoding
- Parallelepiped
- Minimum Distance
- Mahalanobis Distance
- Maximum likelihood
- Spectral Angle Mapping
- Winner Takes All

(Conrad, O., 2005)

Pro klasifikaci bez učitele jsou dostupné následující algoritmy.

- Hill-Climbing
- Iterative Minimum Distance
- Combined Minimum Distance / Hill Climbing

(Conrad, O., 2001)

Velkou výhodou je zmíněný důraz na API. To znamená využití rozhraní pro komunikaci s jiným softwarem. To umožňuje zahrnout tyto nástroje v rámci vlastního vývoje a integrace s jinými knihovny a nebo pouze automatizaci procesů *SAGA* v dané posloupnosti.

### Orfeo ToolBox

*Orfeo* je knihovna napsaná v jazyce *C++* pro zpracování dat DPZ publikovaná pod licencí *CeCILL-v2*. Využívá řadu knihoven třetích stran jako například *GDAL*, pro správu prostových dat, a *OpenCV*, pro strojové učení (CNES, 2002). Podporuje většinu využívaných formátů pro publikaci prostorových dat a jejich předzpracování. Mezi hlavní oblasti analýz, na které se zaměřuje, patří extrakce příznaků, obrazová segmentace a klasifikace. Má však mnoho nástrojů i pro další úlohy DPZ.

Pro klasifikaci s učitelem nabízí tyto algoritmy.

- Support Vector Machine
- Bayes
- Boost
- Decision Tree
- Random Forests
- Gradient Boosted Tree
- K-Nearest Neighbors
- Artificial Neural Network

(OTB Development Team, 2015)

Pro klasifikaci bez učitele jsou dostupné následující algoritmy.

- K-Means
- Kohonen's Self Organizing Map
- Bayesian Plug-In Classifier
- Expectation Maximization Mixture Model Estimation
- Statistical Segmentations
- Classification using Markov Random Fields

(OTB Development Team, 2015)

Jelikož se jedná výhradně o programovou knihovnu, nemá v základní výbavě žádné uživatelské rozhraní. Hovoříme tedy opravdu pouze o sadě nástrojů, jejichž

využití už záleží pouze na vývojáři, jak ho integruje do svého řešení. *Orfeo* má implementované vazby také na jazyk *Java* a *Python*. Díky tomuto je možné její široké využití.



Obrázek 8: Ukázka výsledků klasifikace pomocí aplikace QGIS využívající modulů SAGA. Zdroj: <http://2010.foss4g.org/presentations/3655.pdf>, s. 15

## GRASS GIS

*GRASS* (Geographical Resources Analysis Support System<sup>2</sup>) je geoinformační systém pro práci s rastrovými a vektorovými daty kombinovaný s integrovanými nástroji pro zpracování obrazu a vizualizaci dat. Zahrnuje přes 350 modulů pro správu, zpracování, analýzu a vizualizaci georeferencovaných dat (Neteler, Markus a Mitsova, Helena, 2004). Je dostupný pod licencí *GNU GPL*<sup>3</sup> což umožňuje jeho volné využití.

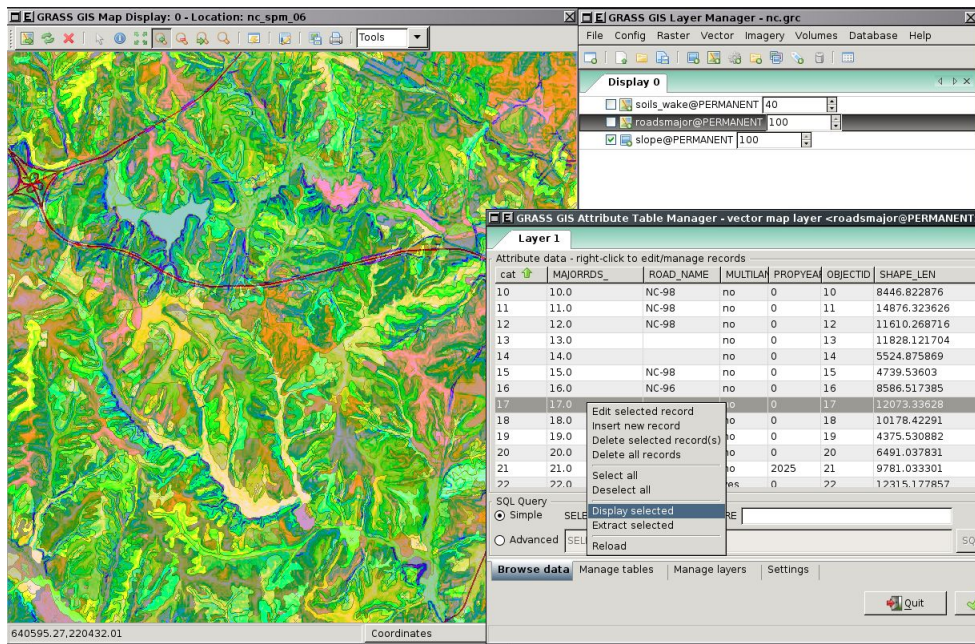
Jedná se o kompletní nástroj s grafickým rozhraním pokrývající jak import dat, tak jejich správu, předzpracování, analýzu a následnou vizualizaci a publikaci. Je však možné využít jeho jednotlivé moduly a algoritmy samostatně. Pro klasifikaci nabízí následující algoritmy.

---

<sup>2</sup>Systém pro podporu analýz geografických zdrojů

<sup>3</sup>General Public License





Obrázek 9: Ukázka výsledků klasifikace pomocí GIS GRASS. Zdroj: [https://grass.osgeo.org/uploads/images/Gallery/vector/grass6\\_wxgui-attrib\\_manager.png](https://grass.osgeo.org/uploads/images/Gallery/vector/grass6_wxgui-attrib_manager.png)

Pro klasifikaci s učitelem nabízí tyto algoritmy.

- Equal interval
- Quantiles
- Standard deviations
- Equiprobabilites algorithm
- Discont

Pro klasifikaci bez učitele jsou dostupné následující algoritmy.

- Modifikace k-means
- Maximum-likelihood

## Zhodnocení vhodnosti dostupných řešení

Tabulka 1: Zhodnocení vhodnosti dostupných řešení.

Nástroj	open source	API	Preprocessing	Klasifikace	Postprocessing
ENVI	Ne	Ne	Ano	Ano	Ano
ERDAS IMAGINE	Ne	Ano	Ano	Ano	Ne
SAGA GIS	Ano	Ano	Ano	Ano	Ano
Orfeo ToolBox	Ano	Ano	Ano	Ano	Ne
GRASS GIS	Ano	Ano	Ano	Ano	Ano

Ze závěrečného zhodnocení vychází nejlépe *GRASS GIS* a *SAGA GIS*. Oba tyto nástroje obsahují algoritmy pro preprocessing, klasifikaci a postprocessing. Zároveň také umožňují jejich integraci do vlastních řešení za využití *API* a jedná se o open source projekty umožňující jejich bezplatné využití.

## 3.5 Další software pro práci s geografickými daty

V oblasti GIS se využívá řada dalších nástrojů pro práci s geografickými daty a jejich zpracování. Níže jsou uvedeny nejvýznamnější open source aplikace.

### GDAL

*GDAL*<sup>4</sup> je knihovna napsaná v jazyce *C++*. Slouží pro čtení a zápis různých geografických rastrových a vektorových formátů. Nad těmi vytváří jednoduchý abstraktní model umožňující jejich další zpracování. Nabízí rozhraní pro jazyky *Python*, *Java*, *Perl*, *Ruby* a *C#*. Jedná se o jeden z nejdůležitějších projektů v oblasti open source GIS. Využívá ho většina ostatních aplikací například *GRASS GIS*, *QGIS*, *ESRI ArcGIS*, *Google Earth* a mnoho dalších. (Landa, M., 2015)

### PROJ.4

*PROJ.4* je knihovna pro práci s kartografickými zobrazeními implementovaná v jazyce *C*. Je využívána knihovnou *GDAL* a řadou dalších aplikací. (Landa, M., 2015)

<sup>4</sup>Geospatial Data Abstraction Library

## PostGIS

*PostGIS* je rozšíření relačního databázového systému *PostgreSQL*. Implementuje specifikaci *Simple Features*<sup>5</sup> od OGC a je plnohodnotnou open source náhradou geodatabázím jako například *Oracle Spatial* (Hsu, L S., Obe, R O., 2014). Umožňuje jeho využití pro databázovou persistenci dat i jejich zpracování. Obsahuje procedury pro jejich zpracování a analýzu případně i rozšíření o topologie a grafové algoritmy. Převážně se využívá pro práci s vektorovými daty, ale v poslední době se rozšiřuje podpora i rastrových dat.

### 3.6 Projekty zabývající se analýzou pokrytí území

V této části jsou popsány ukázkové přístupy a využití metody projektů, které se zabývají konkrétními případy analýzy pokrytí území. V této oblasti se většinou nejedná o obecnou analýzu pokrytí, ale zaměřenou na některý konkrétní druh pokrytí. Modely jsou specificky parametrizovány na daný druh, velmi specializované a mají přesně definované požadavky na přesnost výstupu.

#### Czech Terra

Projekt *Czech Terra* se zabývá multizdrojovým celorepublikovým statistickým šetřením. Využívá kombinaci analýzy leteckých snímků a pozemních šetření pro obstarání informací o pokrytí území. Celá plocha České Republiky je rozdělena do 1599 čtverců o rozloze  $7 \times 7$  km. V každém tomto čtverci je náhodně umístěn menší čtverec o rozloze  $450 \times 450$  m, tzv. lokalita. Na území každé této lokality je provedena klasifikace leteckých snímků. Pro tu je oblast rozdělena na čtverce o velikosti  $10 \times 10$  m, které jsou následně klasifikovány do jednotlivých pokrývných tříd. Ve středu lokality je umístěna jedna kruhová vzorkovací plocha o výměře 0,05 ha, tzv. inventarizační plocha. Šetření na inventarizačních plochách je realizováno v terénu tehdy, pokud je na nich identifikován les nebo zeleň mimo les se stromovou vegetací. (Černý, M., 2009)

#### Vyhodnocení odumírání horského smrkového lesa na Trojmezí (NP Šumava) metodou automatizované klasifikace leteckých snímků

Cílem tohoto projektu je vyhodnocení rozsahu odumírání lesa v důsledku žíru lýkožrouta smrkového pomocí metody automatizované klasifikace časové řady leteckých snímků. Pro klasifikaci bylo využito červené, zelené, modré a blízké infračervené pásmo. Klasifikaci předcházela objektová klasifikace. Ta rozděluje data do stejnorodých ploch, které jsou teprve po té klasifikovány do jednotlivých tříd na základě vlastností agregovaných ze vzorků, které do dané oblasti spadají. (Hájek, F., Svoboda, M., 2007)

---

<sup>5</sup>Standard, který specifikuje uložení geografických dat v digitální podobě.



### **DPZ pro monitoring luční vegetace a managementových zásahů v Krkonoších**

Iniciativa pro klasifikaci luční vegetace v modelovém území v Krkonoších, rozkládajícím se SV od města Vrchlabí na ploše asi 80 km<sup>2</sup>. Využívá metod DPZ pro rychlou kontrolu způsobu managementu lučních enkláv. K tomu využívá družicových dat satelitu *WorldView-2*. Cílové třídy pro klasifikaci jsou pokosené louky, spasené louky, louky s dominantní smilkou tuhou, degradované louky, louky s dominantními brusnicovitými keříčky a vřesem obecným, louky s dominantním trojštětem žlutavým, srhou říznačkou nebo psárkou luční, podmáčené porosty a cenná antropogenní společenstva. Trénovací data pro klasifikaci byla určena v terénu botanikem, který determinoval 65 exemplárních vzorků pro tyto třídy. Pro klasifikaci bylo využito algoritmů *Maximum likelihood* a neuronových sítí. (Pomahačová, M., 2012)

U těchto projektů lze pozorovat rozdílnost využitých metod a postupů. V případě Czech Terra se jedná o velmi komplexní statistické vyhodnocování celorepublikového pokrytí, kde metody *DPZ* slouží jen jako jedny ze dvou hlavních metod pro určení pokrytí. Dále se využívá pozemního průzkumu a následně jsou tato data statisticky vyhodnocována. U vyhodnocení odumírání horského smrkového lesa je zase využito objektové klasifikace, která předchází samotnému rozdělení do požadovaných tříd. Poslední příklad monitoring luční vegetace využívá klasického přístupu klasifikace s učitelem za využití satelitních snímků a trénovacích dat obstaraných v terénu.

## 4 Metodika práce

### 4.1 Stanovení požadavků na použití a výstup

Snahou bude vytvořit aplikaci umožňující jednoduchou analýzu pokrytí nad volně dostupnými daty, kterou zvládne i uživatel, který nebude problematice rozumět. Metoda by měla být jednoduše parametrizovatelná, aby měl uživatel možnost do určité míry ovlivnit výsledek analýzy.

Cílový uživatel tedy nebude mít představu jak aplikace funguje a jediný jeho požadavek bude rozdělení dané oblasti na jednotlivé druhy pokrytí. Dalším případem je snaha získat pouze určitou oblast ze zdrojových dat pro další zpracování.

Zdrojová data by měla být jednoduše volitelná. Vzhledem k tomu, že uživateli půjde více o výsledná data než o zdrojová, bude jednodušší, aby byla data předem v aplikaci připravena k použití.

Výsledkem analýzy by měla být informace o plochách různých druhů povrchu. K tomuto bude nejvhodnější georeferencovaná vektorová forma dat v jednoduše zpracovatelném standardizovaném formátu. Ta bude možné exportovat pro další využití.

Přesnost výstupních dat o pokrytí území nebude požadována taková, jako u profesionálních analýz. Bude se jednat spíše o orientační výstup s předpokladem, že pro vyšší přesnost by bylo třeba provést odbornou analýzu, která je mimo schopnosti uživatele. Důraz bude spíše kladen na očištění výsledku o nechtěné artefakty klasifikace pocházející z různorodosti a chyb vstupních dat.

### 4.2 Volba architektury systému

Na základě stanovených požadavků bude zhodnocena vhodnost architektur stávajících řešení pro tuto implementaci. Od nich se bude odvíjet návrh vlastní architektury. Cílem je dosáhnout co nejjednoduššího řešení pro uživatele, což bude hlavním hlediskem.

Hlavním rozhodnutím bude volba mezi desktopovou a webovou aplikací. Každá z těchto možností nabízí řadu přístupů a možností návrhu. Měl by být kladen důraz na jednoduchost ovládání a spouštění aplikace uživatelem, nezávislost na platformě a nároky na hardware.

### 4.3 Návrh uživatelského rozhraní

S ohledem na jednoduchost ovládání bude navrženo uživatelské rozhraní, které by mělo být co nejvíce intuitivní a nevyžadovalo větší zkušenosti ze strany uživatele. Mělo by splňovat veškeré požadavky na funkcionalitu navrženého řešení. Ty obsahují zobrazení mapy s podkladovými daty a definici oblastí nad těmito daty. Dále by mělo být schopno vykreslit do mapového okna výsledná data a případně umožnit nastavení jejich vizuální podoby pro jednodušší vyhodnocení výsledků. Poslední

důležitou funkcionalitou by měly být doplňkové prvky pro parametrizaci analýzy a export výsledných dat pro další zpracování.

## 4.4 Výběr zdrojových dat

Dalším krokem bude výběr zdrojových dat a formy jejich obstarání. Toto je další obtížnou částí v rámci procesu analýzy pro uživatele. Z toho důvodu je třeba této části procesu přidělit vysokou prioritu.

Datových zdrojů DPZ je nepřehledné množství, ale většina je zpoplatněna a nebo velmi obtížně dostupná. Pokud se je podaří obstarat, tak většinou v rastrových formátech, jako je například *GeoTIFF*. S těmi jsou zvyklí pracovat odborníci v dané oblasti, ale nejedná se o běžné formáty. Ty je dále třeba spojovat do souvislých ploch a nebo pracovat pouze s danou oblastí. Všechny tyto aspekty by měly být zhodnoceny a na základě tohoto vyhodnocení navržena metoda, která bude splňovat nastavené požadavky.

## 4.5 Volba klasifikačního algoritmu

Volba algoritmu pro samotnou klasifikaci bude značně závislá na vybraných zdrojových datech a požadovaném výstupu. Bude třeba otestovat jejich vhodnost a výstupy.

Hlavním rozhodnutím bude volba algoritmů učících se s nebo bez učitele. K tomu budou zásadní právě charakteristiky dat. Bude třeba hodnotit jejich stejnorodost, pravděpodobnostní rozdělení hodnot a celkový vliv ovládání aplikace spolu s uživatelskými vlivy.

Až bude rozhodnuto, které z těchto tříd bude využito, budou testovány konkrétní algoritmy a rozhodnuto o jejich vhodnosti a využití v experimentální implementaci.

## 4.6 Výběr technologií pro samotnou implementaci

V této fázi už bude jasné jaké jsou požadavky, co je potřeba vytvořit a jak by měl systém fungovat. Pro jeho samotnou implementaci je nutné zvolit vhodné technologie, pomocí kterých bude provedena.

V této části by měly být voleny optimální technologie, bohužel jsem omezen svými znalostmi, takže pokud bude na výběr, budou preferovány technologie, se kterými už mám dřívější zkušenosti. Hlavním cílem není rychlostní ani systémová optimalizace, takže by tento ústupek neměl mít vliv na celkovou myšlenku práce.

Bude potřeba zvolit technologie jak ze strany hardware, tak software. Volba hardware, na kterém poběží samotná aplikace, bude závislá na zvolené architektuře a finančních možnostech. Co se týče software, tak volba technologií bude taktéž závislá na architektuře, ale primárně bude třeba zvolit vhodné programovací jazyky, schopné spojit veškerou funkcionalitu dohromady, navrhnout komunikaci v rámci

aplikace a pokud to bude možné, tak využít knihoven třetích stran pro ulehčení vytvoření systému.

## 4.7 Implementace samotného řešení

Tento krok bude obsahovat samotnou implementaci experimentálního systému odpovídajícího předešlým specifikacím. Zároveň zde bude uvedeno jak byly využity zvolené technologie pro dosažení požadovaného výstupu.

Implementace bude zahrnovat jak konfiguraci a nastavení prostředí pro běh systému na hardware a zajištění komunikace, tak realizaci samotné architektury a tvorbu jejích jednotlivých částí. Ty bude probíhat pomocí dříve zvolených technologií.

V případě, že se objeví nějaké závažné chyby nebo nedostatky návrhu, budou zde podrobně rozebrány spolu s popisem jejich vyřešení.

## 5 Vlastní práce

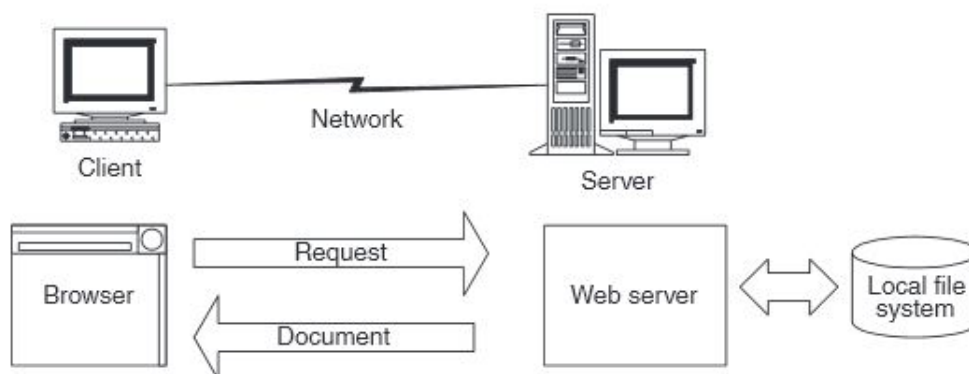
### 5.1 Volba architektury systému

V poslední době lze pozorovat tendence upouštění od desktopových aplikací a přechodu na webové aplikace. Tento přístup ulehčuje práci uživatelům co se týče závislosti na konkrétní platformě, instalace aplikace, aktualizací a mnoha dalších problémech, které se v této oblasti vyskytují.

V oblasti geoinformačních systémů se ještě donedávna pracovalo převážně s robustními desktopovými aplikacemi, od čehož se poslední dobou upouští právě směrem k webovým.

S ohledem na cílenou jednoduchost aplikace bude experimentální implementace realizována jako webová aplikace, kde analýza bude probíhat na straně serveru a na straně uživatele dojde pouze k definici vstupních dat a parametrizaci metody. Po jejím proběhnutí bude ze serveru výsledek vrácen uživateli. Takovýto návrh zbavuje uživatele nutnosti instalace aplikace a možnosti jejího spuštění v běžném internetovém prohlížeči.

Webové aplikace jsou založeny na komunikaci klienta a serveru převážně pomocí protokolu *HTTP*<sup>6</sup>. Ten přesně definuje formu a náležitosti této komunikace. (Conallen, J., 2003)



Obrázek 10: Ukázka komunikace ve webové aplikaci. Zdroj: <http://www.pearsonhighered.com/samplechapter/0201730383.pdf>

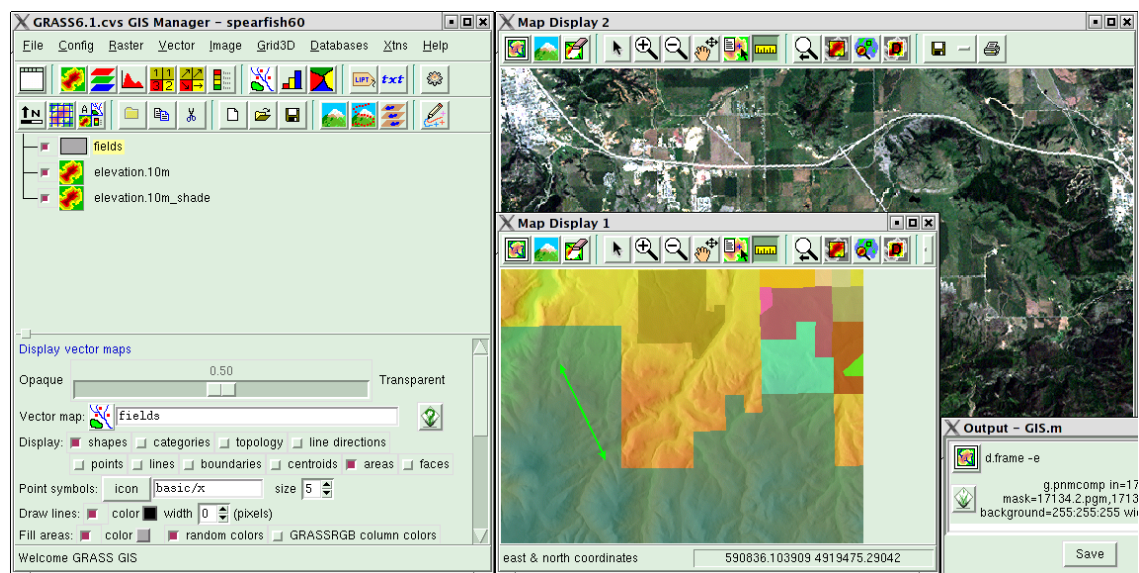
Tímto oddělením uživatelské části (klient) a serverové bude možné kdykoli vytvořit jiné uživatelské rozhraní a nebo využít metodu v cizím řešení pouhým voláním serverové strany.

<sup>6</sup>HyperText Transfer Protocol

## 5.2 Návrh uživatelského rozhraní

Uživatelské rozhraní bude z pohledu jednoduchosti jednou ze stěžejních částí.

Při pohledu na uživatelská rozhraní dostupných systémů, zjistíme, že se jedná o velmi komplexní nástroje. Pokrývají často mnoho funkcionality nepotřebné v tomto postupu pro zajištění variability jejich využití. To je samozřejmě vhodné pro zkušeného uživatele, ale to není tento případ.



Obrázek 11: Ukázka uživatelského rozhraní GIS GRASS. Zdroj: [https://grass.osgeo.org/grass61/screenshots/images/grass61\\_gismanager.png](https://grass.osgeo.org/grass61/screenshots/images/grass61_gismanager.png)

Je tedy třeba vytvořit vlastní, které bude splňovat požadavky na jednoduchost. To bude obsahovat pouze ovládací prvky nezbytné pro spuštění analýzy a její parametrizaci.

Základním prvkem tohoto rozhraní by měla být mapa, pomocí které uživatel definuje oblast zdrojových dat, nad kterou analýza proběhne. V této mapě by měl uživatel být schopen zadat oblasti definující požadované třídy pokrytí. Výstupní data by se měla zobrazit opět v mapě, kde je bude možné vizualizovat pro zhodnocení výsledku analýzy.

## 5.3 Výběr zdrojových dat

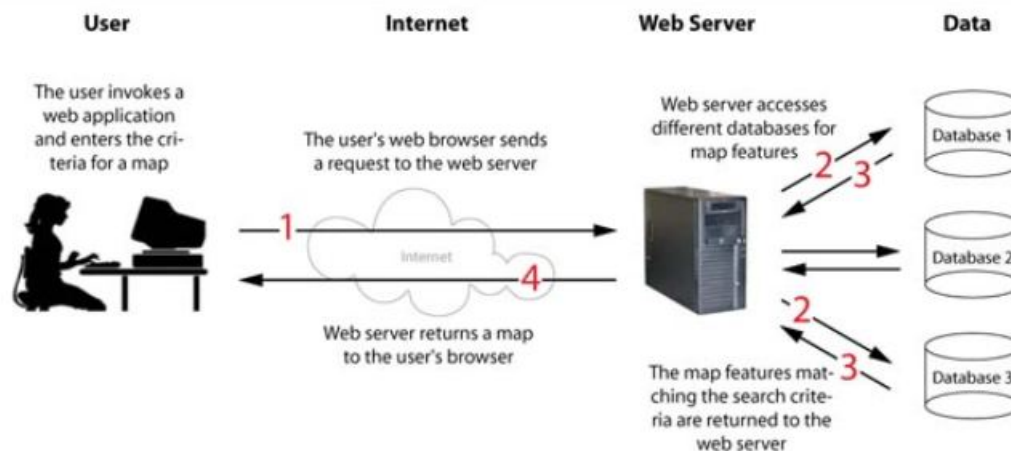
Jak už bylo zmíněno v předchozích částech, obstarání dat DPZ je samo o sobě poměrně velmi náročné. Všechny řešení počítají s tím, že si uživatel data obstará sám. V tomto případě, by ale mělo jít o opravdu jednoduché řešení, které nebude od uživatele vyžadovat jakoukoli znalost o zdrojových datech a jak je obstarat.

Řešením je tedy využít již zpracovaná a publikovaná data poskytovaná v jednoduše dostupné formě. Zřejmě nejrozšířenější jsou v této formě orthofoto snímky poskytované formou standardů *WMS*<sup>7</sup> od *Open Geospatial Consortium*.

### Web Map Service

Jedná se o standard pro vizualizaci geografických dat přes Internet. Umožňuje publikaci celých map, nebo pouze vrstev v rastrovém formátu. Operace mohou být vyvolány odesláním požadavku klasickým webovým prohlížečem ve formě *URL*<sup>8</sup>. (Stopper, R., 2013)

Mapový server, který je poskytuje, zasílá klientovi vyrenderovaný obrázek nebo sadu dlaždic pouze pro požadovaný výřez mapového okna. Data jsou georeferencovaná a obsahují i metadata jako geografické rozlišení a další.



Obrázek 12: Diagram funkce WMS serveru. Zdroj: [http://www.e-cartouche.ch/content\\_reg/cartouche/webservice/en/text/webservice.pdf](http://www.e-cartouche.ch/content_reg/cartouche/webservice/en/text/webservice.pdf)

Nevýhodou těchto dat je, že obsahují pouze tři pásma pro červenou, zelenou a modrou. To eliminuje možnost využití dalších pásem, ale v rámci zjednodušení pro uživatele, je v tomto směru větším přínosem.

Naopak velkou výhodou je, že v této formě je možné získat data pokrytí celé Země i v různých úrovních přiblížení. To umožňuje využití aplikace pro libovolnou oblast a měřítko.

### Konkrétní data

Již zmíněná *WMS* je pouze formou poskytování mapových podkladů přes Internet. S ohledem na to, že cílem práce není vlastní příprava dat nebo jejich publikace, bude využito některé z veřejně dostupných mapových služeb.

<sup>7</sup>Web Map Service

<sup>8</sup>Uniform Resource Locator

Bylo by samozřejmě možné vytvořit vlastní mapový server, na kterém by šlo publikovat libovolná data dle své potřeby. Tím by mohla aplikace nabídnout více druhů zdrojových dat pro analýzu.

V této aplikaci bude využita mapová služba ČUZK<sup>9</sup>. Ta poskytuje ortofoto pro oblast celé České republiky ve formě *WMS*.



Obrázek 13: Ukázka mapových dat služby WMS ČUZK. Zdroj: <http://geoportal.cuzk.cz/geoprohlizec/?wmcid=12232>

## 5.4 Volba klasifikačního algoritmu

Na základě stanovených vstupních dat můžeme zhodnotit vhodnost jednotlivých přístupů metod strojového učení a klasifikačních algoritmů.

### Učení s nebo bez učitele

Prvním krokem je rozhodnout mezi učením s učitelem a bez učitele. Využívejme dále označení *klasifikace* pro algoritmy učící se s učitelem a *shlukování* pro algoritmy s učením bez učitele.

### S učitelem

Jak bylo zmíněno v úvodu práce, výstupem klasifikace je rozdělení vstupních dat do takzvaných tříd. Ty by měly být definovány na základě analyzovaných dat a reprezentovat skupiny, do kterých se mají data rozdělit. Předpokladem pro klasifikaci je tedy určitá prvotní znalost dat, na základě které se algoritmus učí. Tou se rozumí

<sup>9</sup>Český Úřad Zeměměřičský a Katastrální



přidělení příznaků jednotlivým vzorkům dat, které determinují příslušnost daných vzorků do jednotlivých tříd. (Theodoridis, S., Koutroumbas, K., 2009)

Pokud vztáhneme tuto myšlenku na vlastní implementaci. Znalost dat před provedením analýzy může být reprezentována vizuálním vjemem uživatele mapových podkladů. Na základě těchto empirických zkušeností a znalostí, má sám uživatel určitou představu co na mapě vidí. Je schopen přibližně rozeznat základní povrchy jako například vodní plochu, les, pole a další. Ty v tomto případě reprezentují třídy pro klasifikaci. Tuto znalost musí uživatel nějakým způsobem předat algoritmu, aby získal data, na kterých by se mohl učit. To by bylo možné definováním oblastí nad danou mapou, u kterých by uživatel specifikoval třídu, do které spadají. Na základě těchto informací, už je možné provést klasifikaci vstupních dat do požadovaných tříd.

Problém tohoto řešení je, že uživatele mnohdy nezajímá, co se v dané oblasti nachází, protože to sám pozná, ale jde mu pouze o rozdělení do daných oblastí. V tomto případě je pro něj specifikace tříd zbytečně obtížným krokem v postupu. Dalším problémem je průnik množin dat pro učení a samotných klasifikovaných dat. Závislost těchto dvou množin ovlivňuje výsledky klasifikace a často vede k přeučení algoritmu pro daná data.

Také celkový koncept klasifikačních algoritmů spočívá na principu, že se algoritmus naučí na známé oblasti dat a následně se stále využívá daná konfigurace algoritmu pro klasifikaci dat, u nichž už rozdělení známo není. V tomto případě by se algoritmus musel učit stále dokola pro každou prováděnou klasifikaci.

Vylepšením celé aplikace by mohlo být využití persistence konfigurace algoritmu na straně serveru, kdy každým spuštěním by proběhlo jeho zpřesnění a dodatečné učení. Tímto by se aplikace každým spuštěním učila lépe klasifikovat. Zde ale vyvstává problém znalosti dat dopředu. Mapová data poskytovaná pomocí *WMS* umožňují přiblížení nebo oddálení na několika úrovních. To je samozřejmě výhodou pro uživatele, ale z pohledu využití jednoho klasifikátoru pro všechny úrovně přiblížení už ne. Klasifikátor musí mít přesně stanovené a daty definované třídy. Pokud si ale představíme, jak vypadají snímky z maximálního a minimálního přiblížení jako například na obrázku č. 13, vidíme, že využití stejných tříd pro tyto dva případy nepřipadá v úvahu.

Posledním požadavkem klasifikačních algoritmů je, aby definované třídy respektovaly pravděpodobnostní rozdělení klasifikovaných dat. To už je v případě možnosti spuštění analýzy na libovolných datech naprosto neudržitelné.

## Bez učitele

Další možností je tedy učení bez učitele, neboli shlukování. Rozdíl oproti klasifikaci je ten, že dané třídy blíže nespécifikujeme a pouze algoritmu předáváme informaci, na kolik tříd má data rozdělit. V případě shlukování je označujeme jako shluky. Ty se tvoří rozdělením dat do stejnorodých skupin na základě jejich příznaků.

Aplikované na tento případ to tedy znamená, že by uživatel musel pouze definovat počet hledaných druhů povrchu a zbytek by za něj udělal algoritmus. Výhoda shlukování je, že nezáleží na vstupních datech. Různé úrovně přiblížení tedy nebudou hrát roli.

Základní funkce těchto algoritmů spočívá ve vytvoření náhodných středů shluků, vůči kterým se následně vyhodnocuje podobnost analyzovaného vzorku. Tento vzorek se následně přidělí do shluku, jehož středu je nejpodobnější. Následně se přepočítá střed aktualizovaného shluku a takto se postupuje pro všechny vzorky (Tibshirani, R., Hastie, T., Friedman, J., 2009). Další možnou optimalizací je opakované spuštění algoritmu s nově náhodně rozmístěnými středy a srovnání, který běh měl nejlepší výsledek.

Vylepšení tohoto algoritmu by mohlo být označení oblastí uživatelem, na základě kterých by se odvodily středy shluků. To by uživateli dalo větší kontrolu nad algoritmem, urychlilo výpočet, protože by nebylo třeba ho opakovat pro eliminaci chyby náhodného rozdělení středů, a zároveň umožnilo definici počtu shluků nezbytnou pro algoritmus.

Další možností je předem analyzovat vstupní data a na základě jednoduchých charakteristik se pokusit predikovat středy shluků. Příkladem by mohlo být vytvoření histogramů pro jednotlivá pásma. Na základě těch se dá předpokládat, že v oblastech lokálních maxim budou vzorky, které budou definovat středy shluků. Tato metoda by umožnila i automatickou volbu počtu shluků, na které mají být data rozdělena.

Na základě těchto poznatků bude vhodnější využití shlukovací analýzy. Je flexibilnější co se vstupních dat týče, což je základním problémem využití klasifikace s učitelem.

## 5.5 Výběr technologií pro samotnou implementaci

### Strana klienta

S ohledem na vybranou architekturu implementované aplikace se bude jednat o webovou aplikaci. V této oblasti se nabízí více technologií, pomocí kterých je možné klientskou část aplikace vytvořit. Nejrozšířenější je však využití jazyka *JavaScript*.

*JavaScript* je interpretovaný jazyk, jehož interpret je součástí všech moderních internetových prohlížečů. To zajišťuje jeho podporu na většině platform. Slouží pro tvorbu interaktivních webových aplikací, které jsou schopny interagovat s uživatelem bez nutnosti obnovení stránky. (Haverbeke, M., 2014)

V tomto jazyce jsou napsány dvě knihovny pro práci s geografickými daty ve webových aplikacích. První je *Open Layers*, která je zřejmě nejrozšířenější. Jedná se o open source nástroj, pomocí kterého lze vytvořit mapovou prohlížečskou službu umožňující správu vrstev, objektů a geometrii v mapě. Stejně možnosti nabízí i

druhá knihovna *Leaflet*. To je o mladší projekt, který se zaměřuje spíše na jednodušší řešení.

Při testování obou těchto knihoven byla vybrána pro experimentální implementaci knihovna *Open Layers*. Tvorba a exportování geometrií, reprezentujících vstup uživatele, bylo o poznání jednodušší a vhodnější pro požadované řešení. Také bylo možné řešit export obsahu mapového okna, který reprezentoval vstupní data pro klasifikaci.

Bude také využito technologie *Ajax*<sup>10</sup>. Jedná se o využití asynchronního volání serveru z *JavaScript*. To umožňuje aktualizaci pouze části dat na stránce a ne celé stránky. (Woychowsky, E., 2007)

Klientská část aplikace tedy bude implementována v jazyce *JavaScript* za využití knihovny *OpenLayers* a spouštění analýz bude probíhat pomocí technologie *Ajax*. To zajistí funkčnost bez nutnosti obnovování stránky a bude uživatelsky přívětivější.

### Serverová strana

Serverová strana by s ohledem na navrženou architekturu aplikace měla obsahovat jak aplikační, tak datovou vrstvu. Vzhledem k tomu, že ale nebude využívat žádnou datovou persistenci, datová vrstva nebude využita. Hlavní důraz tedy bude kladen na aplikační vrstvu. Ta bude provádět celý proces analýzy.

Aplikační vrstva bude mít za úkol:

1. Přijmout vstupní data a uživatelskou konfiguraci.
2. Zpracovat data do přijatelného formátu pro jejich klasifikaci.
3. Extrahovat dodatečné informace pro konfiguraci algoritmu.
4. Provést klasifikaci.
5. Vyčistit data.
6. Převést výstup do požadovaného formátu.
7. Vrátit finální výstup klientu.

Pro implementaci bude využito jazyka *Python*. Jedná se o interpretovaný jazyk s poměrně jednoduchou a srozumitelnou syntaxí. Je široce využíván v oblasti geoinformačních systémů a většina aplikací, pokud vůbec nabízí tu možnost, mají aplikační rozhraní pro tento jazyk.

Bude třeba zprovoznit webový server, který bude zpracovávat požadavky klienta. K tomu poslouží *VPS*<sup>11</sup> s distribucí *Ubuntu* operačního systému *Linux*. Na tom bude spuštěn *Apache HTTP Server*, který bude předávat požadavky samotné

<sup>10</sup>Asynchronous JavaScript And XML

<sup>11</sup>Virtual Private Server

aplikaci spuštěné na interpretu jazyka *Python* pomocí *mod\_wsgi*. Pro kostru aplikace poslouží framework *Flask*, který poskytuje základní rozhraní pro zpracování *HTTP* požadavků a jejich odesílání. Tyto technologie obstarají první krok procesu na serverové části.

Po přijetí dat na serverové části přichází samotná analýza. Ta se skládá ze tří částí. První část je samotné načtení a zpracování dat do přijatelné formy pro klasifikační algoritmus. Teprve až tento krok proběhne úspěšně, může být provedena samotná analýza. Po té bude následovat čištění výstupních dat a jejich konverze do požadované formy.

V rešeršní části vyplynuly jako hlavní kandidáti *SAGA GIS* a *GRASS GIS*. Oba tyto systémy byly podrobeny testování využitelnosti a vhodnosti pro tuto implementaci. Oba tyto systémy uvádějí, že zahrnují aplikační programové rozhraní pro jazyk *Python*. Toto *API* je ale tvořeno knihovnou, která pouze obaluje volání jednotlivých modulů pomocí příkazové řádky, což není zcela ideální. Navíc oba systémy využívají jinou formu vstupních a výstupních dat.

V případě *GRASS GIS*, se jedná o velmi rozsáhlý a komplexní nástroj, jehož většina funkcionality by byla nadbytečná a navíc jeho práce s daty je poměrně nestandardní, což by činilo jeho využití obtížnějším.

*SAGA GIS* využívá pro spouštění jednotlivých modulů knihovnu vyvinutou pro stejnou funkcionalitu v rámci *GRASS GIS*.

Další možností bylo využití pouze knihovny *GDAL* pro zpracování a reprezentaci geografických dat a pro samotnou analýzu využít vlastní implementace algoritmů, nebo využití knihoven zaměřených na obecné úlohy strojového učení, do kterých klasifikační úlohy spadají.

Tato metoda se ukázala jako ideální s ohledem a specifický způsob načítání vstupních dat a možnosti zasahovat libovolně do samotného procesu. Pro práci s maticemi byla využita knihovna *NumPy* a pro postprocessing a filtrování knihovna *SciPy*.

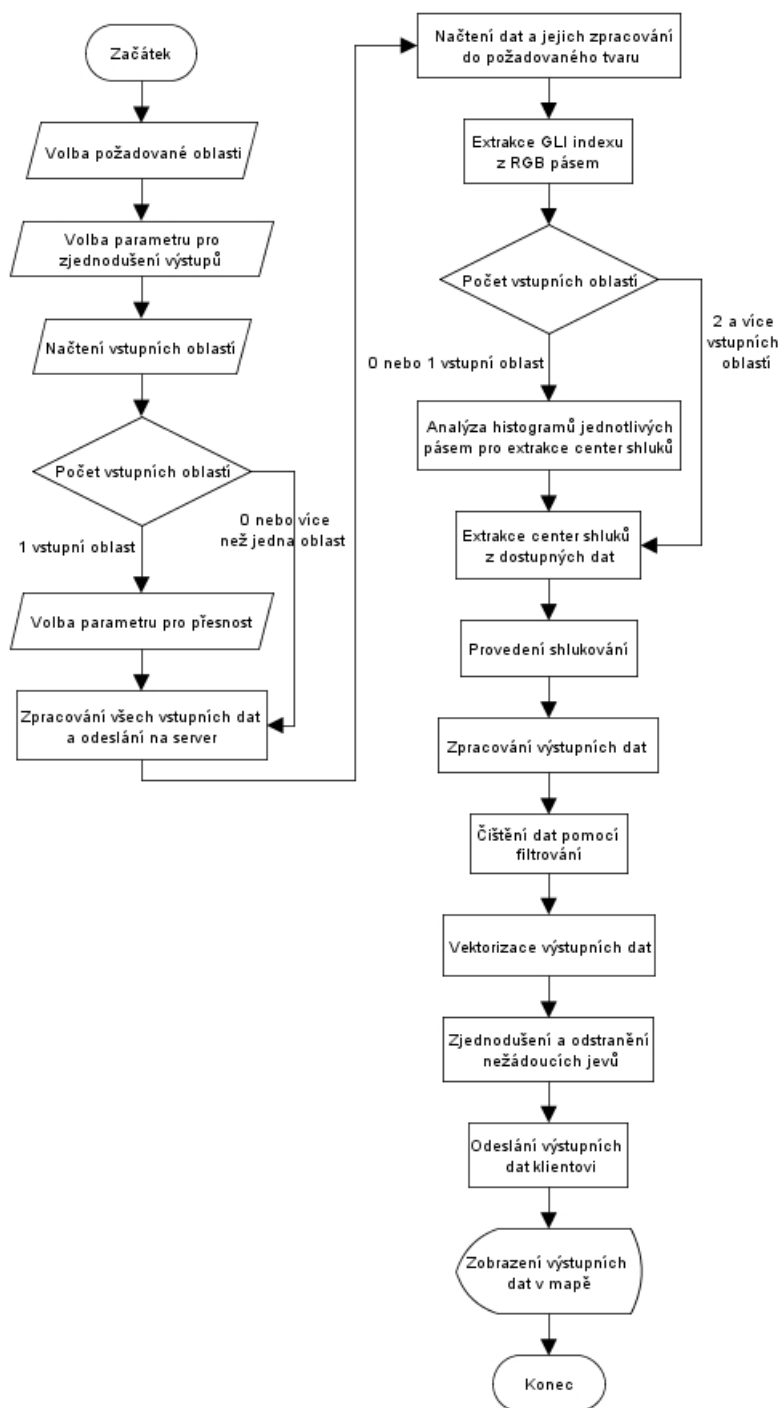
## 5.6 Implementace samotného řešení

Po průzkumu dostupných technologií, stanovení přesných požadavků na řešení, zhodnocení využitelnosti těchto řešení a volbě samotných technologií pro implementaci přichází samotná tvorba tohoto řešení. Nejdříve rozdělíme celý proces do jednotlivých kroků.

### Jednotlivé kroky procesu

1. Nejdříve se odesílá na server výřez mapového okna *WMS* služby spolu se vstupními daty od uživatele
2. Následně probíhá analýza histogramů jednotlivých pásem vstupního snímku pro detekci center shluků parametrizovaná uživatelem.
3. Provádí se extrakce center z masek definovaných uživatelem.
4. Spouští se samotná klasifikace snímku na počet shluků definovaný uživatelem nebo analýzou histogramů.
5. Výsledek prochází maticovými filtry pro vyhlazení výstupu.
6. Dochází k vektorizaci bitmapy s extrakcí příznaků na atributy polygonů.
7. Polygony prochází algoritmus pro odstranění zanedbatelných nebo ořezání nechtěných částí polygonů parametrizovaný uživatelem.
8. Výsledné polygony jsou vráceny na klienta uživateli.

Podrobný průběh celého procesu je názorně ukázán na vývojovém diagramu na obrázku č. 14.



Obrázek 14: Vývojový diagram celého procesu provedení analýzy.

## Uživatelské rozhraní

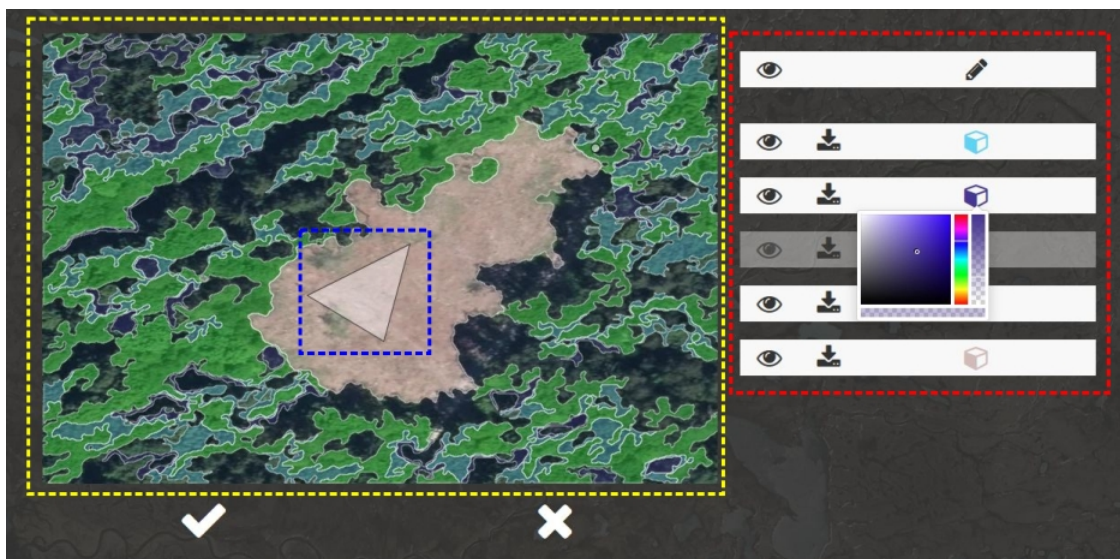
Uživatelské rozhraní aplikace je tedy implementováno pomocí jazyků *HTML*, *CSS* a *JavaScript*.

Vizuální stránka byla vytvořena za pomoci knihovny *Bootstrap*, která usnadňuje tvorbu *GUI* webových aplikací, responzivity a jednotného zobrazení ve většině moderních prohlížečů.

Na obrázku č. 15 můžeme vidět finální návrh uživatelského rozhraní. Jedná se pouze o výřez hlavní oblasti, která obsahuje veškerou funkcionalitu. Na ukázce je vidět dvě základní vyznačené oblasti žlutou a červenou barvou.

Žlutá oblast obsahuje mapové okno, které uživateli umožňuje prohlížení vstupních dat, definici vstupních oblastí a zároveň prohlízet i výstupy analýzy. Modře označený polygon je ukázkou vstupní definované oblasti. Ostatní barevné polygony, překrývající mapový podklad, jsou výstupními daty.

Červená oblast označuje panel vrstev. V tom je vidět jako první položka vrstva, do které uživatel zakresluje vstupní oblasti. Ta umožňuje pouze vytnutí jejího zobrazení, jinak s ní nelze manipulovat. Další vrstvy už reprezentují jednotlivé shluky, do kterých byla vstupní data roztržena. Ty jsou převedeny do polygonů a zobrazeny nad vstupními daty. Pro každou tuto vrstvu je možné změnit její barvu a průhlednost. Je také možné vytnout její zobrazení úplně. Hlavní je možnost jejího vyexportování. To je možné ve formě formátů *GeoJSON* a *KML*. Jedná se o standardizované, široce rozšířené formáty, které lze využít v libovolných GIS.



Obrázek 15: Ukázka vytvořeného uživatelského rozhraní. (Žlutě označená oblast obsahuje mapové okno. Modrá oblast znázorňuje ukázkou uživatelem definované oblasti. červená část obsahuje ovládací panel, ve kterém se zobrazují jednotlivé vrstvy výstupu.)

Dále už rozhraní obsahuje pouze dvě tlačítka pod mapovým oknem. Levé tlačítko spouští analýzu, což vyvolá zablokování mapového okna a zobrazí se ukazatel

průběhu. Po skončení procesu se mapové okno opět aktivuje a vykreslí se do něj výstup a odpovídající vrstvy se zobrazí v panelu pro správu vrstev. Pravé tlačítko vymaže všechny vrstvy kromě té pro vstupní oblasti. U té vymaže pouze všechny definované oblasti. Resetuje tedy aplikaci do původního stavu.

Rozložení a struktura je vytvořena pomocí *gridu*<sup>12</sup> knihovny *Bootstrap*. Ta rozděljuje stránku na jednotlivé řádky a každý řádek se skládá ze dvanácti stejně širokých dílů. V řádku definujeme sloupce, mezi které rozdělíme všech 12 dílů.

Hlavní část aplikace je tedy rozdělena na dva sloupce. Jeden obsahuje mapu a spodní ovládací tlačítka a druhý, který obsahuje seznam vrstev. Toto rozdělení zároveň ovlivňuje chování aplikaci na menších zobrazovacích zařízeních. Tam se přizpůsobí jednotlivé sloupce šířce zobrazení a přeskládají se pod sebe.

### Databáze prostorových dat a jejich zobrazení

Prostorová data využitá pro analýzu jsou tedy poskytována formou *WMS*. Jedním z úkolů práce je vytvoření databáze prostorových dat využitelných pro jejich analýzu. Tu tvoří seznam volně dostupných mapových serverů a služeb, které poskytují data v požadovaném formátu. Z těch má uživatel na výběr.

K jejich zobrazení je využita knihovna *Open Layers*. Ta obsahuje již implementované mapové okno, které je možné libovolně upravovat. Pro vykreslování dat využívá prvek *Canvas* z *HTML5*. Do toho je možné přidávat jednotlivé vrstvy s různými datovými zdroji. Mapové okno automaticky umožňuje přibližování, oddalování a posun ve zobrazených datech.

Mezi tyto zdroje mimo jiné patří i *WMS*. To umožňuje vytvoření vrstvy, které stačí specifikovat URL dané *WMS* služby a knihovna už obstará veškerou komunikaci a zobrazí poskytovaná data. Samotná databáze geografických dat se bude sestávat ze sady *URL* na jednotlivé *WMS*. Tím bude zajištěna datová základna, která uživateli velmi jednoduše umožní výběr oblasti a dat pro analýzu.

### Uživatelský vstup a jeho zpracování

Mapové okno s vrstvou *WMS* umožní uživateli se navigovat k požadované oblasti s daným přiblížením. Výřez mapového okna je zároveň definicí oblasti, která bude analyzována. Při spuštění je vznesen dotaz na *Canvas*, který obstarává vykreslování, o poskytnutí binární reprezentace obrazových dat pro danou oblast. Knihovna musí opět odeslat tento požadavek na *WMS*, a její odpověď je ve formě kódování *base64*<sup>13</sup>. V tomto tvaru už budou data odeslána na server.

Dalším důležitým vstupem jsou uživatelem definované oblasti. Je tedy třeba umožnit uživateli, aby jednoduše definoval plochy nad mapovým oknem. K tomu je využita možnost knihovny *OpenLayers* pro tvorbu geometrií. Nad podkladovou vrstvou *WMS* je definována další vrstva s prázdným zdrojem polygonových vektorových

---

<sup>12</sup>mřížka

<sup>13</sup>Kódování pro převod binárních dat do tisknutelných znaků.



dat. Ta nabízí možnost tvorby jednotlivých prvků jejich definicí v mapě. Uživateli tedy stačí kliknutím do mapy začít definici oblasti a dalšími kliknutími specifikovat zbylé body polygonu. Dvojklikem ukončí tvorbu dané oblasti a může začít s definicí další. Při spuštění analýzy je tato vrstva vyexportována do formátu *GeoJSON*, ve kterém je odeslána na server.

Poslední potřebné údaje pro spuštění analýzy jsou metadata o binární reprezentaci odesílaného rastrového snímku. V této formě se jedná pouze o obyčejná obrazová data bez jakékoli geografické informace. Aby bylo možné je na serveru zpracovat, bude potřeba získat *bounding box*<sup>14</sup>, informace o projekci a geografické rozlišení. Všechny tyto informace umožňuje opět získat knihovna *OpenLayers*.

### Odeslání dat na server

K odeslání požadavku na server bude využito technologie *AJAX* pro asynchronní komunikaci. Ta umožní bezešvý běh aplikace, aniž by ji bylo třeba obnovovat. Požadavek bude odeslán metodou *POST* a bude čekat na odpověď serveru.

### Přijetí dat na serveru

Pro testování byl zřízen *VPS*<sup>15</sup>, na kterém je aplikace spuštěna. Běží na operačním systému *Linux* a pro komunikaci pomocí protokolu *HTTP* slouží webový server *Apache*. Ten zpracovává veškeré požadavky a směřuje je na aplikaci.

Vzhledem k tomu, že je aplikace napsána v jazyce *Python*, bylo nutné přidat rozšíření *mod\_wsgi*. To umožňuje spuštění interpretu jazyka *Python* webovým serverem. Ten tedy při příchozím požadavku inicializuje aplikaci a předává jí *HTTP* požadavek se všemi daty.

Aplikace samotná využívá frameworku *Flask*. Jedná se o jednoduchou kostru pro webové aplikace obsahující pouze implementaci zpracování a odesílání *HTTP* requestů a routing. Routing je volání odpovídajících metod na základě cesty uvedené v *URL* požadavku. Díky tomuto tedy je předán požadavek metodě, kde s ním lze dále pracovat. V požadavku jsou tedy data zobrazená v tabulce č. 2

### Zpracování dat

Prvním krokem je zpracování dat bitmapy. Tu je třeba načíst z požadavku ve formě řetězce v *base64*. Ten je dekodován a uložen do paměti jako *StringIO* objekt.

Pro zjednodušení testování byla vytvořena třída *OrthophotoClassifier* s obecnými metodami pro načtení a zpracování dat. Od této třídy dědí jednotlivé třídy pro různé druhy klasifikačních algoritmů. Ty se liší implementací metody *classify*.

<sup>14</sup>Jedná se o využívanou reprezentaci geografické oblasti pomocí souřadnic levého spodního a pravého horního rohu obdélníku.

<sup>15</sup>Virtual Private Server

Tabulka 2: Data HTTP požadavku.

<code>cols</code>	Počet pixelů vstupních dat na šířku.
<code>rows</code>	Počet pixelů vstupních dat na výšku.
<code>x_max</code>	Maximální hodnota zeměpisné šířky pro daný výřez dat.
<code>x_min</code>	Minimální hodnota zeměpisné šířky pro daný výřez dat.
<code>y_max</code>	Maximální hodnota zeměpisné délky pro daný výřez dat.
<code>y_min</code>	Minimální hodnota zeměpisné délky pro daný výřez dat.
<code>epsg</code>	SRID <sup>16</sup> jednoznačně definující projekci pro daná data.
<code>polygon</code>	Pole definic vstupních oblastí uživatele ve formátu WKT <sup>17</sup> .
<code>data</code>	Binární data bitmapy v kódování base64

Nejdříve je tedy třeba vytvořit instanci třídy daného klasifikátoru. Ta má základní konfiguraci pro počet vstupů algoritmu a počet výstupů. Počet vstupů popisuje počet hodnot každého vzorku, což v tomto případě znamená 3, protože máme hodnoty pro červenou, zelenou a modrou. Počet výstupů je závislý na územích definovaných uživatelem. Pokud není definováno žádné území, třída automaticky detekuje optimální počet výstupních tříd. Pokud je definováno pouze jedno území, bude počet výstupních tříd 3, kdy požadovaná třída bude jen jedna z nich. Při větším počtu definovaných území už bude počet tříd odpovídat jejich počtu.

V tomto bodě je možné načíst vstupní data do klasifikátoru. K tomu slouží metoda `load_orthophoto`. Ta má dva argumenty. Prvním je objekt `StringIO`, ve kterém jsou načtená binární data. Druhý argument je asociativní pole `spat_data`, které obsahuje data potřebná pro georeferenci dané bitmapy. Těm odpovídají parametry `cols`, `rows`, `x_max`, `x_min`, `y_max`, `y_min` a `epsg` z přijatého požadavku.

Metoda načítá binární data pomocí knihovny *SciPy*, odstraňuje hodnoty alfa obsahující informaci o průhlednosti a výsledek ukládá do trojrozměrné matice o velikosti `rows`, `cols`, 3.

### Odvození GLI indexu z pásem RGB

Dalším krokem, pro zlepšení výsledků klasifikace, je přidání dodatečných příznaků. Jak bylo zmíněno v rešeršní části, klasická data využívaná pro klasifikaci jsou většinou pořizována ve více pásmech než jen RGB. Ty se často využívají pro lepší detekci zeleně nebo vodních ploch.

V tomto případě ale není žádné dodatečné pásmo, které by mohlo být využito. Přesto například klasifikace vodních ploch byla velmi nepřesná. Alternativou tedy bylo využití indexů dopočítaných z dostupných RGB pásmech, které by zvýraznily

některé povrchy. Byli testovány dva tyto indexy jedním byl  $GLI$ <sup>18</sup> a druhým byl  $VARI$ <sup>19</sup>.

$$GLI = \frac{2R_g - R_r - R_b}{2R_g + R_r + R_b} \quad (1)$$

$$VARI = \frac{R_g - R_r}{R_g + R_r - R_b} \quad (2)$$

(Hunt, E. R. Jr., Doraiswamy, P. C., 2013)

Index  $VARI$  se se neukázal jako využitelný, ale  $GLI$  zvýrazňoval vodní plochy, které nebyly na klasických RGB dobře detekovatelné, jak je vidět na obrázku č. 16.



Obrázek 16: Ukázka rozdílu zeleného pásma (vlevo) a  $GLI$  (vpravo) na vodní ploše.

Třída `OrthophotoClassifier` má tedy metodu `compute_gli_index`, která spočítá pro každý pixel hodnotu  $GLI$  a tím vznikne čtvrté pásmo a tedy čtvrtý vstupní parametr pro klasifikátor.

### **Načtení a zpracování uživatelem definovaných oblastí**

Načtení uživatelem specifikovaných oblastí definujících požadované třídy pro klasifikaci je provedeno metodou `load_mask_from_polygon` třídy `OrthophotoClassifier`. Ta má vstupní argumenty `index` dané třídy, definici polygonu ve formátu WKT a opět asociativní pole `spat_data` pro georeferenci.

Pro načtení je využita knihovna *GDAL*. Její pomocí je vytvořena vektorová vrstva, která umožňuje načtení geometrií z formátu WKT. Díky tomu je obstarána reprezentace reprezentací území a specifikací projekce se jedná o validní geografická data.

---

<sup>18</sup>Green Leaf Index

<sup>19</sup>Visible Atmospherically Resistant Index

Dalším krokem je kontrola, jestli se dané oblasti nachází v požadovaném výřezu. Ta je provedena pomocí vytvoření další vektorové vrstvy obsahující polygon definovaný hranicemi výřezu. Pro každou oblast je pak proveden její průsečík s tímto polygonem výřezu. Tím získáme buďto tu stejnou oblast, pokud spadá celá do daného výřezu, nebo tuto oblast oříznutou o přesahující část. Poslední možností je, že oblast leží celá mimo daný výřez, kdy se už dále neuvažuje.

Po provedení této kontroly máme zaručeno, že polygony vstupních oblastí leží v daném výřezu. Nyní je potřeba provést jejich převod na bitmapu. To je provedeno pomocí knihovny *PIL*. Ta umožňuje rasterizaci polygonu, pokud definujeme uzly polygonu pixely v daném gridu.

Takto rasterizovaná forma oblasti specifikuje masku nad podkladovými daty pro danou třídu. Máme tedy informaci, které pixely mají definovat danou třídu a na základě těchto informací bude dál determinován střed shluku pro danou třídu.

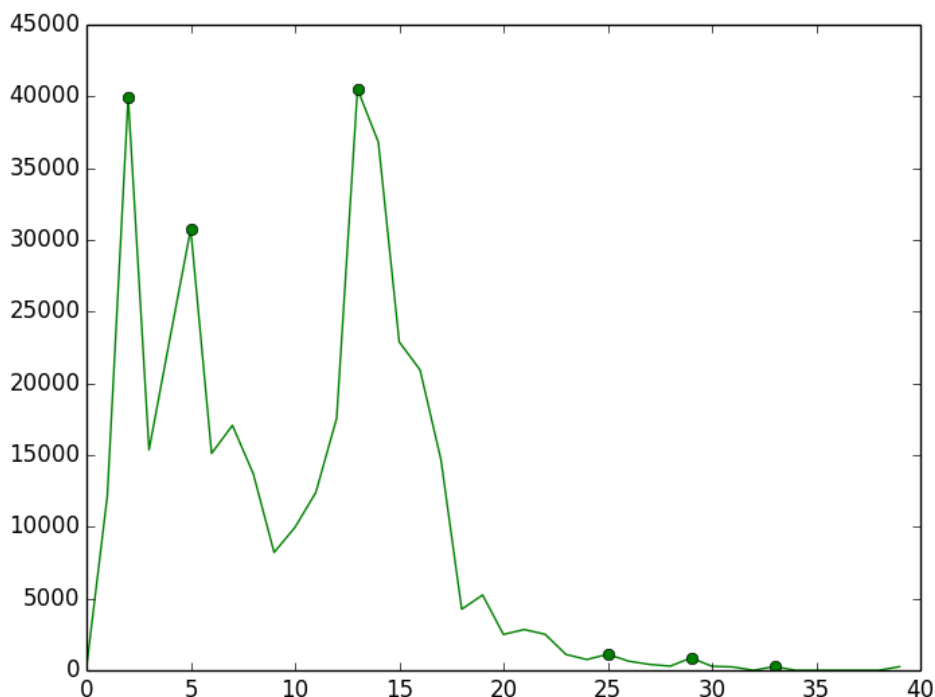
### Extrakce center shluků pomocí histogramů

Pokud je uživatelsky definovaných oblastí méně než dvě, je třeba determinovat zbývající, nebo určit jejich počet. Tento proces je sám o sobě jednou z úloh strojového učení. Z důvodu časové náročnosti na výpočet tak bylo využito analýzy histogramů. Metodu jsem sám navrhl a implementoval, protože se jevila jako nejjednodušší a poměrně efektivní cesta k detekci center shluků.

Je založena na základním principu vlastnosti shluků. Ty obsahují vzorky, jejichž příznaky mají podobné hodnoty a zároveň odlišné od ostatních shluků. Pokud vezmeme v potaz samotný význam histogramu pro jedno z pásem, jedná se o rozložení četností vzorků pro jejich hodnoty v intervalu  $\langle 0, 255 \rangle$ . Shluk ale nebude obsahovat pouze vzorky se stejnými hodnotami, ale i podobnými. Podobnost v tomto případě znamená, že euklidovská vzdálenost daného vzorku od centra shluku, kterému je podobný, je menší, než od center ostatních shluků. Proto tento interval hodnot rozdělíme na sérii menších intervalů. Pro každý tento interval sečteme všechny vzorky do něj spadající a získáme méně početnějších skupin, na základě kterých už je možné hodnotit jejich významnost v datech.

Pokud se podíváme na obrázek č. 17, vidíme ukázkou histogramu pro zelené pásmo. Na ose  $x$  je číslo intervalu. Zde můžeme pozorovat že interval  $\langle 0, 255 \rangle$  byl rozdělen na 40 menších stejně velkých intervalů. Na ose  $y$  je vyneseno počet vzorků spadajících do jednotlivých intervalů. V levé části jsou dobře vidět tři výrazná lokální maxima, která budou bezesporu dobře reprezentovat jednotlivé shluky. V pravé části jsou další tři lokální maxima, která nejsou tak výrazná. U těch by mohlo být sporné jejich vyhodnocení jako shluků.

Velmi důležitou roli zde hrají požadavky na přesnost výstupu analýzy. Pokud si uživatel přeje data rozdělit co nejpřesněji, lze využít všech lokálních maxim. V tom případě bude výstup ale velmi členitý a v některých případech může rozdělit data, která by uživatel očekával v jedné třídě, do dvou nebo více tříd. Možností, jak se tomuto jevu vyhnout, je zkoumat okolí lokálního maxima. Konkrétní příklad lze



Obrázek 17: Ukázka histogramu zeleného pásma rozděleného do 40 stejně velkých intervalů a detekce lokálních maxim.

pozorovat opět na obrázku č. 17. Pokud se podíváme na lokální maximum v intervalu číslo 17, vidíme, že je poměrně zanedbatelné v porovnání s tím v intervalu 14. Toto rozhodnutí už ale není možné vykonat automatizovaně za uživatele. Ten bude mít možnost nastavení parametru pro zpřísnění detekce těchto lokálních maxim.

Pro každé pásmo (včetně *GLI* pokud bylo dopočítáno) je vytvořen histogram. Ten reprezentuje rozložení četností pro dané hodnoty v pásmu. Metoda vychází z předpokladu, že výrazná lokální maxima představují jednotlivé shluky. Evidentně jsou hodnotami vzájemně výrazně odlišné a tedy představují stejnorodé shluky.

Pro každý histogram pásma tedy najdeme lokální maxima a zvolíme to, které jich má největší počet. V tomto pásmu je segmentace nejvýraznější. Počet lokálních maxim tedy determinuje finální počet tříd.

### Shluková analýza

Následně je spuštěna samotná shluková analýza metodou `classify`, kterou implementují jednotlivé instance dědicí ze třídy. Té je předána transformovaná matice s jednotlivými pásmy do dvourozměrné matice ve které skládající se ze sady vzorků. Každý vzorek odpovídá jednomu pixelu bitmapy a má daný počet příznaků. Dále je předán počet počet shluků, do kterých mají být data rozdělena a jejich středy.

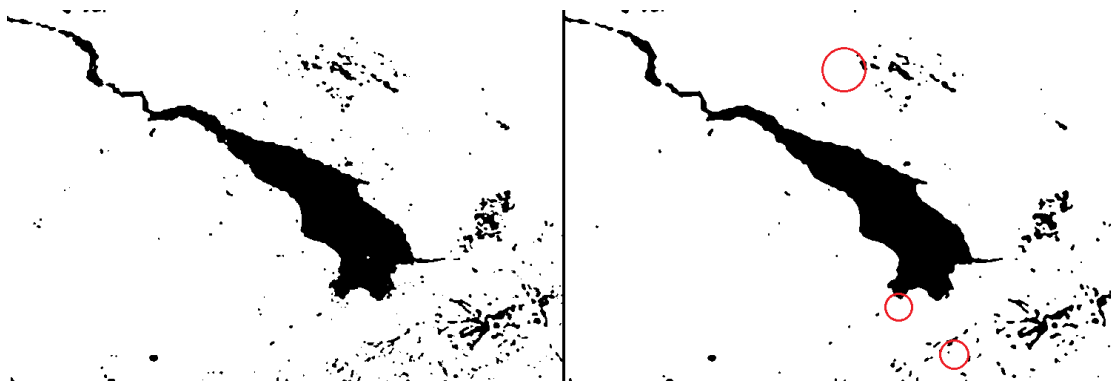
Na základě těchto informací algoritmus projde veškeré vzorky a postupně je rozdělí do jednotlivých shluků.

Výstupem tohoto algoritmu je jednorozměrné pole, které má stejný počet prvků, jako byl počet vzorků vstupní matice. Hodnoty těchto prvků jsou číselné a reprezentují jednotlivé shluky. Tyto data jsou následně opět transformována do tvaru matice o původních rozměrech, čímž získáváme opět bitmapu s hodnotami v intervalu  $\langle 0, n - 1 \rangle$ , kde  $n$  je počet shluků.

### Čištění výsledných dat filtry pro obrazová data

Jelikož jsou data klasifikována po jednotlivých pixelech, je ve výsledných datech mnoho osamocených pixelů uprostřed oblastí s odlišnou třídou. Takto malé oblasti nejsou žádoucí ve výsledných datech, kde by se mělo jednat o souvislé plochy přijatelné velikosti s ohledem na plochu analyzované oblasti.

Z toho důvodu je využito rastrových filtrů pro odstranění tohoto „šumu“. Bylo testováno několik druhů filtrů jako například *gaussian*, *percentile*, *median*, *rank*, *binary opening*, ale jako jediný vhodný se ukázal mediánový filtr. Ten spolu s definicí oblasti pro filtrování a jejími váhami odstraní většinu nežádoucích prvků. Vliv mediánového filtru je zobrazen na obrázku č. 18.



Obrázek 18: Ukázka vlivu mediánového filtru na odstranění šumu. Červeně označené oblasti ukazují nejvýraznější změny.

### Vektorizace a čištění dat

Rastrová data jsou nyní rozdělena do jednotlivých tříd, ze kterých byl odstraněn šum a tvoří více souvislé oblasti. Dalším krokem je vektorizace a souběžné čištění dat od nechtěných jevů.

Nechtěné jevy představují malé až zanedbatelné oblasti, které nemají další využití. U těch nastává otázka, jak hodnotit jejich rozlohu s ohledem na možnosti různého prostorového rozlišení snímků. Univerzálním řešením je využití poměru jejich reálné plochy vůči ploše analyzovaného výřezu. Díky tomu lze hranici nastavit pomocí procentuálního poměru. Jedná se však opět o čištění dat, u kterého by měl

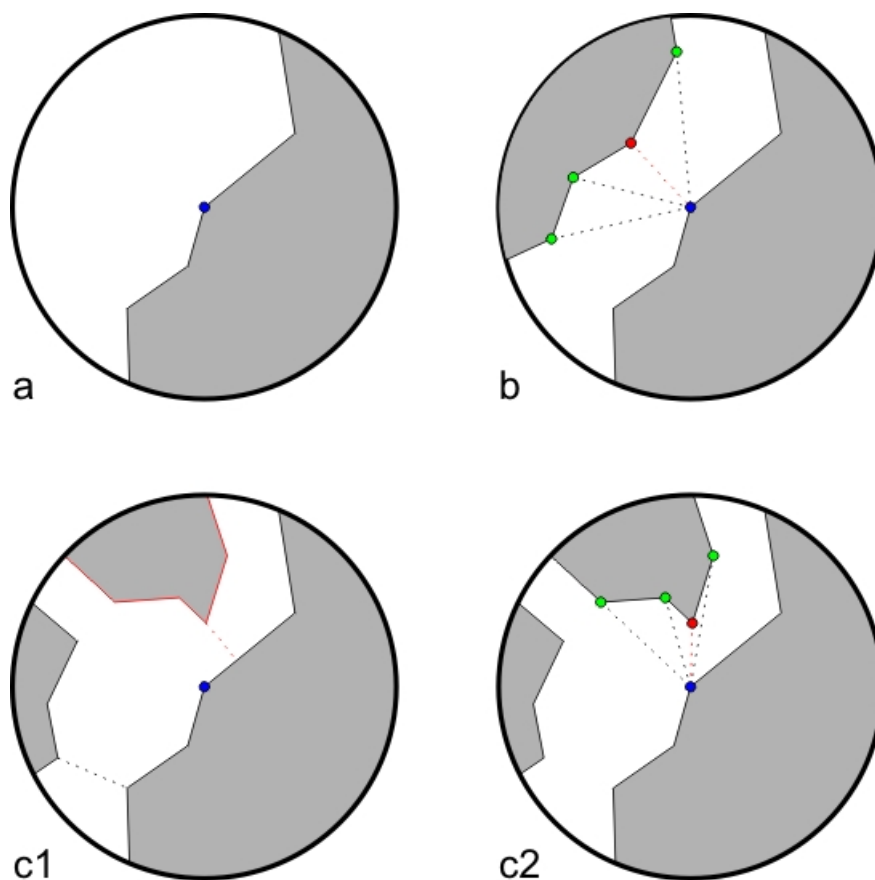
mít poslední slovo uživatel. Z toho důvodu je možné nastavit tento poměr před spuštěním analýzy.

Dalším nechtěným jevem jsou takzvané „mosty“. Název a definici jsem si vymyslel, protože odbornou specifikaci jsem v žádné literatuře nenašel. Jedná se o části polygonů, které jsou velmi úzké a propojují dvě rozsáhlejší oblasti. Pokud si představíme například sklizené pole, na které vede polní cesta, je velká pravděpodobnost, že budou pole i cesta klasifikovány jako stejná třída. Uživatel bude ale chtít pouze polygon daného pole a polní cesta nebude jeho zájem. V tom případě je třeba zkontrolovat každý polygon a v daných zúžených místech ho rozdělit. Opět ale vyvstává otázka minimální šířky těchto zúžení. I tento parametr bude tedy moci nastavit uživatel, aby postprocessing výsledných dat mohl ovlivnit podle své představy.

Jelikož nebyla nikde nalezena ani zmínka o tomto jevu, natož jeho odstranění, musel jsem navrhnout a realizovat postup sám a jeho ukázka je v příloze algoritmus odstranění mostů. Jeho princip spočívá v procházení všech definičních bodů polygonu, kde se analyzuje buffer o daném poloměru. Tento poloměr bude moci ovlivnit uživatel. Buffer je obalovou zónou dané geometrie, což v případě bodu představuje kruh. Pokud zjistíme průsečík tohoto bufferu s daným polygonem, získáme kolekci výsledných polygonů. Výsledkem tohoto může být několik stavů, které musí být vyhodnoceny. Ty jsou názorně ukázány na obrázku č. 19. V části 19(a) vidíme, že výsledkem průniku je pouze jeden polygon, což znamená, že v oblasti daného bodu není zúžení větší než poloměr bufferu. V části 19(b) už jsou výsledkem průniku dva polygony, což znamená, že v tomto místě se nachází zúžení a je třeba polygon rozdělit. Tento proces probíhá tak, že se zjistí vzdálenost všech bodů polygonu, který neobsahuje analyzovaný bod, vůči danému bodu. Je zvolen ten s nejmenší vzdáleností a tento pár je uložen a algoritmus pokračuje analýzou dalšího bodu. V případě 19(c) je znázorněna situace, kdy průnikem je více než dva polygony. V tomto případě je přidán pouze krok, který určí polygon, který je nejbližší polygonu obsahujícímu analyzovaný bod. To je znázorněno na 19(c1). Dále algoritmus pokračuje stejně jako v případě (b) jak je zobrazeno na části 19(c2).

Po tomto průchodu má algoritmus pole dvojic bodů, ve kterých má být polygon rozdělen. Body polygonu jsou tedy iterovány ještě jednou a v daných řezech polygon rozdělují. Při rozdělení vznikají nové zbytkové polygony definované oddělenou částí. Takto nově vzniklé polygony se přidávají do fronty polygonů čekajících na zpracování. To znamená, že budou tímto způsobem analyzovány i tyto nové polygony.

Na obrázku č. 20 je vidět vlevo výsledek klasifikace bez odstranění těchto mostů. Můžeme pozorovat, že silnice je klasifikována do stejné třídy, jako světlé pole a vytváří dohromady souvislý polygon. Na obrázku vpravo je vidět účinek algoritmu, který v zúžení polygon rozřízl. Je také vidět, že vedlejším efektem algoritmu je odstranění všech úzkých polygonů. Ty jsou algoritmem rozřezány na menší segmenty, které jsou následně smazány při kontrole velikosti polygonů.



Obrázek 19: Ukázka možných průniků polygonu s bufferem.

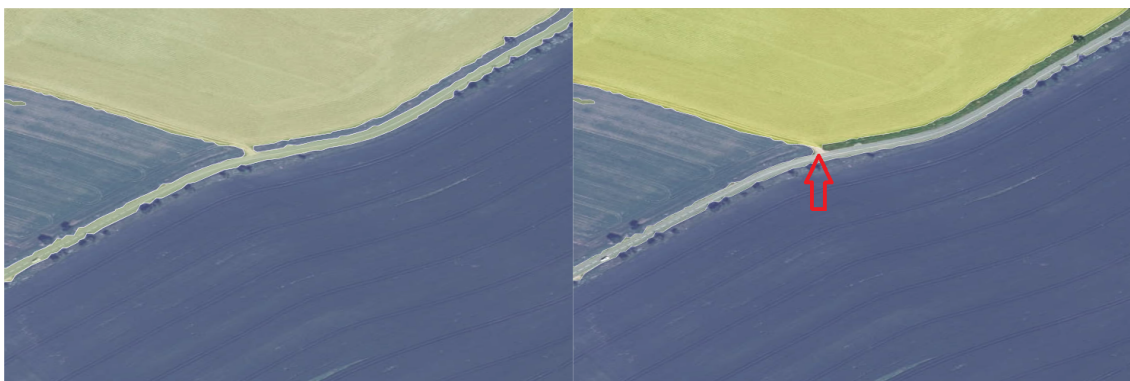
### Odeslání dat zpět klientovi

V této fázi už jsou vektorová data validní a očištěná od šumů i nežádoucích jevů. Tato data jsou převedena do pole záznamů ve formátu *GeoJSON* a odeslána klientovi jako odpověď na jeho *HTTP* požadavek. Framework *Flask* opět vytváří *HTTP* response, kterou odesílá přes webový server ke klientovi, kde se úspěšně ukončuje *AJAX* komunikace.

### Přijetí dat a jejich zpracování

Odpověď na *HTTP* požadavek, odeslaný pomocí *AJAX*, se vrací s daty ve formátu pole záznamů *GeoJSON*. Ten umožňuje knihovna *Open Layers* načíst jako datový zdroj pro vektorovou vrstvu. Každý prvek tohoto pole reprezentuje jednu ze tříd, na které měla být data rozdělena. Cyklus tedy vytváří pro každý záznam novou vrstvu. Ta je vykreslena do mapy a je jí náhodně vygenerována barva, aby byly lépe odlišitelné. Zároveň se pro ni vytváří položka v seznamu vrstev. V této části je možné zapínat a vypínat její viditelnost a nastavit jinou barvu a průhlednost. Hlavní funkcionalitou je ale možnost exportu každé z těchto vrstev.





Obrázek 20: Ukázka vlivu algoritmu pro odstranění mostů. Vlevo bez a vpravo s jeho využitím(červená šipka označuje místo, kde algoritmus polygon rozdělil).

## 6 Diskuze

Po dokončení implementace bylo provedeno testování. Byly využity data z mapového serveru ČUZK. V první fázi je testována přesnost samotné klasifikace bez využití algoritmů pro postprocessing. V další fázi funkčnost těchto algoritmů pro čištění a úpravu dat.

### 6.1 Zhodnocení výstupů klasifikace

Pro hodnocení výsledků klasifikace je potřeba využít referenčních dat. Ty slouží pro porovnání s třídou klasifikovanou pro jednotlivé vzorky. K tomu poslouží data z projektu *Corine Land Cover*. Tento projekt poskytuje zdarma data o územním pokrytí pro celou oblast Evropy. Ta jsou získávána také právě klasifikací satelitních snímků. Konkrétně využitá data *CLC2006* jsou ze satelitů *SPOT-4* a *IRS LISS III* (Caetano, M., Feranec, J., 2007).

Data *CLC* jsou rozdělena do šesti základních tříd:

1. Artificial surfaces
2. Agricultural areas
3. Forest and semi natural areas
4. Wetlands
5. Water bodies
6. No data

Ty se dále dělí na podrobnější třídy se specifitějším určením. Dalším důležitým aspektem je přesnost dat. Geometrická přesnost je minimálně 100 metrů a tématická přesnost je více než 85 %. Ta tedy slouží jako referenční množina.

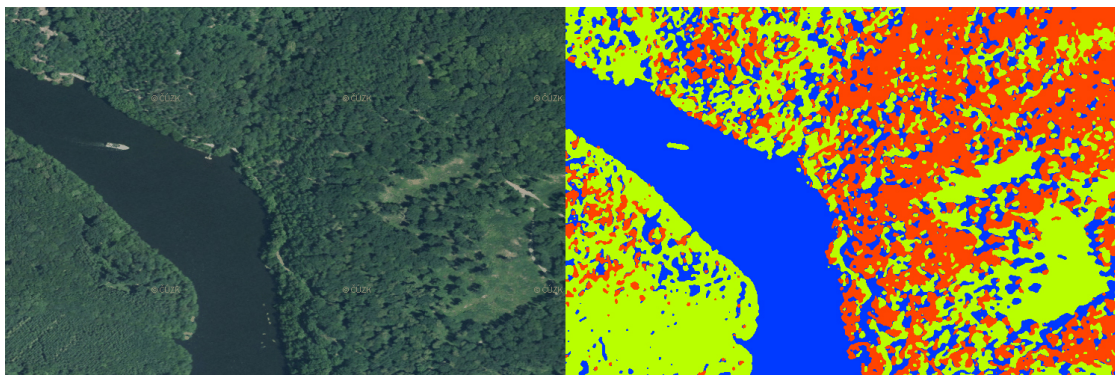
První testovaná oblast je z velké části pokrytá lesní a vodní plochou. Dále obsahuje mýtiny a loď na vodní hladině. Na obrázku č. 21 je vidět na levé straně zdrojová data a na pravé straně výsledek klasifikace v pseudobarvách. Každá barva reprezentuje jednu z výsledných tříd.

Data byla klasifikována do tří tříd na základě dat *CLC*. Tabulka č. 3 zobrazuje matici záměn<sup>20</sup>. Ta se využívá pro zobrazení správnosti zařazení vzorků do požadovaných tříd. V posledním sloupci je procento správně klasifikovaných vzorků v porovnání s daty *Corine Land Cover*.

Z výsledků je dobře viditelné, že vodní plocha je klasifikována s poměrně vysokou přesností 91 %. Zároveň je ale také dobře poznat zbylých 9 špatně klasifikovaných procent, které jsou většinou stíny stromu na snímku. Další dvě třídy reprezentují

---

<sup>20</sup>Confusion matrix



Obrázek 21: Ukázka výsledků klasifikace zalesněné oblasti s vodní plochou.

v databázi *CLC* zalesněné oblasti a to listnatý les a smíšený les. K rozeznání těchto detailů, zvláště co se týče vegetace, se využívá hlavně infračervené pásmo, které vstupní data nemají. To je hlavní důvod špatné klasifikace.

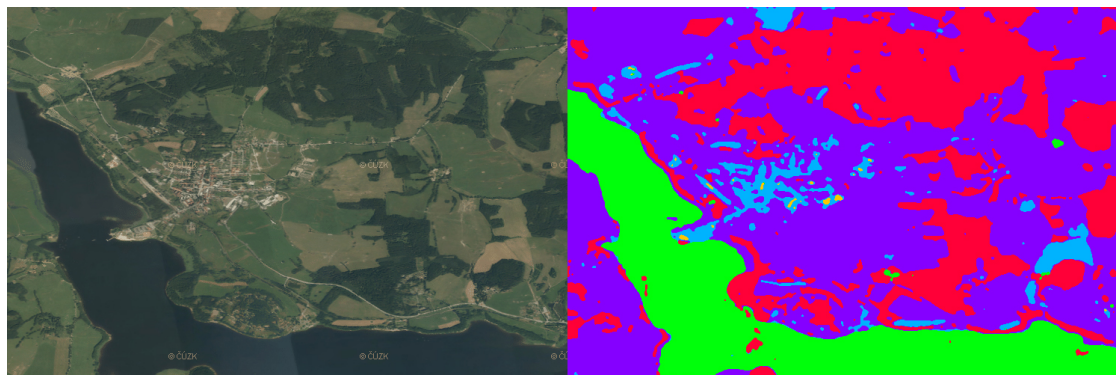
Poslední zajímavostí je klasifikace lodě na vodní hladině. Tak, jak už bylo dříve zmíněno, není s největší pravděpodobností požadovaný výstup. Z tohoto důvodu je zařazeno do postprocessingu odstranění malých polygonů. To by tento problém vyřešilo.

Tabulka 3: Matice záměn pro oblast obrázku č. 21.

	Voda	Smíšený	Listnatý	Přesnost
Voda	122699	5953	5878	91 %
Smíšený	30201	67777	7694	64 %
Listnatý	81310	8899	44589	33 %

Další oblastí je okolí vodní nádrže Lipno. Jedná se o výřez s větším prostorovým rozlišením pokrývající převážně vodní plochu, zemědělské plochy, les a zastavěné oblasti. S ohledem na větší prostorové rozlišení vstupních dat by měla být přesnost v porovnání s daty *CLC* větší. Na obrázku č. 22 je opět vidět výsledky klasifikace a v tabulce č. 4 matici záměn pro jednotlivé třídy. Tentokrát probíhala klasifikace na 5 tříd a to zalesněné plochy, zemědělské oblasti, zastavěné oblasti, vodní plochy a výrazné střechy s vysokou odrazivostí.

Opět můžeme pozorovat vysokou přesnost v detekci vodních ploch. V tomto případě je to 98 %, což je velmi vysoká přesnost. Co se týče lesů a polí, je vidět, že členitost výsledných oblastí je, i při tomto prostorovém rozlišení, stále mnohem větší než u dat *CLC*. Z toho důvodu jsou přesnosti klasifikace těchto dvou tříd nižší. Nejnížší přesnost má třída reprezentující zastavěnou oblast. Ta je v datech *CLC*



Obrázek 22: Ukázka výsledků klasifikace větší oblasti se zemědělskou plochou, vodní nádrží a obydlými oblastmi.

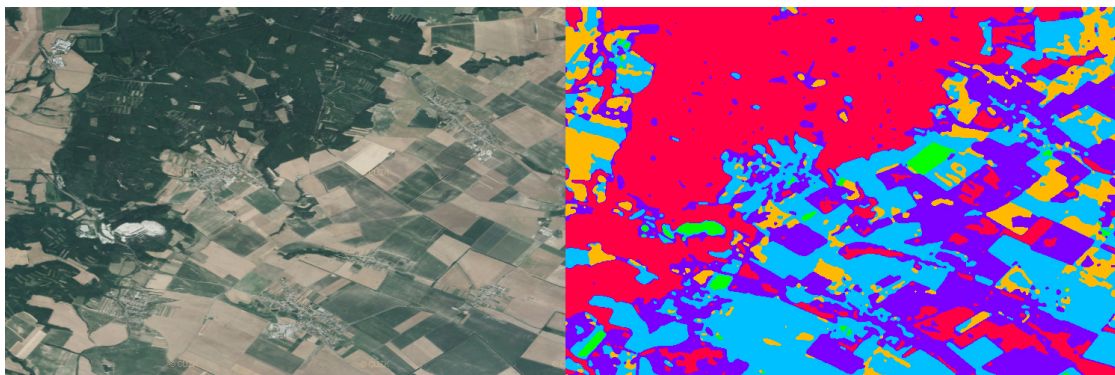
poměrně hodně generalizovaná což může být jednou z příčin. Další může být její nejasnost i na samotných vstupních datech.

Tabulka 4: Matice záměn pro oblast obrázku č. 22.

	Les	Pole	Město	Voda	Střechy	Přesnost
Les	73520	24350	143	1554	23	73 %
Pole	6182	166348	4093	5422	3207	89 %
Město	257	8367	4502	809	2524	27 %
Voda	466	437	0	72441	11	98 %
Střechy	0	53	52	14	225	65 %

Poslední oblastí pro testování klasifikace je území na jihovýchod od Moravského Krumlova. Zde se jedná převážně o zemědělské oblasti s různými stavy obdělání a druhů plodin na nich pěstovaných. Díky tomu jsou poměrně rozdílná, přestože data z *CLC* všechny označují jako jednotnou zemědělskou oblast.

Na výsledcích je vidět poměrně úspěšná klasifikace lesních ploch s přesností 79 %, stejně jako u osetých polí. Lze ale pozorovat, že zoraná pole byla klasifikována do třídy s občasnou vegetací, což značně zvyšuje její odchylku. V tomto případě je navržený klasifikátor citlivější na tyto rozdíly než rozdělení dat *CLC*. Hlavním důvodem je, že pro běžné druhy této analýzy se využívají data z daných ročních období, kde tyto rozdíly nejsou pozorovatelné a všechny zemědělské plochy vypadají velmi podobně.



Obrázek 23: Ukázka výsledků klasifikace větší oblasti se zemědělskou plochou v různých stavech obdělání a rozsáhlým zalesněným územím.

Tabulka 5: Matice záměn pro oblast obrázku č. 23.

	Les	Obilí	Vegetace	Těžba	Město	Přesnost
Les	108199	20901	66	2617	4955	79 %
Obilí	2111	79388	6341	487	11765	79 %
Vegetace	621	89367	6798	424	3979	6 %
Těžba	0	2643	90	131	38	4 %
Město	183	31597	166	81	2052	6 %

### Závěrečné zhodnocení přesnosti klasifikace

Z testovaných případů vychází řada poznatků, které do velké míry ovlivňují přesnost a sémantickou správnost klasifikace. Hlavním problémem jsou podle předpokládání vstupní data.

S ohledem na využití ortofoto snímků, které jsou často pořizovány s určitým náklonem, se v datech často objevují stíny. Ty, jak bylo ukázáno, jsou často klasifikovány jako vodní plochy. Dalším velkým problémem jsou rozdíly způsobené špatně zvoleným vegetačním obdobím při snímání. Tento jev způsobuje, že na polích je například různě zralé obilí, jiné plodiny nebo jsou porostlé a neosetá. V těchto případech se jedná o velmi odlišné povrchy. Díky tomu nelze očekávat, že by je všechny klasifikátor zařadil do jedné třídy jako zemědělské oblasti. Takový výstup by byl potřeba, aby srovnání s daty *CLC* bylo adekvátní.

Celkové srovnání s daty *CLC* je obtížné z pohledu různosti prostorového rozlišení a postprocessingu výstupních dat. Bohužel nebyla nalezena lepší data pro provedení srovnání přesnosti klasifikace, aby nebyla subjektivní. Samozřejmě, že klasifikátor



je zaměřen spíše na konfiguraci uživatelem, aby odpovídal jeho požadavkům. Na takovéto metodě je ale bohužel statistické vyhodnocení přesnosti zavádějící.

## 6.2 Zhodnocení postprocessingu výstupních dat

Prvním hodnoceným algoritmem je odstranění zanedbatelných a nechtěných polygonů. Pro ilustraci slouží obrázek č. 24. Na tom je vidět již dříve ukázaná vodní plocha, na které je loď. Lze předpokládat, že uživatele zajímá samotná vodní plocha, a ne tato loď. Z toho důvodu je potřeba ji odstranit. K tomu přesně slouží tento algoritmus. Na levé části obrázku je vidět výsledek klasifikace bez tohoto odstranění, což se projevuje tím, že plocha lodi není zahrnuta do polygonu vodní plochy. Na pravé straně obrázku vidíme efekt tohoto algoritmu, takže polygon už překrývá i loď a tím je obsažena celá vodní plocha.



Obrázek 24: Ukázka funkce algoritmu pro odstranění malých polygonů. V tomto případě modrá šipka označuje loď na vodní hladině která je na obrázku vpravo už odstraněna.

Druhou důležitou částí postprocessingu je rozdělení výše zmíněných mostů. Tento algoritmus řeší dvě základní situace.

První situací jsou dva rozsáhlé polygony propojené zúženým místem, tedy mostem. Cílem algoritmu je tento polygon rozdělit na dva v tomto úzkém místě. Tento jev je ukázán na obrázku č. 25. Na levé straně je vidět originální data, která byla klasifikována. Na pravé straně je znázorněna klasifikovaná oblast lomu. Ten se sestává z dvou velkých ploch spojených pravděpodobně cestou. V místě označeném červenou šipkou je vidět, kde algoritmus polygon rozdělil na dva. Díky tomu může uživatel pracovat s každým zvlášť.

Druhou situací, kterou má algoritmus řešit, je odstranění úzkých a dlouhých výstupků polygonů. Ty byly už dříve připodobněny například cestám vycházejícím z polí, vodním tokům vytékajícím z vodních těles a podobně. Na obrázku č. 26 vidíme konkrétně případ pole, ze kterého vycházejí remízy klasifikované do stejné třídy. Ty opět nejsou požadovaným výstupem, a proto je třeba je odstranit. V levé části obrázku je vidět výstup klasifikace bez využití algoritmu pro rozdělování mostů. Na pravé straně jsou výstupní data po jeho použití, kde můžeme pozorovat že tyto remízy jsou odstraněny z výsledného polygonu. Jsou označeny červenými šipkami.



Obrázek 25: Ukázka funkce algoritmu pro rozdělení mostů. Na obrázku je dva lomy spojené cestou. Algoritmus oba lomy oddělil v nejužším místě cesty kde ukazuje červená šipka.

Jako vedlejší efekt tohoto algoritmu se ukázalo i odstranění všech polygonů, které jsou podlouhlého tvaru s malou šířkou. Tento jev je vidět také na obrázku č. 26 v případě polygonu označeného žlutou šipkou. Je vidět že nepřísluší žádnému polygonu s větší plochou a po průchodu algoritmem je zcela odstraněn. Z tohoto plyne, že algoritmus automaticky odstraňuje tento druh ploch, do kterých mohou spadat například cesty, řeky a podobné.



Obrázek 26: Ukázka funkce algoritmu pro rozdělení mostů. Na obrázku je vidět pole, ze kterého vycházejí remízy klasifikované do stejné třídy. Ty jsou algoritmem odstraněny.

### Závěrečné zhodnocení postprocessingu výstupních dat

Oba tyto hlavní algoritmy postprocessingu splňují nastavené požadavky a je možné je parametrizovat uživatelem. V případě rozdělování mostů se sice vyskytl vedlejší efekt odstranění cest a podobných úzkých pokrývných oblastí, ale uživatel může tento jev eliminovat právě zmíněnou parametrizací.

Co se týče výpočetní rychlosti, vyžaduje odstranění malých polygonů průchod všemi detekovanými polygony, což znamená, že je velmi závislý na členitosti vstupních dat. V případě rozdělení mostů, se ale jedná o poměrně velmi náročnou operaci,

kteřou je třeba provést také pro každý polygon a pro oddělené části i vícekrát. Navíc by bylo třeba ho výrazně optimalizovat.

S ohledem na to, že se jedná o pouze experimentální implementaci jsou výsledky postprocessingu uspokojivé a dávají velký prostor dalšímu rozvoji v této oblasti.

### 6.3 Zhodnocení vhodnosti návrhu

Celkový návrh architektury a zvolených technologií se ukázal jako vhodný pro tuto formu aplikace. Architektura *MVC* v podobě webové aplikace je velmi jednoduchá z pohledu uživatele pro spuštění. Pro to stačí podle požadavků pouze standardní webový prohlížeč, který je v dnešní době běžnou součástí většiny operačních systémů. Jednoduché uživatelské prostředí a předem dostupná data z vytvořené datové báze jsou ihned k dispozici a nevyžadují žádnou další iniciativu ze strany uživatele.

Z pohledu dat byly objeveny některé nedostatky, které by bylo možné eliminovat například vytvořením vlastního mapového serveru, ve kterém by bylo možné zpracovat a poskytovat data ze satelitního snímkování. Bohužel formou *WMS* by stále nebylo možné přenášet dodatečná data například o infračerveném pásmu. Možností by bylo ho zakódovat do alfa kanálu obrazových dat. To by ale zase ztížilo uživateli práci s nimi.

Co se týče vhodnosti zvolených technologií na straně serveru, tak využití knihovny *GDAL*, pro načítání a exportování dat, se ukázalo jako ideální řešení. Jazyk *Python* umožnil poměrně jednoduše propojit všechny potřebné technologie a nabídl knihovny pro samotnou klasifikaci.

Optimalizace rychlosti prováděné klasifikace nebyla cílem práce. Z tohoto pohledu by ale samozřejmě bylo nejjednodušším řešením využití výkonnějšího technického vybavení. Dalším výrazným zrychlením by mohla být paralelizace výpočtů a algoritmů. Zde by bylo třeba velmi dobře vyřešit jejich synchronizaci, nebo vstupní data rozdělit do dlaždic, které by se klasifikovaly zvlášť. V tomto případě by ale zase nastávaly problémy v rámci jejich rozdělení.

### 6.4 Ekonomická zhodnocení navrženého řešení

Navržené řešení bylo určeno pouze pro otestování zvolené metody, nebylo tedy zamýšleno pro komerční využití. Z tohoto hlediska by ho ale určitě bylo možné také využít. S ohledem na zvolenou architekturu by se jednalo o začlenění do libovolného webového *GIS*, ale stejně tak by této funkcionality mohla využívat desktopová aplikace.

Hlavní výhodou je přesunutí výpočetních operací na stranu serveru, čímž se eliminuje potřeba výkonných zařízení na straně uživatele. Z ekonomického hlediska je tento princip daleko levnější s ohledem na stávající možnosti hostovaných virtuálních serverů. Bylo by také možné službu spustit ve formě cloudu pro škálování přidělených prostředků v závislosti na složitosti prováděných analýz.



Rozhraní pro spouštění analýzy je běžný *HTTP* požadavek, který obsahuje potřebná data a binární reprezentaci rastru. Při implementaci tohoto rozhraní je možné metodu začlenit do libovolného systému a tím nahradit navržené grafické rozhraní a integrovat ji do vlastního řešení.

Hlavním ekonomickým aspektem je založení metody na otevřeném software, což znamená, že jediné výdaje jsou na provoz. Ty představuje pronájem serveru, na kterém je služba a případně i uživatelské rozhraní spuštěno.

Nabízí se tedy dvě možnosti využití, kdy jednou je rozšíření funkcionality této aplikace a jejího využití jako samostatného produktu a nebo její začlenění do již existujícího produktu jak bylo zmíněno výše.

Přesnost, v porovnání s placeným software a odbornými metodami, je nesrovnatelná, ale požadavky stanovené na tuto metodu preferovaly jednoduchost pro uživatele. Z tohoto hlediska je přesnost dostačující.

## 7 Závěr

Cílem práce bylo navrhnout metodu pro analýzu využití území. Tento cíl byl splněn s důrazem na jednoduché a rychlé využití pro koncového uživatele. Implementovaná metoda nedosahuje přesnosti specializovaného software, nicméně to nelze očekávat s ohledem na použitá vstupní data. Datové podklady ČUZK byly zvoleny z důvodu snadné dostupnosti pro uživatele ve standardizovaném formátu. Odpadla tedy nutnost obstarání dat a jejich importu do aplikace. Implementovaná metoda navíc nabízí poměrně komplexní možnosti postprocessingu, díky kterým si lze výstup přizpůsobit konkrétním potřebám.

Klíčovým přínosem aplikace je inovativní využití klasifikace snímků za účelem zjednodušení výběru objektu na mapě (jezero, pole atp.). Dnes je běžně nezbytné vybrat požadovanou oblast ručně výběrem bodů na hranici zájmové oblasti. Tento přístup je zdlouhavý a nepřesný. Přístup automatického výběru popsáný v této práci poskytuje uživateli nesrovnatelně vyšší komfort a přesnost. Uživatel pouze rámcově vybere zájmovou oblast a implementovaný algoritmus automaticky určí přesnější hranice objektu. Uživatel se tak velice rychle dosáhne označení cílového objektu.

## 8 Reference

- [Agrawal, S., Gupta, R. D., 2014] AGRAWAL, S., GUPTA, R. D.. *Development and Comparison of Open Source based Web GIS Frameworks on WAMP and Apache Tomcat Web Servers* [online]. 2014 [cit. 2015-11-10]. Dostupné z: <http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-4/1/2014/isprsarchives-XL-4-1-2014.pdf>.
- [Caetano, M., Feranec, J., 2007] CAETANO, M., FERANEC, J.. *CLC2006 technical guidelines* [online]. 2007. Dostupné z: [http://www.eea.europa.eu/publications/technical\\_report\\_2007\\_17/at\\_download/file](http://www.eea.europa.eu/publications/technical_report_2007_17/at_download/file).
- [Conallen, J., 2003] CONALLEN, JIM. *Building web applications with UML*. 2nd ed. Boston: Addison-Wesley, c2003, xxi, 468 s. Addison-Wesley Object Technology Series, ISBN 1402080654.
- [CNES, 2002] CNES. *Orfeo ToolBox* [online]. 2002 [cit. 2015-11-10]. Dostupné z: <https://www.orfeo-toolbox.org/>.
- [Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V. a Böhner, J., 2015] CONRAD, O., BECHTEL, B., BOCK, M., DIETRICH, H., FISCHER, E., GERLITZ, L., WEHBERG, J., WICHMANN, V. a BÖHNER, J. (2015): *System for Automated Geoscientific Analyses (SAGA)* v. 2.1.4, Geosci. Model Dev., 8, 1991-2007, doi:10.5194/gmd-8-1991-2015.
- [Conrad, O., 2007] CONRAD, O.. *SAGA – Entwurf, Funktionsumfang und Anwendung eines Systems für Automatisierte Geowissenschaftliche Analysen*, Dissertation, Faculties Natural Sciences, Mathematics and Informatics, Faculty of Geosciences and Geography, 221 pp., 2007.
- [Conrad, O., 2005] CONRAD, O.. *Module Supervised Classification for Grids* [online]. 2005 [cit. 2015-11-10]. Dostupné z: [http://www.saga-gis.org/saga\\_module\\_doc/2.2.2/imagery\\_classification\\_0.html](http://www.saga-gis.org/saga_module_doc/2.2.2/imagery_classification_0.html).
- [Conrad, O., 2001] CONRAD, O.. *Module Cluster Analysis for Grids* [online]. 2001 [cit. 2015-11-10]. Dostupné z: [http://www.saga-gis.org/saga\\_module\\_doc/2.2.2/imagery\\_classification\\_1.html](http://www.saga-gis.org/saga_module_doc/2.2.2/imagery_classification_1.html).
- [Černý, M., 2009] ČERNÝ, M.. *Výroční zpráva k projektu SP2d19307 CzechTerra* [online]. 2009. Dostupné z: [http://www.czechterra.cz/documents/Zprava\\_CT\\_Nov2009.pdf](http://www.czechterra.cz/documents/Zprava_CT_Nov2009.pdf).
- [Exelis Visual Information Solutions, 2013] EXELIS VISUAL INFORMATION SOLUTIONS. *Using ENVI and Geographic Information Systems (GIS)* [online]. 2013 [cit. 2015-11-10]. Dostupné z: <http://www.exelisvis.com/Learn/WhitepapersDetail/TabId/802/ArtMID/2627/ArticleID/9892/Using-ENVI-and-Geographic-Information-Systems-GIS.aspx>.

- [Harikrishnan, Poongodi, 2015] HARIKRISHNAN, R. a POONGODI, S.. *Satellite image classification for IRS data based on supervised and unsupervised algorithms using ERDAS* [online]. 2015 [cit. 2015-11-10]. Dostupné z: <http://www.internationaljournalssrg.org/IJECE/2015/Special-Issues/ICETSH/IJECE-ICETSH-P120.pdf>.
- [Haverbeke, M., 2014] HAVERBEKE, M.. *Eloquent JavaScript* [online]. 2014 [cit. 2015-11-10]. Dostupné z: [http://eloquentjavascript.net/Eloquent\\_JavaScript.pdf](http://eloquentjavascript.net/Eloquent_JavaScript.pdf).
- [Hájek, F., Svoboda, M., 2007] HÁJEK, F. a SVOBODA, M.. *Vyhodnocení odumírání horského smrkového lesa na Trojmezí (NP Šumava) metodou automatizované klasifikace leteckých snímků* [online]. 2007. Dostupné z: [http://www.npsumava.cz/storage/vyzkum/SGpdf/SG13\\_1\\_HajekSvoboda.pdf](http://www.npsumava.cz/storage/vyzkum/SGpdf/SG13_1_HajekSvoboda.pdf).
- [HEXAGON, 2015] HEXAGON. *ERDAS IMAGINE® 2015 Product Features and Comparisons* [online]. 2015 [cit. 2015-11-10]. Dostupné z: <http://www.hexagongeospatial.com/technical-documents/erdas-Imagine-2015-product-description>.
- [Hsu, L S., Obe, R O., 2014] HSU, L S. a OBE, R O. *PostGIS in Action, Second Edition*. USA: Manning Publications Co, 2014. 625 s. ISBN 978-1-6172-9139-5.
- [Hunt, E. R. Jr., Doraiswamy, P. C., 2013] HUNT, E. R. JR., DORAISWAMY, P. C.. *A visible band index for remote sensing leaf chlorophyll content at the canopy scale* [online]. 2013 [cit. 2015-11-10]. Dostupné z: <http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=2161&context=usdaarsfacpub>.
- [Landa, M., 2015] LANDA, M.. *Knihovna GDAL, PROJ.4* [online]. 2015 [cit. 2015-11-10]. Dostupné z: <http://geo.fsv.cvut.cz/~gin/yfsg/Free-Software-GIS-03-gdal-proj.pdf>.
- [Lillesand, T., Kiefer, R W., Chipman, J., 2008] LILLESAND, T., KIEFER, R W. a CHIPMAN, J. *Remote sensing and image interpretation*. New York: John Wiley & Sons, 2008. 804 s. ISBN 978-0-470-05245-7.
- [Neteler, Markus a Mitsova, Helena , 2004] NETELER, MARKUS a MITSOVA, HELENA. *Open source GIS: a GRASS GIS approach*. 2nd ed. Boston: Kluwer Academic Publishers, c2004, xxvi, 401 p. Kluwer international series in engineering and computer science, SECS 773. ISBN 1402080654.
- [OTB Development Team, 2015] OTB DEVELOPMENT TEAM. *The ORFEO Tool Box Software Guide* [online]. 2015 [cit. 2015-11-10]. Dostupné z: <https://www.orfeo-toolbox.org//packages/OTBSoftwareGuide.pdf>.
- [Pomahačová, M., 2012] POMAHAČOVÁ, M.. *Možnosti využití DPZ při monitoringu luční vegetace a managementových zásahů v Krkonoších* [online].

2012. Dostupné z: <http://druzicovesnimky.cz/wp-content/uploads/6-DPZ-pro-monitoring-lucni-vegetace-a-managementovych-zasahu-v-Krkonosich.pdf>.
- [QGIS Development Team, 2014] QGIS DEVELOPMENT TEAM. *QGIS User Guide* [online]. 2015 [cit. 2015-11-10]. Dostupné z: <http://docs.qgis.org/2.8/pdf/en/QGIS-2.8-UserGuide-en.pdf>.
- [Rodríguez Luaces, M. A., 2004] RODRÍGUEZ LUACES, M. A.. *A Generic Architecture for Geographic Information Systems* [online]. 2004 [cit. 2015-11-10]. Dostupné z: <http://lbd.udc.es/Repository/Thesis/432677332J.pdf>.
- [Santiago, A., 2015] SANTIAGO, A.. *The book of OpenLayers 3* [online]. 2015 [cit. 2015-11-10]. Dostupné z: <http://samples.leanpub.com/thebookofopenlayers3.pdf>.
- [Stopper, R., 2013] STOPPER, R.. *Open Geospatial Consortium (OGC) and Web Services (WMS, WFS)* [online]. 2013 [cit. 2015-11-10]. Dostupné z: [http://www.e-cartouche.ch/content\\_reg/cartouche/webservice/en/text/webservice.pdf](http://www.e-cartouche.ch/content_reg/cartouche/webservice/en/text/webservice.pdf).
- [Theodoridis, S., Koutroumbas, K., 2009] THEODORIDIS, S. a KOUTROUMBAS, K. *Pattern recognition*. 4. vyd. Burlington, Mass.: Academic Press, 2009. 961 s. ISBN 978-1-59749-272-0.
- [Tibshirani, R., Hastie, T., Friedman, J., 2009] TIBSHIRANI, R., HASTIE, T. a FRIEDMAN, J. *The Elements of Statistical Learning 2nd edition*. USA: Springer, 2009. 768 s. ISBN 0-387-84857-6.
- [Wikipedia, 2015] WIKIPEDIA: THE FREE ENCYCLOPEDIA. *Leaflet (software)* [online]. San Francisco (CA): Wikimedia Foundation, 2015 [cit. 2015-11-10]. Dostupné z: <http://samples.leanpub.com/thebookofopenlayers3.pdf>.
- [Woychowsky, E., 2007] WOYCHOWSKY, E. *Ajax: creating Web Pages with Asynchronous JavaScript and XML*. Upper Saddle River, NJ: Prentice Hall, c2007, xxii, 384 p. ISBN 9780132272674.

# Přílohy



```
50         dst = base_ring[1].Distance(  
51             intersection.GetGeometryRef(ring_id)  
52         )  
53     if dst < min_dist[0]:  
54         min_dist = [  
55             ring_id,  
56             intersection.GetGeometryRef(ring_id).GetGeometryRef(0)  
57         ]  
58     clip_dst = [99999, None]  
59     for pid in range(0, min_dist[1].GetPointCount()):  
60         tmp_point = ogr.Geometry(ogr.wkbPoint)  
61         tmp_point.AddPoint(*min_dist[1].GetPoint_2D(pid))  
62         b_bound = buff.Boundary()  
63         if not b_bound.Intersects(tmp_point.Buffer(0.00000001)):  
64             dst = point.Distance(tmp_point)  
65             if dst < clip_dst[0]:  
66                 clip_dst = [dst, tmp_point]  
67     if clip_dst[1]:  
68         destroy_intervals.append([  
69             list(point.GetPoint_2D()),  
70             list(clip_dst[1].GetPoint_2D())  
71         ])
```



## 10 CD

### Obsah

1. Zdrojové kódy všech aplikace
2. Zdrojové soubory textové části
3. Obrazová data využitá v textové části
4. Textová část ve formátu PDF