

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Jiří Palas

Studijní program: Otevřená informatika (magisterský)

Obor: Počítačové vidění a digitální obraz

Název tématu: Automatizovaná analýza částic v mikroskopických snímcích

Pokyny pro vypracování:

1. Seznamte se se stávajícími metodami pro automatickou detekci a klasifikaci částic v dvourozměrných mikroskopických snímcích.
2. Na základě provedené analýzy navrhněte a implementujte vhodný způsob (metodu, algoritmus) pro automatizovaný předvýběr hledaných částic v mikroskopickém snímku. Současně navrhněte a implementujte způsob, jakým může laborant ručně opravit automatizovaně provedený výběr a označit jednotlivé třídy částic.
3. Navrhněte atributy (příznaky) popisující částice, které mohou sloužit pro automatizovanou identifikaci různým tříd částic, a implementujte metody pro jejich extrakci.
4. Implementujte vhodný klasifikátor (např. K-NN, Bayesův klasifikátor, SVM, ...), který umožní automatizovaně klasifikovat částice do příslušných tříd na základě extrahovaných atributů v dalších mikroskopických snímcích.
5. Implementujte statistické metody, které budou následně aplikovány na vybrané částice, pro vypočet požadovaných charakteristik, jako je např. počet částic ve třídách, zda částice tvoří/netvörí shluky, kde se částice nachází v prostoru ad.
6. Pro implementaci metod použijte programovací jazyk Java v prostředí platformy NetBeans a využijte knihovnu OpenCV.
7. Navržené a implementované metody ověřte na reálných datech a získané výsledky vyhodnoťte. Navrhněte možná další zlepšení.

Seznam odborné literatury:

- [1] Szeliski R. 2010. Computer Vision: Algorithms and Applications (1. vydání). Springer-Verlag New York, Inc., New York, NY, USA.
[2] Ashbrook A., Thacker N.A. Tutorial: Algorithms for 2-Dimensional Object Recognition. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 1998.

Vedoucí diplomové práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

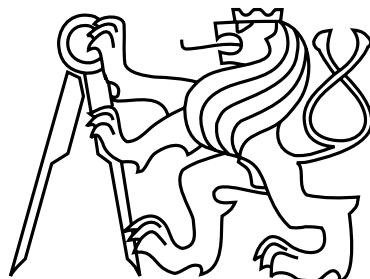
Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra kybernetiky



Diplomová práce

**Automatizovaná analýza částic v mikroskopických
snímcích**

Bc. Jiří Palas

Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina, PhD.

Studijní program: Otevřená informatika, Magisterský

Obor: Počítačové vidění a digitální obraz

11. května 2015

Poděkování

Na tomto místě bych rád poděkoval maminec za vaření obědů a večeří během krušných hodin psaní. Rád bych poděkoval svému vedoucímu za dostatek trpělivosti. A dále všem, kteří mě obdařují každý den úsměvem.

Prohlášení

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

V Praze dne 11. 5. 2015

Abstract

The aim of the thesis is to provide tool for automatic analysis of nanoparticles in microscopy images. Main goal is to design and implement functional software which deals with automatic analysis of nanoparticles in microscopic images. Thesis provides analysis for the state-of-art image processing and machine-learning methods suitable to achieve the main goal. On the base of the analysis is designed and implemented an algorithm for automatic detection and classification of nanoparticles. Implemented algorithms are supplemented with user interface developed in NetBeans Platform.

klíčová slova: nanoparticles, segmentation, classification, microscopy

Abstrakt

Tato práce se zabývá automatizací analýzy nanočástic v mikroskopických snímcích. Hlavním cílem je návrh a implementace software zaměřeného na automatizovanou analýzu nanočástic v mikroskopických snímcích. Práce se analyzuje současné metody zpracování obrazu a strojového učení využitelné pro automatickou detekci, klasifikaci a analýzu nanočástic v mikroskopických snímcích. Na základě provedené analýzy je navržen a implementován algoritmus pro automatickou detekci a klasifikaci částic. Implementované algoritmy jsou začleněny do funkční aplikace, založené na platformě NetBeans, která poskytuje rozhraní k další analýze.

klíčová slova: nanočástice, segmentace, klasifikace, mikroskopie

Obsah

0.1 Motivace	1
0.2 Přehled	2
1 Analýza	3
1.1 Specifikace	3
1.2 Data	4
2 Detekce nanočástic	7
2.0.1 Úvod do problematiky	7
2.0.2 Popis použitého řešení	8
2.0.3 Výsledky segmentace	9
3 Klasifikace	11
3.1 Úvod do problematiky	11
3.2 Trénovací množina a volba příznaků	13
3.3 Volba klasifikačního algoritmu	13
3.4 Výsledky klasifikace	17
3.4.1 Vstupní data	17
3.4.2 Vzorek 1	18
3.4.3 Vzorek 2	19
3.4.4 Vzorek 3	20
3.4.5 Vzorek 4	21
3.4.6 Vzorek 5	22
3.4.7 Vzorek 6	23
3.4.8 Celkový přehled	24
4 Návrh a realizace aplikace	25
4.1 Způsob práce s daty	25
4.2 Uspořádání aplikace	26
4.2.1 Editační panel	27
4.2.1.1 Editor snímků	27
4.2.1.2 Panel analýzy	27
4.2.2 Přehled projektů	28
4.2.3 Přehledu klasifikátorů	29
4.2.4 Inspekční panel	29
4.3 Uchování dat	30
4.3.0.1 Ukládání dat snímků	32

4.3.0.2 Ukládání dat klasifikátoru	33
4.4 Realizace aplikace	33
4.4.1 Vývojové prostředí	34
4.4.1.1 Netbeans Platform	34
4.4.2 Podpůrné knihovny	34
5 Diskuze	35
6 Závěr	37
6.1 Shrnutí práce	37
6.2 Další směrování	38
A Seznam použitých zkratek	43
B Obsah přiloženého CD	45
C Snímky obrazovky	47

Seznam obrázků

1.1	Příklad vyhodnocovaného snímku.	4
2.1	Příklad výstupu segmentačního algoritmu na vzorku 5.	10
2.2	Příklady chybně provedené segmentace.	10
3.1	Prostor příznaků pro částici, která by byla klasifikována jako typ tyčinka.	15
3.2	Prostor příznaků pro částici, která by byla klasifikována jako typ koláč.	16
3.3	Prostor příznaků pro částici, která by byla klasifikována jako typ kolečko.	16
3.4	Množina trénovacích dat.	17
3.5	Výsledek klasifikace částic pro vzorek 1.	18
3.6	Výsledek klasifikace částic pro vzorek 2.	19
3.7	Výsledek klasifikace částic pro vzorek 3.	20
3.8	Vyhodnocení klasifikace částic pro vzorek 4.	21
3.9	Výsledek klasifikace částic pro vzorek 5.	22
3.10	Výsledek klasifikace částic pro vzorek 6.	23
4.1	Přehled rozložení aplikace.	26
4.2	Ukázka panelu přehledem projektů a snímků.	28
4.3	Ukázka panelu s přehledem klasifikátorů.	29
4.4	Ukázka panelu s informacemi o snímku.	30
4.5	Ukázka panelu s informacemi o snímku.	31
4.6	Příklad souborové a složkové struktury projektu.	31

Seznam tabulek

2.1 Výsledky segmentace.	9
3.1 Vyhodnocení klasifikace částic pro vzorek 1.	18
3.2 Vyhodnocení klasifikace částic pro vzorek 2.	19
3.3 Vyhodnocení klasifikace částic pro vzorek 3.	20
3.4 Vyhodnocení klasifikace částic pro vzorek 4.	21
3.5 Vyhodnocení klasifikace částic pro vzorek 5.	22
3.6 Vyhodnocení klasifikace částice pro vzorek 6.	23
3.7 Vyhodnocení klasifikace pro všechny vzorky.	24

Úvod

0.1 Motivace

V současné době se čistý a aplikovaný výzkum v biologii, medicíně a nanotechnologiích opírá o pozorování procesů uvnitř buněk, tkání a orgánů. Mezi takové procesy patří pozorování distribuce léčiv a genů, lokalizaci nádorů, toxikologickému testování, prověřování zdravotních rizik, reakce na chirurgické implantáty atd. [25] [22] [13].

Pro pozorování těchto procesů se využívají nano-částice (NP). NP jsou částice o rozměrech mezi 1-100 nm. Mezi nejčastěji používané NP patří částice vytvořené chemických sloučenin s kovy. Různé typy NP se odlišují buď tvarem nebo velikostí. Na základě svých vlastností pak NP vykazují specifické chování ve výše zmíněných procesech [17].

NP mají velice malý rozměr na to aby se daly snímat pomocí světelné mikroskopie. Proto se k jejich pozorování používají biologické zobrazovací metody pracující ve velikém rozlišení. Mezi takové metody patří například transmisní elektronová mikroskopie (TEM), rastrovací elektronová mikroskopie (SEM) a atd. [25] [22] [13]. Výsledkem těchto zobrazovacích metod je snímek nebo série snímků.

Snímky jsou následně vědci či laboranty vyhodnocovány. Pracovníci Akademie věd České republiky na Ústavu Molekulární geneticky (UMG) uvedli, že postrádají kvalitní, specializovaný a přitom jednoduchý nástroj pro vyhodnocování snímků s nanočásticemi.

Tato práce si klade za cíl vytvořit aplikaci pro vyhodnocování snímků. Práce klade důraz na vytvoření co největší úrovně automatizace v aplikaci. Tak aby došlo k celkovému zkvalitnění a zrychlení vyhodnocování jednotlivých snímků. Základem pro vyhodnocování snímků je lokalizace jednotlivých částic. Dále jejich úspěšná klasifikace do jednotlivých tříd (kategorií) podle typů NP. Následná expertní analýza laborantem.

0.2 Přehled

Kapitole práce definuje požadavky na aplikaci a popisuje data s částicemi.

Kapitola *analýza* se zabývá problémem detekce částic. Nejprve jsou rozebrány možné způsoby jak částice v obraze detektovat. Následně je popsán vybraný algoritmus pro jejich detekci. Nakonec jsou uvedeny výsledky detekce častic na vybraných snímcích.

Kapitola *klasifikace* se zabývá problematikou klasifikace částic. Jsou zmíněny jednotlivé přístupy ke klasifikaci. Dále je popsán vybraný klasifikační algoritmus. Nakonec jsou uvedeny výsledky klasifikace častic na vybraných snímcích.

Kapitola návrh a realizace popisuje co stálo za návrhem uživatelského rozhraní aplikace. Jsou detailně rozebrány jednotlivé části aplikace a jejich funkcionality. Kapitola se také zabývá realizací aplikace a použitým software.

Kapitola 1

Analýza

Tato kapitola nejprve popisuje základní definované požadavky na funkčnost aplikace. Dále pak definuje data, se kterými aplikace pracuje.

1.1 Specifikace

Na základě návštěvy pracoviště UMG jsou popsány požadavky na aplikaci pro automatizovanou analýzu mikroskopických snímků obsahující nanočástice:

Typická práce laboranta vyhodnocujícího NP se sestává z následujících částí:

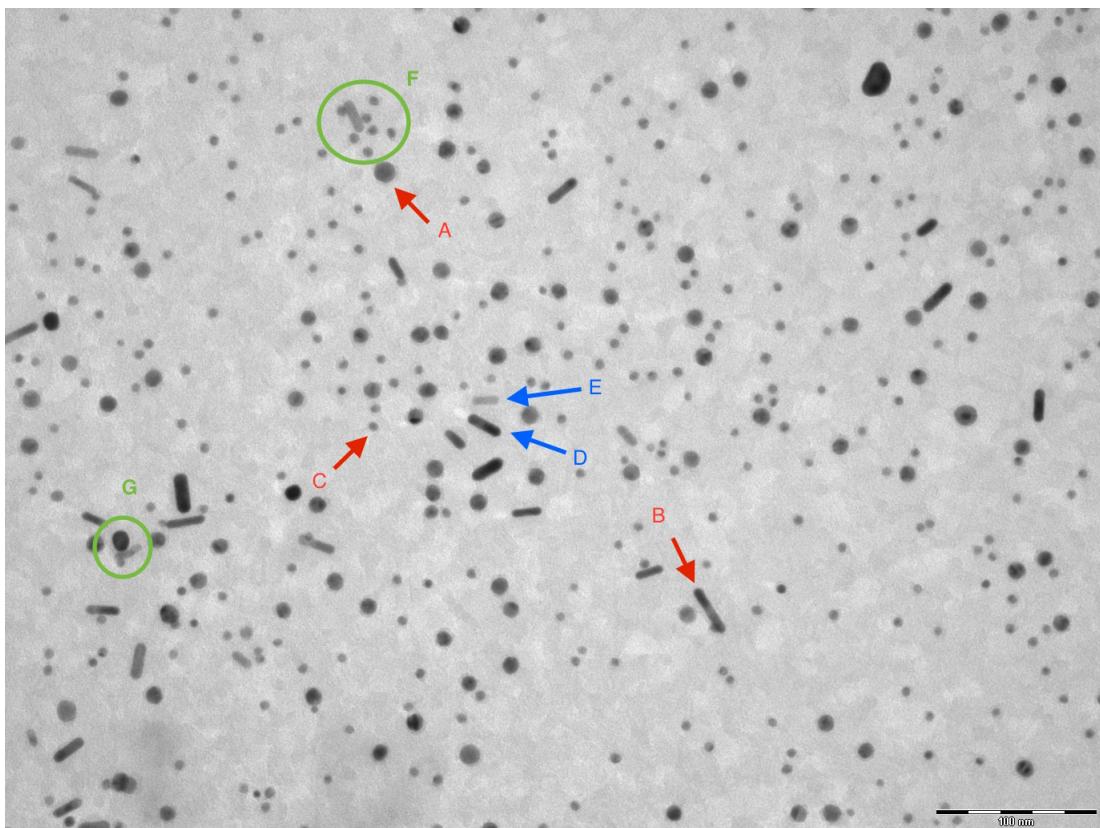
- příprava biologického vzorku
- nasnímání vzorku pomocí elektronového mikroskopu
- identifikace částic
- klasifikace (kategorizace) vzorků
- analýza vzorku

Práce se zaměřuje na automatizaci nebo částečnou automatizaci bodů 3-5. Ty se pak podrobněji zahrnují

- načítání mikroskopické obrázky ve formátech TIFF, JPEG
- zobrazení načtených obrázků
- schopnost detekovat NP v obrázku
- schopnost nalézt těžiště NP
- zobrazení detekovaných NP v obrázku
- možnost označit jednotlivé třídy NP

- automatizace označování NP
- vizuální shluková analýza NP

1.2 Data



Obrázek 1.1: Příklad vyhodnocovaného snímku. Červené šipky ukazují různé tvary vyhodnocovaných NP. Tvar A: Koláč. Tvar B: Tyčinka. Tvar C: Kolečko. Částice E a D demonstруjí možné rozdíly v jasech částic. Region G zobrazuje příklad překryvu částic. Region F zobrazuje příklad shlukování částic.

Aplikace se zabývá vyhodnocováním NP obsažených ve snímcích jako na obr. 1.1. Snímky jsou 8-bitové jednokanálové (šedotónové) obrázky s rozlišením mezi 2 a 3 megapixely.

Částice ve snímku můžeme pozorovat jako kontrastní útvary s nízkým jasem (tmavě šedé až černé) na světlém pozadí.

Snímané vzorky jsou v mikroskopu nerovnoměrně osvětleny. To se projevuje na nerovnoměrném rozložení jasu v obrázku. Viz. levý horní roh a pravý horní roh na obr. 1.1. Dalším negativním jevem, který znesnadňuje detekci je tendence NP tvořit shluky viz. region F na obr. 1.1 a překrývat se viz. region G

na obr. 1.1. Navíc ani částice nemají jednotnou úroveň jasu viz. částice E a D na obr. 1.1.

Práce rozlišuje tři typy částic A, B, C viz. obr. 1.1. Pracovní názvy těchto částic jsou koláč (A), tyčinka (B) a kolečko (C). K rozlišení těchto částic uvažujeme informaci o tvaru a velikosti. Jednotlivé částice lze považovat za konvexní útvary [38]. NP mohou být v obrázku libovolně orientované. V určitých částech snímku splývají s pozadím více viz částice D na obr. 1.1 a v některých méně viz částice E na obr. 1.1. Částice nemají výraznou texturu. Snímek typicky obsahuje 300–700 částic.

Na základě provedené charakteristiky obrazu a částic je vybrán vhodný detekční a klasifikační algoritmus viz kapitola 2 a kapitola 3.

Kapitola 2

Detekce nanočástic

Tato kapitola se zabývá problematikou segmentace nanočástic z jednotlivých mikroskopických snímků.

2.0.1 Úvod do problematiky

Cílem detekčního algoritmu je úspěšně nalézt jednotlivé částice v obrazu. Dále je třeba nalezené částice popsat modelem na jehož základě bude částice možné klasifikovat pomocí klasifikačního algoritmu a dále vyhodnocovat.

Pro úspěšnou detekci částic je nejprve potřeba jejich segmentace. Segmentace je proces při kterém jsou ze snímku odděleny jednotlivé objekty a části od pozadí. Jedná se často o nejsložitější a nejdůležitější krok při analýze obrazu.

Navzdory důležitosti, existuje pouze limitované množství zdrojů o automatické segmentaci nanočástic využitelných pro náš případ. Všechny nalezené zdroje se zabývají převážně detekcí NP kruhovitého tvaru [23][15][11]. Na druhou stranu ale existuje rozsáhlý výzkum zabývající sesegmentací biologických buněk. V následujícím textu je shrnuta literatura zabývající se segmentací biologických buněk. Tento směr výzkumu je přímo spjat s jedním z cílů této práce [26].

Velice často používanou metodou pro segmentaci buněk je prahování založené na charakteristikách histogramu obrázku. Přehled prahovacích technik uvádí [30]. Pro obdržení uspokojivého výsledku segmentace za pomoci prahování je ale zapotřebí pozadí obrázku s neměnnou intenzitou. Je také třeba pamatovat na to, že malá změna v úrovni prahu může mít negativní dopad na další analýzu obrázku [36].

Adaptivní prahování nebo také lokální automatické prahování může být řešení pro prahování obrázku ve kterém intenzita pozadí kolísá. Tato metoda ale stále neřeší problémy spojené s segmentací objektů které tvoří shluky. Prahování nemusí být finálním krokem v segmentačním procesu. Naopak často bývá

využíván právě jako první krok před dalším zpracování např. pomocí morfologických operací [16] [36].

Další hojně užívanou metodou pro segmentaci buněk je tzv. **watershed** a jeho odvozeniny. Metoda nejprve nalezne body (markery) ukazující přibližnou polohu buněk a následně segmentuje obrázek na několik oblastí ovlivněných markery [21] [35][4][36].

Graph-cut metody jsou také aplikovány na segmentaci buněk. Metoda zkonstruuje graf z obrázku kde každý pixel odpovídá jednotlivým vrcholům grafu. Každý pár vrcholů je spojen se sousedními vrcholy hranou jejíž váha odpovídá podobnosti intenzit sousedních pixelů. Segmentace pak hledá normalizovaný minimální řez grafem [31]. Avšak tento přístup špatně separuje překrývající se buňky, obzvláště pokud je jejich intenzita podobná. Existuje úprava která tento problém řeší, funguje však pouze na částice buňky kruhového tvaru [26].

Mezi další segmentační metody používané při segmentaci buněk se řadí **aktivní kontury** [6] [24]. Aktivní kontury jsou originálně navržené k segmentaci jednoho objektu s komplikovaným tvarem. Avšak po úpravě vícefázový algoritmus může být použit k segmentaci více nepřekrývajících se objektů. Článek [26] použití na segmentaci překrývajících se buněk nedoporučuje.

2.0.2 Popis použitého řešení

Na základě předchozí analýzy byla ozkoušena segmentace metodami prahování *watershed* a adaptivního prahování.

Adativní prahování překvapivě produkuje od začátku dobrou míru segmentace. Avšak nedokáže oddělit částice, které se dotýkají nebo překrývají. Proběhl pokus zpřesnit výsledky pomocí watershed algoritmu. Avšak výsledek ke zpřesnění výstupu segmentace nedošlo, protože nebyl nalezen korektní způsob umístění markerů. Bylo vyzkoušeno umístění na základě distanční transformace [20] a na základě *ultimátní eroze pro konvexní sety* [26]. Mnou navržené řešení se tedy ustálilo na níže popsaném řešení.

Snímky s NP obsahují malé množství šumu, proto je obrázek předzpracován mediánového filtru s oknem o velikosti 3x3 pixely.

Z důvodů problémů s jasem popsaných dříve je pro úvodní segmentaci vybrán algoritmus lokálního adaptivního prahování. Algoritmus volí práh pro každý pixel individuálně na základě rozsahu hodnot intenzity v lokálním okolí [28]. Výstupem algoritmu je pak binární obrázek.

V této aplikaci se hodnota prahu v bodě (x, y) získává jako:

$$t(x, y) = \sum_{i=x-(B-1)/2}^{x+(B-1)/2} \sum_{j=y-(B-1)/2}^{y+(B-1)/2} f(i, j) - C \quad (2.1)$$

Kde $t(x, y)$ je hodnota prahu v bodě (x, y) , $f(i, j)$ hodnota intenzity v bodě (i, j) , B velikost okna a C konstanta. Velikost okna musí být liché číslo.

Binární obrázek získaný prahováním je následně upraven pomocí morfologické operace otevření s jádrem:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.2)$$

V binarní obrázek, upraveným morfologickou operací jsou následně prostřednictvím algoritmu [34] nalezeny obrysy jednotlivých NP. Obrys je reprezentován vektorem bodů, kterého ho tvoří.

K detekci byla využita implementace adaptivní filtrace a algoritmu [34] z OpenCV[9]. Adaptivní filtrace využívá funkce *adaptiveThresh* a algoritmus na vyhledávání obrysů pak funkci *findContours*.

2.0.3 Výsledky segmentace

Segmentace a její vyhodnocení výsledků probíhalo na datech snímcích pocházejících z UMG. K dispozici je 7 mikroskopických snímků obsahujících variující tvary částic. Ve vzorcích 1, 2, 7 se vyznačují malou mírou shlukování částic. Naopak ve vzorcích 3, 4, 5, 6 jsou částice poskládány blíže sobě a některé se překrývají.

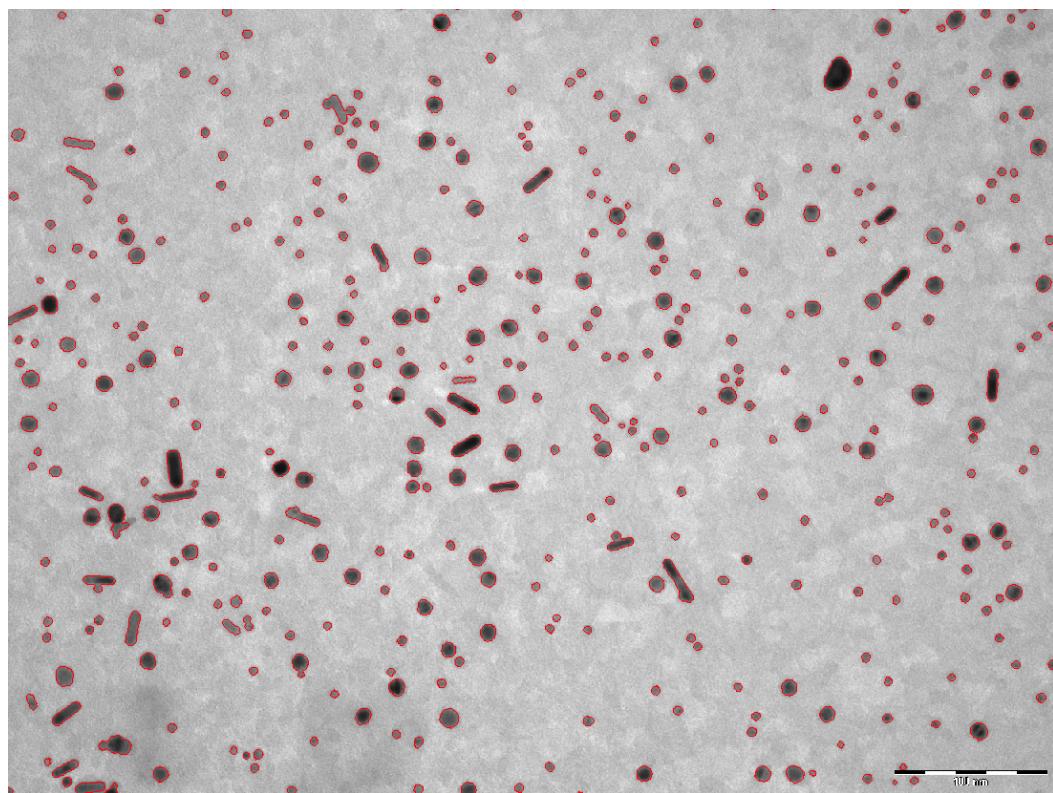
Výstupy segmentace algoritmu jsou prezentovány pomocí barevného značení, kde kontura nalezené částice je zobrazena červeně.

Příklad výstupu segmentačního algoritmu je možné pozorovat na obr. 2.1. Detaily chybně provedené segmentace jsou znázorněny na obr. 2.2.

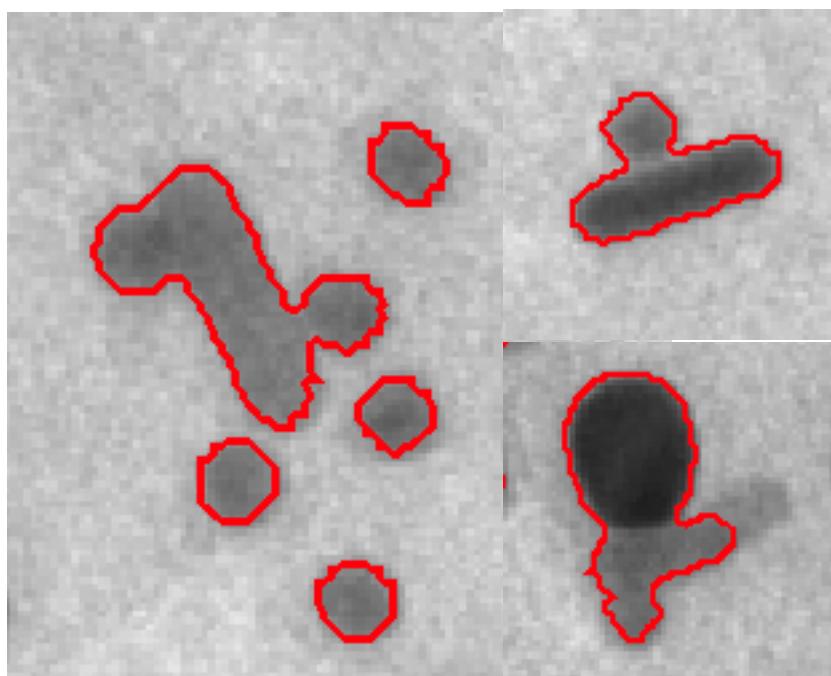
Vyhodnocení výstupu segmentace je možné pozorovat v tabulce 2.1. Nejčastějším problémem při segmentaci je segmentace spojených částic typu tyčinka a kolečko.

Vzorek	Chybná segmentace	Očekáváno	Přesnost
1	1	353	0.997
2	1	383	0.997
3	29	435	0.932
4	31	458	0.932
5	27	426	0.936
6	8	411	0.980
7	1	338	0.997
Celkem	98	2809	0.965

Tabulka 2.1: Výsledky segmentace.



Obrázek 2.1: Příklad výstupu segmentačního algoritmu na vzorku 5.



Obrázek 2.2: Příklady chybně provedené segmentace.

Kapitola 3

Klasifikace

Tato kapitola se se zabývá problematikou klasifikace detekovaných částic. V úvodu kapitoly je rozvedena problematika klasifikace a uvedeny metody strojového učení, které je možné využít při řešení této problematiky. Následně je rozveden přístup pro klasifikaci částic do jednotlivých tříd A, B nebo C, použitý v aplikaci V závěru kapitoly pak jsou uvedeny jednotlivé výsledky klasifikace pomocí zvoleného K-NN klasifikátoru.

3.1 Úvod do problematiky

Klasifikace je jedním z nejčastějších problémů se kterými se člověk setkává. Klasifikační problém se objevuje vždy, když určitý objekt potřebuje být začleněn do předdefinované skupiny [40].

Mezi metody, které řeší klasifikační problémy patří metody strojového učení. V biologii bylo strojové učení již aplikováno na řadu problému v oblastech vývoje léků [27], analýza sekvencí DNA [10], proteomika [39] a další. Mimo biologii se strojové učení využívá např. pro rozpoznávání textu, obličejů atd. Metody strojového učení dělíme na dvě skupiny, učení bez učitele a učení s učitelem.

První skupinou jsou metody tzv. *učení bez učitele*. Tyto metody se obvykle využívají při zkoumání významu dat o kterých máme málo informací nebo nedokážeme vytvořit trénovací metody. Cílem učení bez učitele je spojovat skupiny dat do shluků na základě měření podobnosti nebo usnadnění následného vytěžování dat redukcí jejich dimensionality.

Druhou skupinu tvoří tzv. *metody učení s učitelem*. V tomto přístupu, uživatelé generují, z celé sady dat, data trénovací pomocí anotace reprezentativních vzorků. Anotaci dat se rozumí přiřazení vzorku do skupiny. Na základě těchto trénovacích dat je poté algoritmus strojového učení schopen podle definovaných pravidel a postupů zařadit do jednotlivých tříd i dosud neznámé vzorky dat.

Protože jednotlivé částice jsou klasifikovány na základě anotací poskytnutých laborantem, jsou nadále uvažovány pouze metody strojového učení s učitelem.

V případě strojového učení s učitelem definuje trénovací data člověk, který je považován za neomylného expertsa. Data jsou definována anotováním vybraných příkladů z celkové množiny dat. V našem případě tvoří množinu všech dat detekované částice z jednoho nebo několika snímků. Trénovací množina je pak vybírána z této množiny anotováním jednotlivých častic. Ve strojovém učení proces trénování slouží k získání vnitřních parametrů klasifikačního modelu na jejichž základě jsou pak jednotlivé příklady klasifikovány.

Mezi současné klasifikační algoritmy učení s učitelem patří např. rozhodovací stromy, Support Vector Machine (SVM), naivní Bayes, AdaBoost, neuronové sítě nebo algoritmus k-nejbližších sousedů (K-NN) a další [12].

Bayesovská rozhodovací metodika spoléhá na statistický pravděpodobnostní model. Podle tohoto modelu je pak určena jednotlivá příslušnost instancí do tříd [40].

Metoda *rozhodovacích stromů* je založena na víceúrovňovém rozhodování. Spočívá na principu kdy je komplexní rozhodovací problém rozložen na více jednodušších rozhodovacích problémů na jednotlivých úrovních [29].

Support vector machines (SVM) je metoda která se snaží nalézt dělící nadroviny, která rozdělí data rozdílných tříd, tak aby vzdálenost nejbližších trénovacích příkladů z různých tříd byla co největší. Protože data různých tříd nemusí být vždy oddělitelná pomocí nadroviny většina implementací SVM je založena na *měkkém okraji* (soft-margin), který umožňuje špatnou klasifikaci. Špatně klasifikované příklady jsou pak váženy cenovou funkcí. SVM je robustní na příznaky s vysokým podílem šumu [32].

Adaptive bootsing (AdaBoost) je založen na kombinaci několika slabých klasifikátorů k formování silného klasifikátoru iterativním přidáváním a vážením jednotlivých slabých klasifikátorů. Příkladem slabého klasifikátoru může být např. prahová funkce. Nevýhodou AdaBoostu je náchylnost na data s vysokým podílem šumu a tzv. *outlierů*.

Neuronové sítě je skupina algoritmů inspirovanými neuronovými spojeními v lidském mozku. Skládá se z umělých neuronů, jejichž inspirace pochází právě z biologických neuronů. Jednotlivé neurony jsou navzájem propojeny a navzájem si předávají signály, které se transformují pomocí přenosových funkcí. Neuron může mít mnoho vstupů ale pouze jeden výstup. Příkladem neuronu může být např. perceptron [40].

Typické kroky při tvorbě klasifikátoru učení s učitelem jsou:

- Určení trénovací množiny dat
- Zisk trénovací množiny dat
- Tvorba příznaků
- Volba trénovacího algoritmu
- Vyhodnocení výsledků

3.2 Trénovací množina a volba příznaků

Trénovací množinu dat tvoří laborantem anotované částice s přiřazenou příslušností do A, B nebo C. Částice jsou získávány jako výstup detekčního algoritmu. Anotaci a výběr trénovací množiny pak zajistí laborant skrze rozhraní aplikace.

Jednotlivé částice od sebe odlišujeme tvarem, definovaným vektorem bodů, a velikostí plochy v pixelech. Tyto dva příznaky použijeme k zformování vektoru příznaků na který budeme aplikovat klasifikační algoritmus.

3.3 Volba klasifikačního algoritmu

Na základě malého počtu příznaků, tvaru prostoru příznaků a průběžných experimentálních výsledků byl ke klasifikaci je vybrán algoritmus **k-nejbližších sousedů (K-NN)**.

K-nejbližších sousedů (K-NN) je neparametrická metoda využívaná pro klasifikaci a regresi. Spočívá ve vyhledávání k nejbližších sousedů v prostoru příznaků. K-NN potřebuje pro svou funkci definovat funkci vzdálenosti. Existuje řada metrik jak vzdálenost měřit. Mezi takové metriky patří např. Minowského metrika také uváděná jako L_k norma. Její podmnožiny tvoří např. Manhattanská či Eukleidovská vzdálenost [12].

Práce využívá pro měření vzdálenosti eukleidovský metriku na vektor o dvou příznacích. Prvním příznakem je plocha částice a druhým tvarová podobnost.

Tvarovou podobnost částice porovnáváme na základě Hu-momentů[18]. Jedná se o momenty pomocí nichž dokážeme tvar identifikovat. Výsledkem je deskripitor invariantní vůči translaci, rotaci z změně měřítka. Moment tvaru v binárním obrázku o rozměrech M x N je popsán jako:

$$u_{pq} = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} i^p j^q f(i, j) \quad (3.1)$$

Kde $f(x, y)$ je intenzita pixelu (v binárním obrázku 1 nebo 0) na souřadnicích (x, y) a $p + q$ udává stupeň momentu.

Protože výpočet momentu je funkce vzdálenosti mezi pixely tvaru a středem všech měření vztaženými relativně k těžišti tvaru (x', y') za účelem odstranění stupně volnosti v posuvu. Souřadnice těžiště pak lze vypočítat za použití rovnice 3.1 jako:

$$i' = \frac{u_{10}}{u_{00}} \quad j' = \frac{u_{01}}{u_{00}} \quad (3.2)$$

Odtud pak relativní momenty:

$$u_{pq} = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (i' - i)^p (j' - j)^q f(i, j) \quad (3.3)$$

Jednotlivé momenty nejsou samy o sobě dostatečně popisné aby dokázaly reprezentovat libovolné tvary, nebo udržovaly invariantní charakteristiky tvaru. Na jejich základě je odvozeno sedm invariantních momentových funkcí jejichž forma je již v hodná pro reprezentaci tvaru. Tyto momenty jsou:

$$m_1 = (u_{20} + u_{02}) \quad (3.4)$$

$$m_2 = (u_{20} - u_{02})^2 + 4u_{11}^2 \quad (3.5)$$

$$m_3 = (u_{30} - 3u_{12})^2 + (3u_{21} - u_{30})^2 \quad (3.6)$$

$$m_4 = (u_{30} + u_{12})^2 + (u_{21} + u_{03})^2 \quad (3.7)$$

$$\begin{aligned} m_5 = & (u_{30} - 3u_{12})(u_{30} + u_{12})((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2) \\ & + (3u_{21} - u_{03})(u_{21} + u_{03})(3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2) \end{aligned} \quad (3.8)$$

$$\begin{aligned} m_6 = & (u_{20} - u_{02}) - (u_{30} + u_{12})^2 - (u_{21} + u_{03})^2 \\ & + 4u_{11}(u_{30} + 3u_{12})(u_{21} + u_{03}) \end{aligned} \quad (3.9)$$

$$\begin{aligned} m_7 = & (3u_{21} - u_{03})(u_{30} + u_{12})((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2) \\ & - (u_{30} - 3u_{12})(u_{21} + u_{03})(3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2) \end{aligned} \quad (3.10)$$

Podobnost jednoho tvaru s druhým pak vypočítáme jako:

$$s(A, B) = \sum_{i=1 \dots 7} |m_i^A - m_i^B| \quad (3.11)$$

Kde $s(A, B)$ je podobnost tvarů A a B, m_i je jeden ze sedmi Hu-momentů z rovnic 3.4-3.10. Výsledkem je hodnota rostoucí od 0 a platí, že čím větší menší podobnost tím větší hodnota.

Jeden z hlavních nedostatků K-NN při výpočtu vzdálenosti od trénovacích dat je příklad kdy jednotlivé příznaky mají rozdílné škály nebo pokud se jedná o směs numerických a kategorických hodnot. Toto je příklad mnou zvolených příznaků. Velikost plochy částic se pohybuje mezi desítkami až stovkami pixelů.

Zatímco podobnost se pohybuje v přibližném rozpětí mezi 0 a 2. Hodnoty je proto třeba normalizovat. Jinak by klasifikátor klasifikoval částice pouze podle velikosti [14] [5].

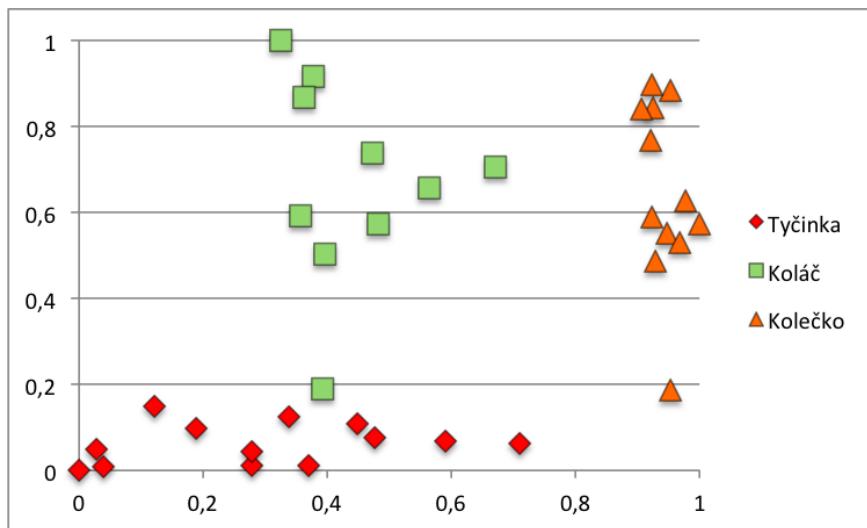
Každý příznak $x_{ij}(i = 1 \dots k, j = 1 \dots n)$, kde k je velikost vektoru příznaků a n velikost trénovací množiny, je pak normalizován jako:

$$x'_{ij} = \frac{x_{ij} - min_i}{max_i - min_i} \quad (3.12)$$

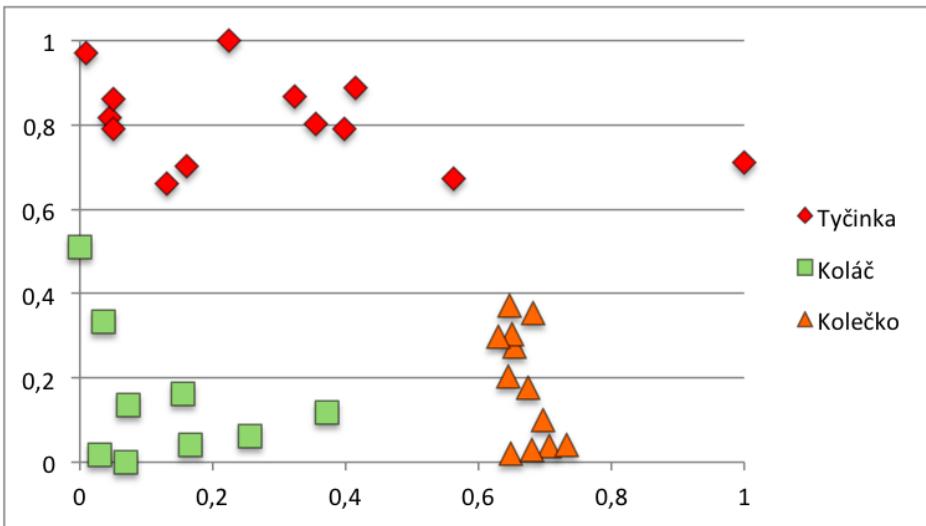
Kde $min_i = \min_{i=1 \dots k} (x_{ij})$ a $max_i = \max_{j=1 \dots k} (x_{ij})$.

Normalizace se provádí vždy při přidání nové částice mezi trénovací data. V případě, že příznaky nově přidané částice posouvají min_i nebo max_i jsou přeškálovány i ostatní hodnoty podle nového rozpětí.

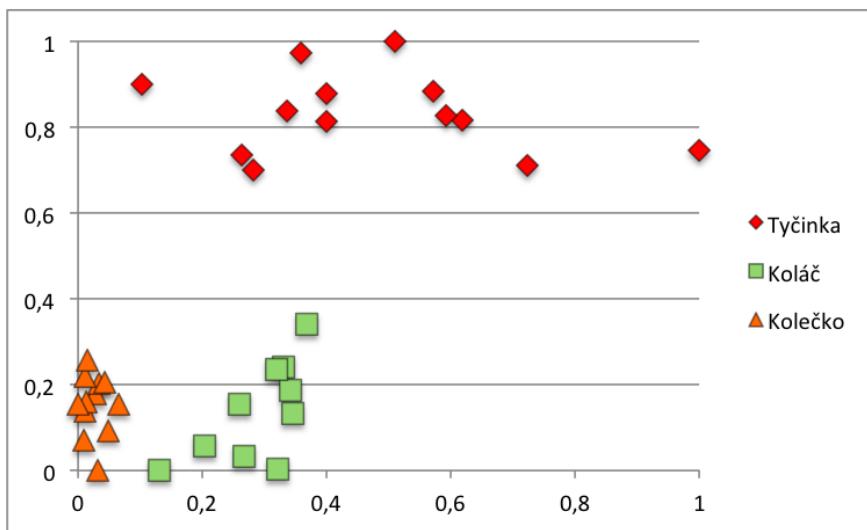
Originální prostor, ve kterém modely jednotlivých částic existují je vícedimenzionální. Z tohoto důvodu je originální prostor špatně zobrazitelný. Můžeme ale zobrazit transformovaný prostor příznaků pro jednotlivé částice viz. grafy na obr. 3.1, 3.2, 3.3.



Obrázek 3.1: Prostor příznaků pro částici, která by byla klasifikována jako typ tyčinka. Osa x: rozdíl ve velikostech částic. Osa y: tvarová podobnost.



Obrázek 3.2: Prostor příznaků pro částici, která by byla klasifikována jako typ koláč. Osa x: rozdíl ve velikostech částic. Osa y: tvarová podobnost.



Obrázek 3.3: Prostor příznaků pro částici, která by byla klasifikována jako typ kolečko. Osa x: rozdíl ve velikostech částic. Osa y: tvarová podobnost.

Souřadnice x_i v transformovaném prostoru vypočítáme jako:

$$x_i = |S - S_i| \quad (3.13)$$

Kde $i = 1 \dots N$ (N udává velikost trénovací množiny), S plochu testované částice, S_i plochu trénovací částice i .

Souřadnici y_i pak udává tvarovou podobnost mezi testovanou částicí a porovnávanou trénovací částicí viz rovnice 3.11.

Z grafů je pozorovatelné, že jednotlivé typy částic jsou od sebe výrazně odděleny. Na tomto základě byla vybrána metoda K-NN.

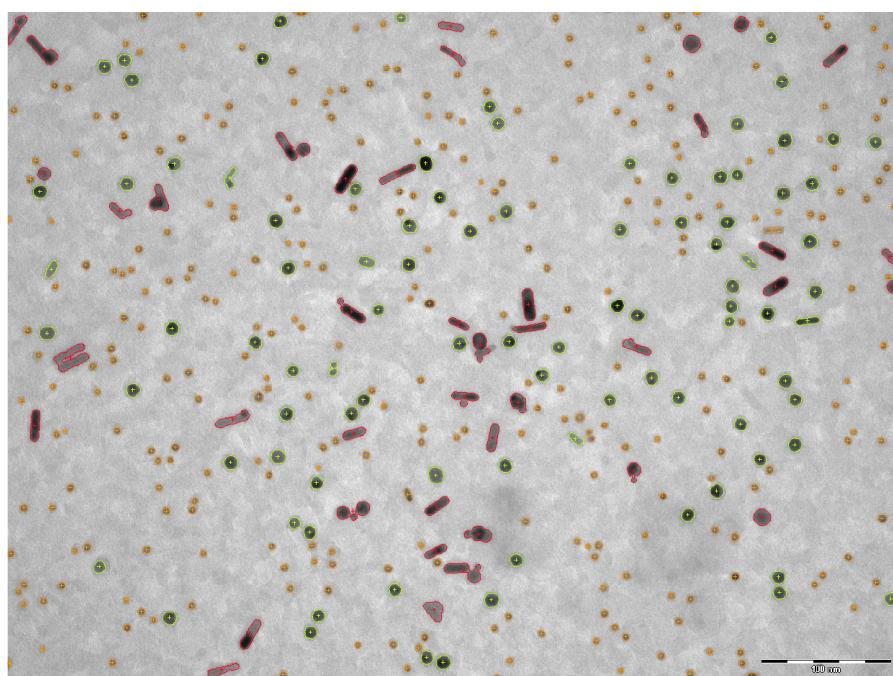
Pro výpočet podobnosti tvarů byla využita funkce *matchShapes* z OpenCV[9].

3.4 Výsledky klasifikace

3.4.1 Vstupní data

Trénování a vyhodnocení výsledků probíhalo na datech snímcích pocházejících z UMG. K dispozici je 7 mikroskopických snímků obsahujících variující tvary nanočastic. Výstupy klasifikačního algoritmu jsou prezentovány pomocí barevného značení, kde oražová barva značí částici typu kolečko, červená částici typu tyčinka a žlutozelená typ koláč.

Výstup automatické klasifikace je porovnáván vůči výstupu jak by částice klasifikoval laborant.



Obrázek 3.4: Množina trénovacích dat.

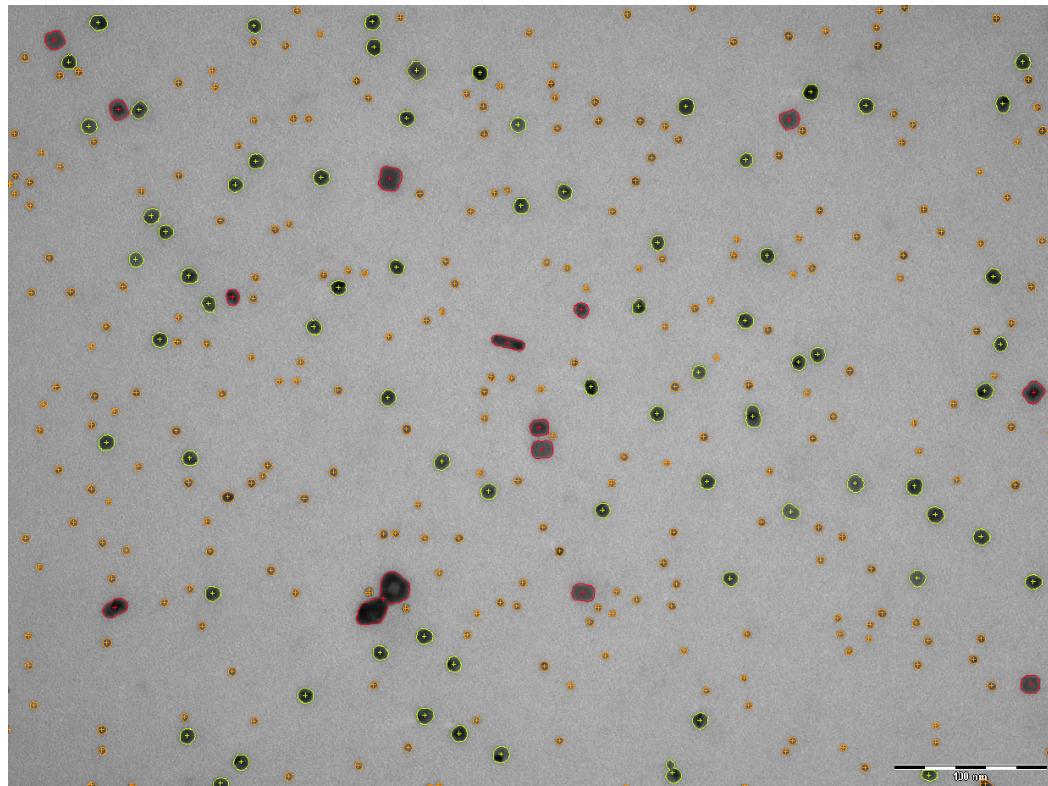
Vstupní trénovací množina je zobrazena na obr. 3.4.

Nastavení parametrů:

- Velikost okna: 51
- C: 21
- K: 5

3.4.2 Vzorek 1

V této části se nachází zjištěné hodnoty výsledků klasifikace pro vzorek 1.



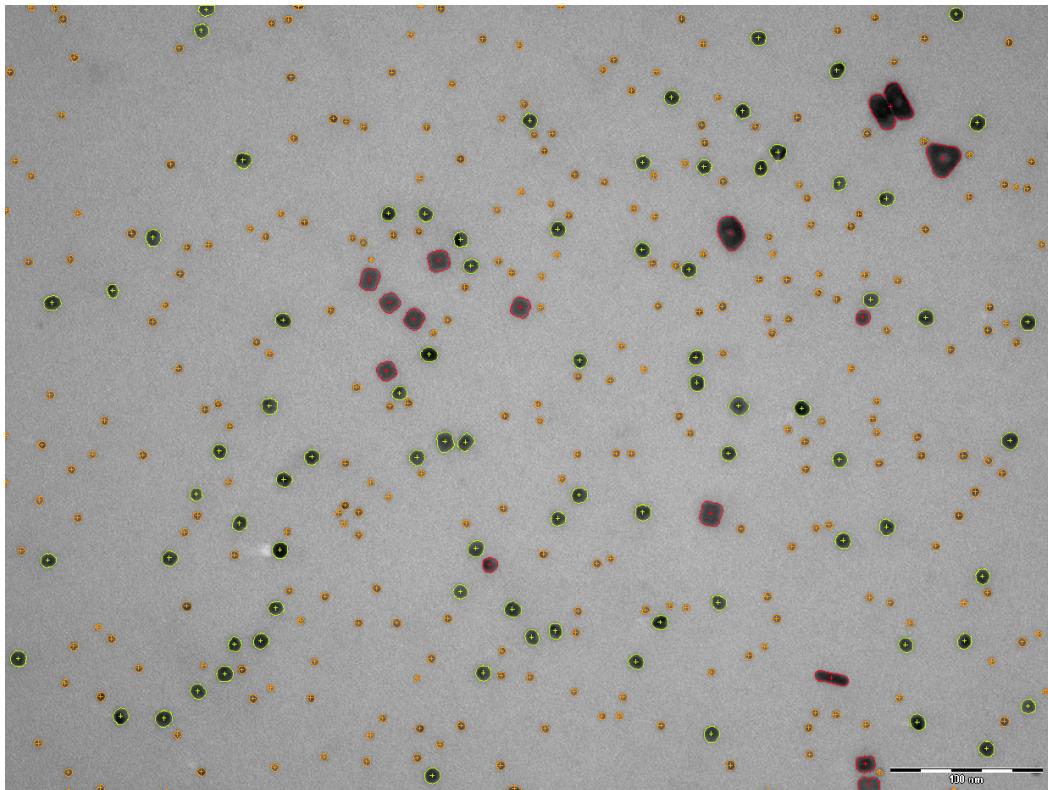
Obrázek 3.5: Výsledek klasifikace částic pro vzorek 1.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	2	0	2	100,000
koláč	267	0	267	100,000
kolečko	72	11	83	86,747
Celkem	341	11	352	96,875

Tabulka 3.1: Vyhodnocení klasifikace částic pro vzorek 1.

3.4.3 Vzorek 2

V této části se nachází zjištěné hodnoty výsledků klasifikace pro vzorek 2.



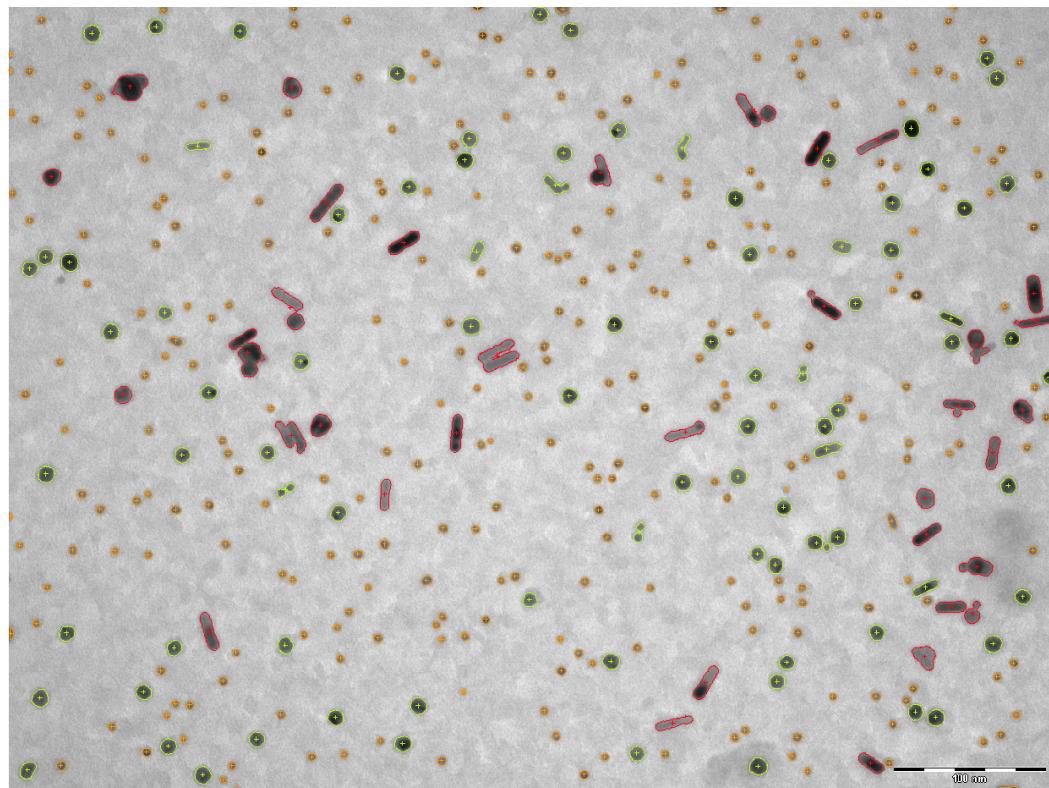
Obrázek 3.6: Výsledek klasifikace částic pro vzorek 2.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	2	0	2	100,000
koláč	267	0	267	100,000
kolčko	72	11	83	86,747
Celkem	341	11	352	96,875

Tabulka 3.2: Vyhodnocení klasifikace částic pro vzorek 2.

3.4.4 Vzorek 3

V této části se nachází zjištěné hodnoty výsledků klasifikace pro vzorek 3.



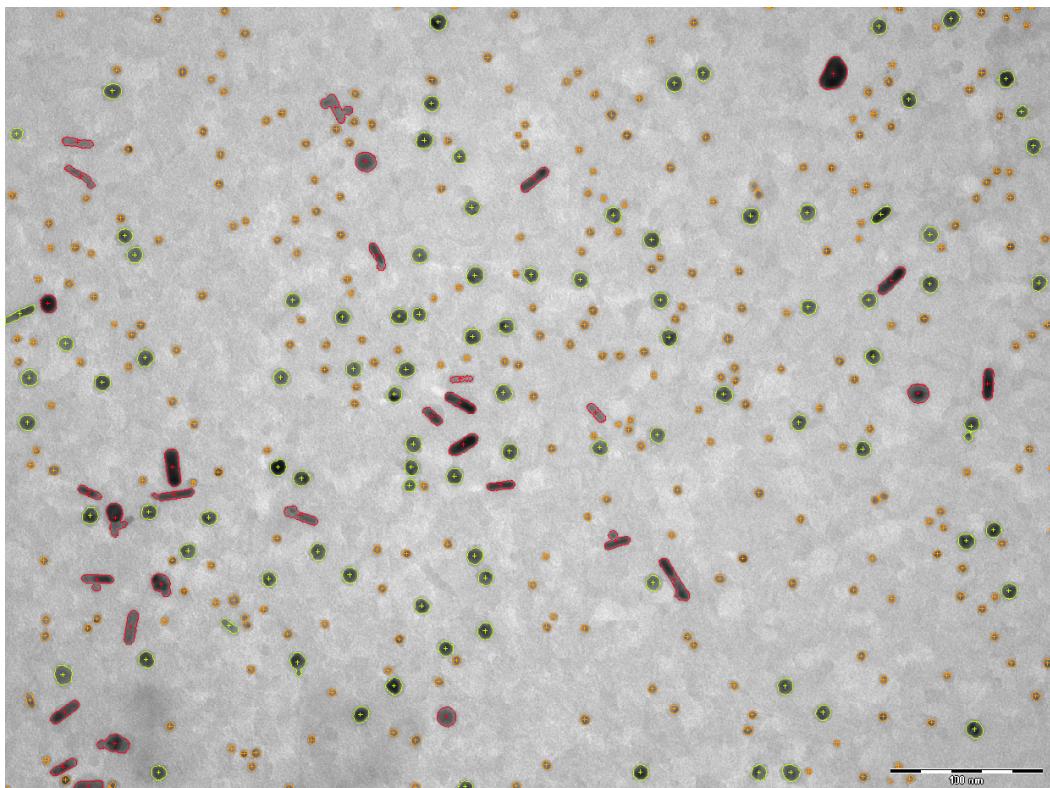
Obrázek 3.7: Výsledek klasifikace částic pro vzorek 3.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	20	2	22	90,909
koláč	296	0	296	100,000
kolečko	86	10	96	89,583
Celkem	402	12	414	97,101

Tabulka 3.3: Vyhodnocení klasifikace částic pro vzorek 3.

3.4.5 Vzorek 4

V této části se nachází zjištěné hodnoty výsledků klasifikace pro vzorek 4.



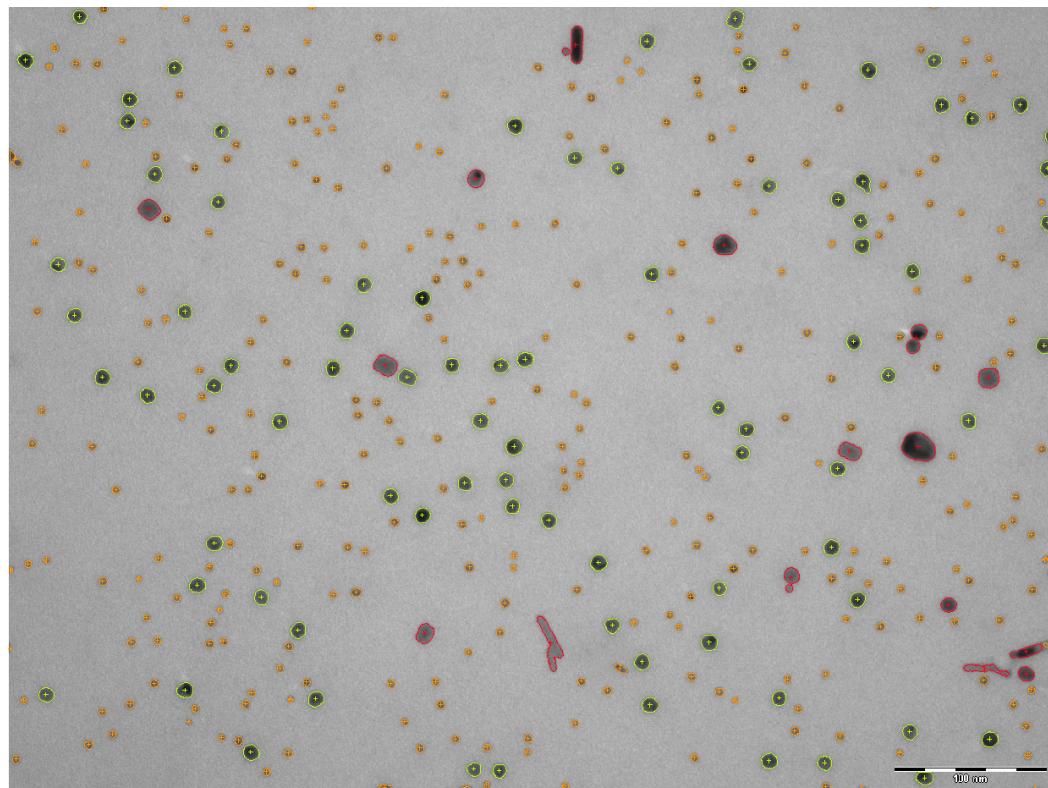
Obrázek 3.8: Vyhodnocení klasifikace částic pro vzorek 4.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	17	3	20	85,000
koláč	289	2	291	99,313
kolečko	93	9	102	91,176
Celkem	399	14	413	96,610

Tabulka 3.4: Vyhodnocení klasifikace částic pro vzorek 4.

3.4.6 Vzorek 5

V této části se nachází zjištěné hodnoty výsledků klasifikace pro vzorek 5.



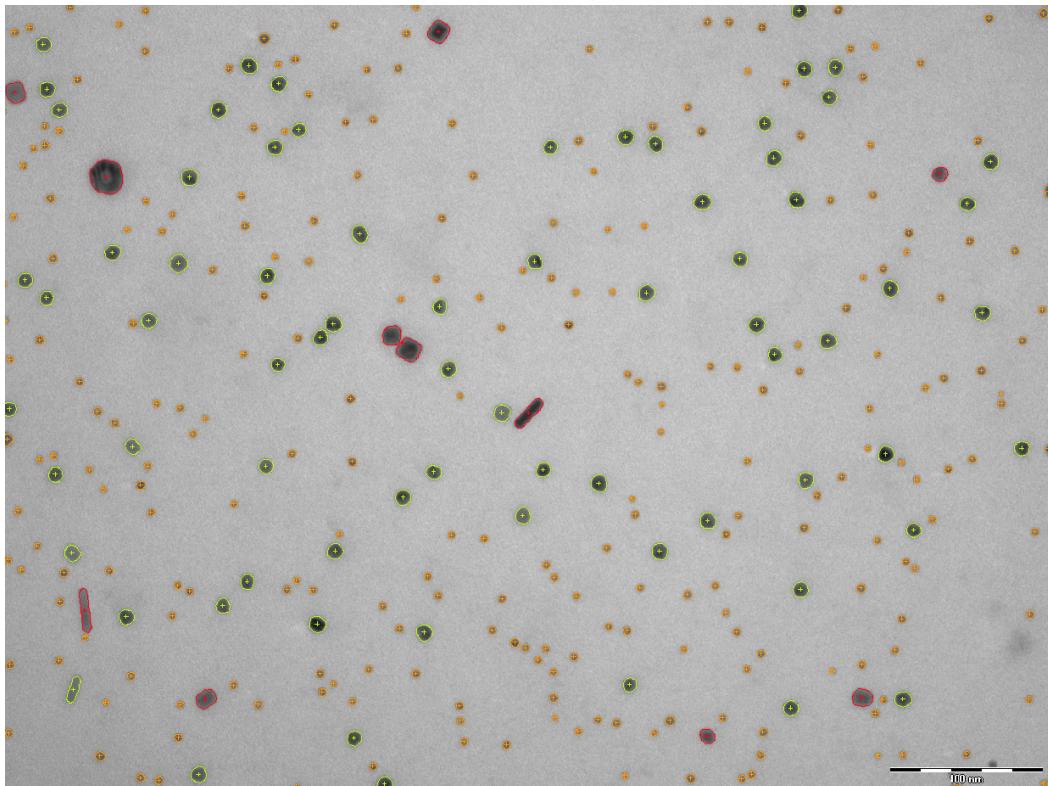
Obrázek 3.9: Výsledek klasifikace částic pro vzorek 5.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	4	0	4	100,000
koláč	307	0	307	100,000
kolečko	85	9	94	90,426
Celkem	396	9	405	97,778

Tabulka 3.5: Vyhodnocení klasifikace částic pro vzorek 5.

3.4.7 Vzorek 6

V této části se nachází zjištěné hodnoty výsledků klasifikace pro vzorek 6.



Obrázek 3.10: Výsledek klasifikace částic pro vzorek 6.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	2	1	3	66,667
koláč	254	0	254	100,000
kolečko	74	4	78	94,872
Celkem	330	5	335	98,507

Tabulka 3.6: Vyhodnocení klasifikace částice pro vzorek 6.

3.4.8 Celkový přehled

V této části se nachází shrnutí výsledků pro všechny klasifikované vzorky.

Třída	Dobře	Špatně	Celkem	Přesnost
tyčinka	73	12	85	85,882
koláč	1983	2	1985	99,899
kolečko	570	64	634	89,905
Celkem	2626	100	2718	96,615

Tabulka 3.7: Vyhodnocení klasifikace pro všechny vzorky.

Kapitola 4

Návrh a realizace aplikace

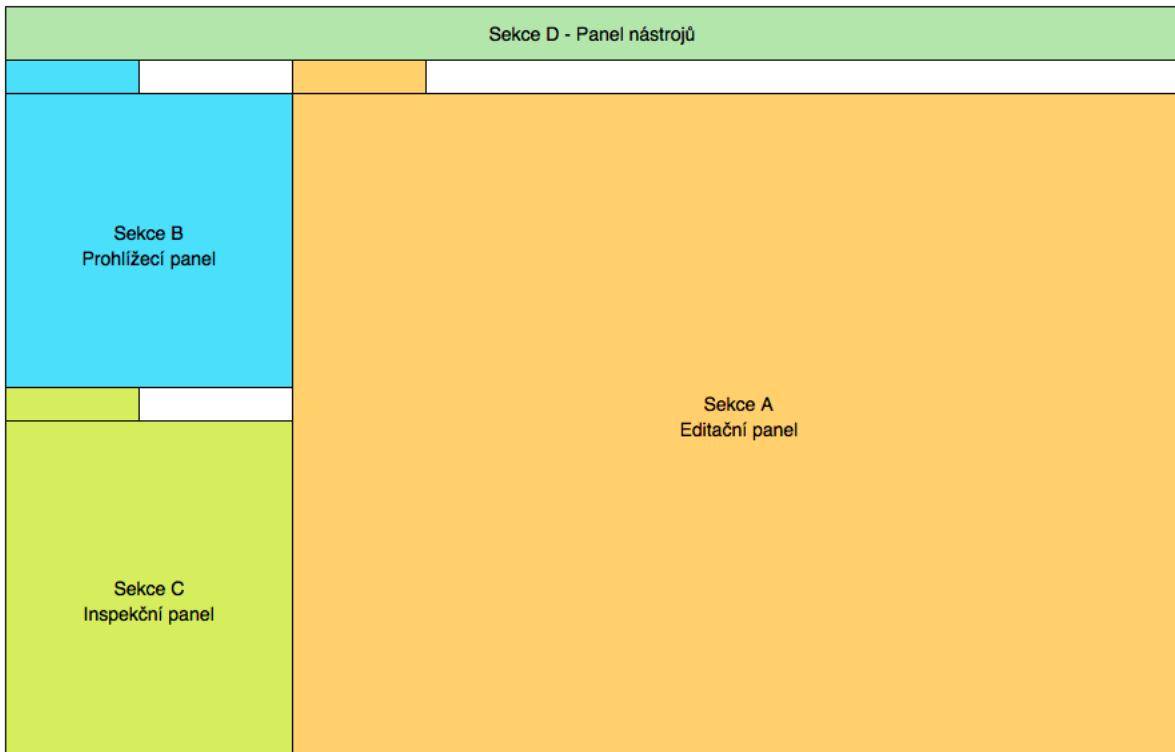
Tato kapitola popisuje návrh uživatelského rozhraní aplikace framework NetBeans [3], na kterém je aplikace postavena postavena. Nakonec jsou uvedeny knihovny a zdroje třetích stran, které aplikace využívá.

4.1 Způsob práce s daty

Zpracování mikroskopických snímků v laboratoři probíhá typicky po dávkách. Za tímto účelem bylo navrženo rozhraní, které umožňuje seskupovat snímky do skupin. Tyto skupiny nazýváme v aplikaci *Projekty*. Projekt je stromová struktura složek obsahující soubory se snímkami a uloženými daty. Více o projektech je uvedeno v sekci o uchování dat 4.3.

Typický způsob práce s aplikací je předpokládán následujícím způsobem:

1. Pořízení skupiny snímků.
2. Vytvoření projektu a načtení skupiny obrázků.
3. Detekce částic ve vybraném obrázku.
4. Možná úprava parametrů detektoru při špatně provedené detekci a opětovná detekce.
5. Ruční označení vybraného počtu částic.
6. Přiřazení označených částic klasifikátoru jako trénovací množinu.
7. Klasifikace dosud neoznačených částic natrénovaným klasifikátorem.
8. Oprava nebo trénovací množiny dat.
9. Aplikace detekce a klasifikace na další snímky
10. Analýza snímků s částicemi.



Obrázek 4.1: Přehled rozložení aplikace.

4.2 Uspořádání aplikace

Na základě těchto požadavků bylo navrženo rozložení aplikace viz 4.1.

Hlavní části okna aplikace se skládají z následujících částí:

Sekce A - Hlavní editační panel

Hlavní editační panel je standardně umístěn uprostřed aplikačního okna a zastává jeho největší část. Panel je určen k zobrazování hlavního obsahu aplikace jako jsou mikroskopické obrázky a analýzy. Jednotlivé položky je možné zobrazit v záložkách.

Sekce B - Prohlížecí panel

Prohlížecí panel je standardně umístěn v horní levé části aplikačního okna. Komponenty zobrazují jednotlivé projekty a klasifikátory.

Sekce C - Inspekční panel

Inspekční panel je standardně umístěn v dolní levé části hlavního okna. Slouží k zobrazování doplňkových informací o mikroskopickém obrázku nebo o datech v něm umístěných.

Sekce D - Panel nástrojů

Panel nástrojů je umístěn v horní části okna aplikace. Obsahuje kontrolní prvky pro manipulaci s daty v aplikaci.

4.2.1 Editační panel

V editačním panelu je zobrazen buď *editor snímků* nebo *panel analýzy*.

4.2.1.1 Editor snímků

Editor snímků je hlavní a nejdůležitější součást aplikace. Slouží primárně k práci obrazovými daty jako je detekce a klasifikace. Hlavní funkcí panelu je zobrazování obrazových dat a nanočástic.

Ten umožňuje práci ve třech režimech.

- výběr
- značení
- tvoření

První režim **výběr**, je určený k interaktivnímu výběru detekovaných částic. To umožňuje uživateli jak interaktivně zkoumat data a případně opravit výsledky detekce.

Druhý režim **značení**, je určený k přiřazování částic do jednotlivých tříd. V tomto režimu uživatel bud' poskytuje trénovací data pro klasifikátor, aplikuje klasifikátor na aktuálně zobrazené nanočástice nebo může upravovat výsledek klasifikace.

Třetí režim **tvoření** umožňuje manuální přidávání částic. Tento režim umožňuje umístit novou částici na libovolné místo v obraze.

Nedílnou součástí editace částic je jejich vizualizace. Vizualizace částic realizována přímo pomocí Java 2D Graphics. Jednotlivé částice jsou zobrazeny na obrazovém podkladě. Zobrazení lze přiblížovat, oddalovat a volit různé režimy zobrazení částic.

Částice je možné zobrazovat s/bez obrysu a s/bez zobrazeného středu.

4.2.1.2 Panel analýzy

Panel analýzy umožňuje analyzovat data pomocí hierarchického shlukování a výsledek vizualizovat pomocí dendrogramu.

Hierarchické shlukování je sekvence vnořených rozkladů, která na jedné straně začíná triviálním rozkladem, kdy každý objekt dané množiny objektů tvoří jednoprvkový shluk, a na druhé straně končí triviálním rozkladem s jedním shlukem obsahujícím všechny objekty. Podle směru postupu při shlukování dělíme metody hierarchického shlukování na aglomerační a divizivní [19].

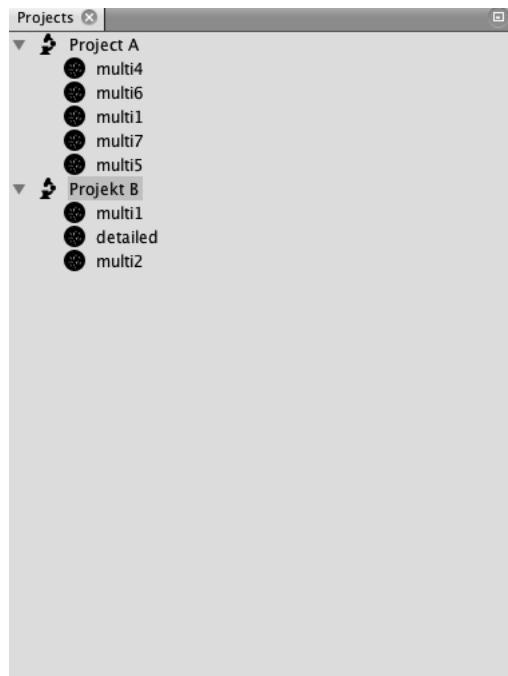
Dendrogram je binární strom znázorňující hierarchické shlukování. V aplikaci je dendrogram vizualizován tak, že každý uzel tohoto stromu představuje shluk. Vertikální řezy dendrogramem jsou rozklady ze shlukovací sekvence. Horizontální směr v dendrogramu představuje vzdálenost mezi shluky (rozklady).

Dendrogramem ve možné vést řezy danou vzdáleností, které rozdělí dendrogram na dílčí shluky. V tomto rozdelení je každý shluk barevně oddělen. Výsledek vzniklého rozdelení má pak uživatel možnost sledovat v původním snímku v NP, kde každá NP má barvu podle příslušnosti do shluku.

Implementace konstrukce dendrogramu vychází z [8]. Původní řešení je modifikováno pro účely aplikace. Byla přidána funkctionalita tvorby řezů v histogramu a upravena vizualizace histogramu.

Ukázku dendrogramu a analýzy pomocí dendrogramu je možné nalézt v příloze C jako obrázky 1 a 2.

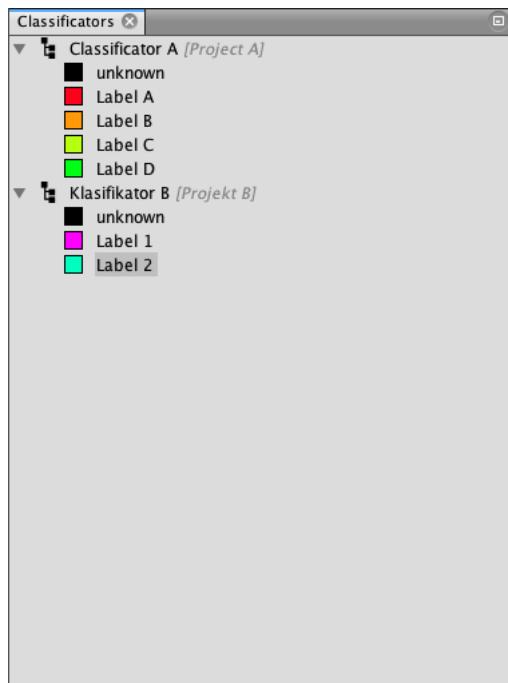
4.2.2 Přehled projektů



Obrázek 4.2: Ukázka panelu přehledem projektů a snímků.

Přehled otevřených projektů je zobrazen na panelu přehledu projektů. Panel poskytuje uživateli přehled o otevřených projektech a obrázcích v přehledné stromové struktuře. Kořen stromu tvoří projekt. Listy projektu pak znázorňují jednotlivé snímky přiřazené k projektu. Panel je také zodpovědný o přiřazování jednotlivých snímků k projektu a jejich ukládání jednotlivých obrázků.

4.2.3 Přehledu klasifikátorů



Obrázek 4.3: Ukázka panelu s přehledem klasifikátorů.

Panel klasifikátoru je standardně součástí prohlížecího panelu. Jeho funkcí je zobrazovat vytvořené klasifikátory, vytvářet klasifikátory nové a poskytovat uživateli možnost s klasifikátorem manipulovat.

Klasifikátoru lze vytvořit jednotlivé třídy pro zařazení částic. Každou třídu je možné pojmenovat a přiřadit k ní barvu. Přiřazená barva klasifikátoru ovlivňuje jakou barvou jsou částice, přiřazené do dané třídy, zobrazovány v editoru obrázků.

Klasifikátor je možné exportovat, importovat a tím přenášet mezi jednotlivými projekty.

4.2.4 Inspekční panel

Inspekční panel obsahuje v základní verzi dva panely. Panel s informacemi o snímku a inspekční panel.

Panel s **informacemi o snímku** zobrazuje uživateli obecné informace o snímku jako je jeho rozlišení, absolutní počet detekovaných částic, počet částic v jednotlivých třídách atd. Ukázka panelu je zobrazena na obr. 4.4.



Obrázek 4.4: Ukázka panelu s informacemi o snímku.

Inspekční panel zobrazuje informace o částicích které má uživatel vybrané. Takovou informací je např. počet vybraných částic, příslušnost částic do tříd, těžiště atd. Vybrané částice je pak možné exportovat do textového formátu CSV k další analýze. Ukázku panelu je možné pozorovat na obr. 4.5.

4.3 Uchování dat

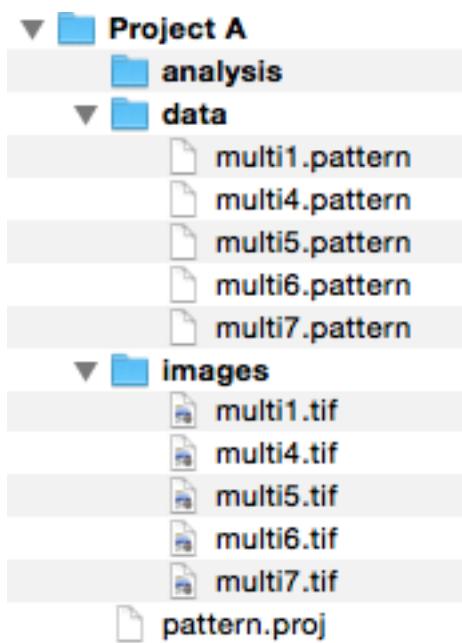
Pro uchovávání dat je v aplikaci navržena struktura složek a souborů nazvaná *projekt*. Struktura projektu se sestává ze tří složek, souboru *pattern.proj* a souborů s daty viz obr. 4.6.

Aplikace umožňuje uchovávat:

- pro každý snímek informace o detekovaných a klasifikovaných hodnotách NP.
- informace o klasifikátoru projektu

Image Info																																																																												
Inspector																																																																												
Classifiers																																																																												
Selected: 67																																																																												
<ul style="list-style-type: none"> ■ 0 ■ 9 ■ 45 ■ 13 ■ 0 																																																																												
<table border="1"> <thead> <tr> <th>id</th> <th>x</th> <th>y</th> <th>label</th> </tr> </thead> <tbody> <tr><td>122</td><td>493,48</td><td>707,46</td><td>■ 45</td></tr> <tr><td>126</td><td>613,14</td><td>687,38</td><td>■ 45</td></tr> <tr><td>128</td><td>570,23</td><td>686,45</td><td>■ 45</td></tr> <tr><td>135</td><td>432,64</td><td>666,21</td><td>■ 13</td></tr> <tr><td>136</td><td>640,25</td><td>658,45</td><td>■ 45</td></tr> <tr><td>138</td><td>337,13</td><td>654,47</td><td>■ 45</td></tr> <tr><td>141</td><td>460,82</td><td>646,66</td><td>■ 45</td></tr> <tr><td>148</td><td>363,33</td><td>636,06</td><td>■ 13</td></tr> <tr><td>149</td><td>493,84</td><td>641,95</td><td>■ 9</td></tr> <tr><td>152</td><td>413,88</td><td>622,26</td><td>■ 45</td></tr> <tr><td>158</td><td>301,88</td><td>610,73</td><td>■ 45</td></tr> <tr><td>163</td><td>537,11</td><td>602,31</td><td>■ 45</td></tr> <tr><td>167</td><td>287,54</td><td>588,04</td><td>■ 45</td></tr> <tr><td>168</td><td>496,34</td><td>585,67</td><td>■ 45</td></tr> <tr><td>170</td><td>339,57</td><td>586,96</td><td>■ 13</td></tr> <tr><td>173</td><td>619,87</td><td>576,65</td><td>■ 45</td></tr> <tr><td>176</td><td>711,53</td><td>572,60</td><td>■ 45</td></tr> </tbody> </table>					id	x	y	label	122	493,48	707,46	■ 45	126	613,14	687,38	■ 45	128	570,23	686,45	■ 45	135	432,64	666,21	■ 13	136	640,25	658,45	■ 45	138	337,13	654,47	■ 45	141	460,82	646,66	■ 45	148	363,33	636,06	■ 13	149	493,84	641,95	■ 9	152	413,88	622,26	■ 45	158	301,88	610,73	■ 45	163	537,11	602,31	■ 45	167	287,54	588,04	■ 45	168	496,34	585,67	■ 45	170	339,57	586,96	■ 13	173	619,87	576,65	■ 45	176	711,53	572,60	■ 45
id	x	y	label																																																																									
122	493,48	707,46	■ 45																																																																									
126	613,14	687,38	■ 45																																																																									
128	570,23	686,45	■ 45																																																																									
135	432,64	666,21	■ 13																																																																									
136	640,25	658,45	■ 45																																																																									
138	337,13	654,47	■ 45																																																																									
141	460,82	646,66	■ 45																																																																									
148	363,33	636,06	■ 13																																																																									
149	493,84	641,95	■ 9																																																																									
152	413,88	622,26	■ 45																																																																									
158	301,88	610,73	■ 45																																																																									
163	537,11	602,31	■ 45																																																																									
167	287,54	588,04	■ 45																																																																									
168	496,34	585,67	■ 45																																																																									
170	339,57	586,96	■ 13																																																																									
173	619,87	576,65	■ 45																																																																									
176	711,53	572,60	■ 45																																																																									

Obrázek 4.5: Ukázka panelu s informacemi o snímku.



Obrázek 4.6: Příklad souborové a složkové struktury projektu.

4.3.0.1 Ukládání dat snímků

Data pro jednotlivé snímky jsou uchovávány ve složce *data* v souborech s příponou *.pattern*. Název souboru s daty pak odpovídá názvu souboru snímku, ke kterému data patří. Pro ukládání dat je využito formátu JSON [1]. Příklad struktury ukládaného dokumentu je následující:

```
1 {  
2     "name": "multi2",  
3     "image": "multi2.tif",  
4     "particles": [  
5         {  
6             "id": 0,  
7             "cog": {  
8                 "x": 1003.0,  
9                 "y": 1029.5  
10            },  
11            "contour": [  
12                {  
13                    "x": 1001.0,  
14                    "y": 1029.0  
15                },  
16                {  
17                    "x": 1001.0,  
18                    "y": 1030.0  
19                },  
20                {  
21                    "x": 1005.0,  
22                    "y": 1030.0  
23                },  
24                {  
25                    "x": 1005.0,  
26                    "y": 1029.0  
27                }  
28            ]  
29        }
```

4.3.0.2 Ukládání dat klasifikátoru

Data pro klasifikátor se ukládají přímo do souboru *pattern.proj*, který se nachází v kořenovém adresáři projektu. Příklad struktury ukládaných dat ve formátu JSON:

```
1 {
2     "info": {
3         "id": "Classifier A",
4         "path": "path/to/pattern.proj"
5     },
6     "labels": [
7         {
8             "id": -1,
9             "name": "unknown",
10            "color": {
11                "value": -16777216,
12                "falpha": 0.0
13            }
14        }
15    ...
16 ]
17 "selectedLabel": 2,
18 "examples": [
19     {
20         "id": 47,
21         "cog": {
22             "x": 841.6866566716642,
23             "y": 918.4822588705647
24         },
25         "contour": [ ... ]
26     }
27 ]
28 }
```

4.4 Realizace aplikace

Celá aplikace je realizována v prostředí prostřednictvím programovacího jazyka **Java** a pomocí frameworku **NetBeans Platform** [3].

4.4.1 Vývojové prostředí

Netbeans Platform je programové prostředí patřící do skupiny *Rich Client Platform (RCP)*. RCP je soubor programátorských nástrojů zjednodušující vývoj aplikace. Většina RCP poskytuje následující funkce:

- manažer životního cyklu aplikace
- bundling framework
- pokročilé nástroje pro tvorbu uživatelského rozhraní
- nástroje pro internacionálizaci
- a další

Příklady dalších frameworků pro jazyk Java jsou *Eclipse RCP* [2] a *Spring Framework* [33].

4.4.1.1 Netbeans Platform

Pro vývoj aplikace byla zvolena NetBeans Platform na základě dřívějších zkušeností autora. NetBeans Platform nabízí ucelenou sadu komponent, podpůrných tříd a propracovanou modulární architekturu s konkrétními promyšlenými API, které výrazně urychlují vývoj a správu aplikací [7][37].

4.4.2 Podpůrné knihovny

V aplikace používá následující dostupné opensource knihovny:

OpenCV Knihovna s funkcemi pro zpracování obrazu, počítačové vidění a strojové učení.

Bio-formats Knihovna podporující načítání snímků v rozličných formátech.

hierarchical-clustering-java Knihovna s implementací hierarchického shlužování.

japura-gui Knihovna s dodatečnými komponentami uživatelského rozhraní.

gson Knihovna zjednodušující serializaci a de-serializaci objektů ve formátu JSON.

Kapitola 5

Diskuze

V této kapitole jsou rozebrány výsledky detekce a klasifikace.

Navržená metoda detekce částic má problémy při separování částic, které jsou příliš blízko u sebe. Takové částice se vyznačují nízkou nebo téměř žádnou magnitudou gradientu v místě přechodu mezi částicemi.

Tento problém jsem se pokoušel velice aktivně řešit, ale žádné zvolené řešení nevedlo k ideální segmentaci. Proběhl pokus o zpřesnění segmentace pomocí metody *watershed*. Problémem *watershed* algoritmu je, že potřebuje znát počáteční místa ve snímku, odkud může růst. Právě nalezení těchto míst je klíčovou součástí úspěšné segmentace pomocí *watershed* algoritmu.

Vyzkoušel jsem hledání těchto míst pomocí distanční transformace i iterativní eroze na nekonvexní objekty. Konvexita se projevila být dobrým testem pro špatně nebo dobře segmentované částice. Ty špatně segmentované jsou nekonvexní.

Takto aplikovaný *watershed* dokázal oddělit některé částice lépe. Avšak na jiných případech vykazoval chybu v podobě pře-segmentování a negativně ovlivňoval výsledky celé segmentace a následně klasifikace. Proto nakonec jako výsledné řešení bylo ponecháno "pouze primitivní" adaptivní prahování aby nedocházelo ke zbytečné zátěži programových a hardwarových komponent. S tím, že případná oprava bude ponechána na interaktivním zásahu uživatele.

Navržený klasifikátor umožňuje klasifikovat data a jeho přesnost byla uznána laboranty jako dostačující. Velikým přínosem je, že klasifikátor dokáže úspěšně klasifikovat částice typu kolečko od zbytku. Jelikož částic typu kolečko se ve snímcích nachází nejvíce a problematicky se trefují pomocí myší, jejich přeznačování by stalo značné úsilí. Takto laborant může přeznačit okolo 10-ti špatně klasifikovaných částic, které se dobře trefují myší. Tento úkon pak zabírá minimální čas.

Implementovaná aplikace byla otestována a funguje na operačních systémech Windows 7/8.1 a Mac OS X 10.9.

Aplikace nyní míří do testovacího provozu, ve kterém se prověří její skutečná funkčnost. Aplikaci chci dále ladit a rozvíjet jako opensource projekt.

Kapitola 6

Závěr

6.1 Shrnutí práce

Diplomová práce se zabývá problematikou automatizované analýzy nanočástic v mikroskopických snímcích. Je rozebrána důležitost automatizované analýzy nanočástic. Následně jsou uvedeny základní požadavky na program schopný takovou analýzu provádět. Je zjištěno že pro úspěšnou analýzu nanočástic v mikroskopických snímcích je potřeba částice nejprve správně detekovat a následně klasifikovat.

Pro účely detekce jsou nejprve algoritmy které umožňují detekci nanočástic v mikroskopických snímcích. Na základě analýzy je vybrán zvolený způsob detekce nanočástic. Zvolený algoritmus detekce využívá adaptivního prahování a morfologických operací. Algoritmus byl implementován v jazyce Java za využití knihovny OpenCV. Výstupy detekčního algoritmu jsou analyzovány a vyhodnoceny na reálných testovacích datech. Úspěšnost detekce obecně nejde pod 90%. Zvolený algoritmus má problémy při detekci častic, které se překrývají nebo jsou blízko u sebe.

Dále jsou analyzovány způsoby klasifikace nanočástic v mikroskopických snímcích. Na základě analýzy je vybrán vhodný klasifikační algoritmus. Jako klasifikační algoritmus bylo vybrán algoritmus k-nejbližších sousedů. Jsou uvedeny výsledky klasifikace za použití K-NN na reálných testovacích datech pocházejících z provozu UMG. Úspěšnost klasifikace pomocí K-NN nejde pod 80%.

Detekční a klasifikační algoritmus je součástí implementovaného programu, který umožňuje automatizovanou detekci, klasifikaci a analýzu nanočástic. Program poskytuje pro analýzu nanočástic rozličné inspekční nástroje jako je shluková analýza pomocí dendrogramu a další. Program je naimplementován v programovacím jazyce Java za využití frameworku NetBeans Platform.

Výše zmíněné postupy pak vedou k celkovému urychlení procesu analýzy snímků.

6.2 Další směřování

Aplikace nyní míří do testovacího provozu na UMG, kde bude vyhodnocen její přínos. Další možný vývoj na aplikaci se bude skládat z vylepšování uživatelského rozhraní a doplnění o další funkctionalitu, která vyplyne z testovacího provozu.

Detekční a klasifikační algoritmus bude testován v reálném provozu, kde se ověří jeho funkčnost na větším počtu snímku. V případě nefunkčnosti některé částic algoritmu na budoucích datech budou selhání vyhodnocena a algoritmus upraven.

Mikroskopické vzorky se jsou často objemové a snímají se po jednotlivých řezech. Další rozvoj aplikace a algoritmů by pak mohl směřovat na poskytnutí nástrojů pro automatizovanou analýzu nanočástic ve třírozměrném prostoru.

Literatura

- [1] Json, 1999.
- [2] Eclipse RCP. [software] <https://wiki.eclipse.org/index.php/Rich_Client_Platform>, 2014.
- [3] Netbeans platform. "<<https://netbeans.org/features/platform/index.html>>."
- [4] ADIGA, P. U. – CHAUDHURI, B. An Efficient Method Based on Water-shed and Rule-Based Merging for Segmentation of 3-D Histo-Pathological Images. *Pattern Recognition*. 2001, 34, 7, s. 1449–1458.
- [5] ARSHBROOK, A. – THACKER, N. Tutorial: Algorithms For 2-Dimensional Object Recognition, 1998. Dostupné z: <<http://www.tina-vision.net/docs/memos/1996-003.pdf>>.
- [6] BAMFORD, P. – LOVELL, B. Unsupervised Cell Nucleus Segmentation with Active Contours. *Signal Processing*. 1998, 71, 2, s. 203–213.
- [7] BCK, H. *The Definitive Guide to NetBeans Platform* 7. Berkely, CA, USA : Apress, 1st edition, 2011. ISBN 1430241012, 9781430241010.
- [8] BEHENKE, L. *Hierarchical clustering Java* [online]. Dostupné z: <<https://github.com/lbehnke/hierarchical-clustering-java>>.
- [9] BRADSKI, G. OpenCV. *Dr. Dobb's Journal of Software Tools*. 2000.
- [10] CASTELO, R. – GUIGO', R. Splice site identification by idlBNs. *Bioinformatics 20 Suppl. 1*. 2004, s. 69–76.
- [11] CHEN, L. – HO, C. Development of Nanoparticle Shape Measurement and Analysis for Process Characterization of TiO₂ Nanoparticle Synthesis. *Advanced Material Science*. 2008, 18, 8, s. 677–684.
- [12] DUDA, R. O. – HART, P. E. – STORK, D. G. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.

- [13] D'AMICO, F. – SKARMOUTSOU, E. Clustering and colocalization in transmission immunoelectron microscopy: A brief review. *Micron*. 2008, 39, 1, s. 1 – 6. ISSN 0968-4328. doi: <http://dx.doi.org/10.1016/j.micron.2007.09.005>. Dostupné z: <<http://www.sciencedirect.com/science/article/pii/S0968432807001400>>.
- [14] FLUSSER, J. On the independence of rotation moment invariants. *Pattern Recognition (PR)*. 2000, 33, 9, s. 1405–1410. Dostupné z: <<http://library.utia.cas.cz/prace/20000033.pdf>>.
- [15] GLOTOV, O. Image Processing of the Fractal Aggregates Composed of Nanoparticles. *Russian J. Physical Chemistry A, Focus on Chemistry*. 2008, 82, 13, s. 2213–2218.
- [16] GONZALEZ, R. C. – WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2006. ISBN 013168728X.
- [17] HORIKOSHI, S. – SERPONE, N. *Microwaves in nanoparticle synthesis: fundamentals and applications*. John Wiley & Sons, 2013.
- [18] HU, M.-K. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*. February 1962, 8, 2, s. 179–187. ISSN 0096-1000.
- [19] JAN, K. – ŠILHÁN, D. *Shluková analýza* [online]. Dostupné z: <<http://www.fd.cvut.cz/personal/nagyivan/Projekty/Classification/ShlukovaAnalyza.pdf>>.
- [20] K., A. M. . A. *Image segmentation with wattershed algorithm* [online]. 2014. [cit. 8. 5. 2015]. Dostupné z: <http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_watershed/py_watershed.html>.
- [21] MALPICA, N. et al. Applying Watershed Algorithms to the Segmentation of Clustered Nuclei. *Cytometry Part A*. 1997, 28, 4, s. 289–297.
- [22] MAYHEW, T. M. et al. A review of recent methods for efficiently quantifying immunogold and other nanoparticles using {TEM} sections through cells, tissues and organs. *Annals of Anatomy - Anatomischer Anzeiger*. 2009, 191, 2, s. 153 – 170. ISSN 0940-9602. doi: <http://dx.doi.org/10.1016/j.aanat.2008.11.001>. Dostupné z: <<http://www.sciencedirect.com/science/article/pii/S094096020800157X>>.
- [23] MCFARLAND, A. – DUYNE, R. V. Single Silver Nanoparticles as Real-Time Optical Sensors with Zeptomole Sensitivity. *Nano Letters*. 2003, 3, 8, s. 1057–1062.

- [24] NATH, S. – PALANIAPPAN, K. – BUNYAK, F. Cell Segmentation Using Coupled Level Sets and Graph-Vertex Coloring. *Proc. Ninth Int'l Conf. Image Computing and Computer-Assisted Intervention.* 2006, s. 101–108.
- [25] OSTROWSKI, A. et al. Overview about the localization of nanoparticles in tissue and cellular context by different imaging techniques. *Beilstein Journal of Nanotechnology.* 2015, 6, s. 263–280. ISSN 2190-4286. doi: 10.3762/bjnano.6.25.
- [26] PARK, C. et al. Segmentation, Inference and Classification of Partially Overlapping Nanoparticles. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2013, 35, 3, s. 1. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.163.
- [27] PERLMAN, Z. E. et al. Multidimensional drug profiling by automated microscopy. *Science 306.* 2004, s. 1194–1198.
- [28] R. FISHER, A. W. S. P. – WOLFART., E. *Adaptive thresholding* [online]. 2003. [cit. 8. 5. 2015]. Dostupné z: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm>>.
- [29] SAFAVIAN, S. R. – LANDGREBE, D. A survey of decision tree classifier methodology. 1990.
- [30] SAHOO, P. K. et al. A Survey of Thresholding Techniques. *Comput. Vision Graph. Image Process.* February 1988, 41, 2, s. 233–260. ISSN 0734-189X. doi: 10.1016/0734-189X(88)90022-9. Dostupné z: <[http://dx.doi.org/10.1016/0734-189X\(88\)90022-9](http://dx.doi.org/10.1016/0734-189X(88)90022-9)>.
- [31] SHI, J. – MALIK, J. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence.* 2000, 22, 8, s. 888–905.
- [32] SOMMER, C. – GERLICH, D. W. Machine learning in cell biology—teaching computers to recognize phenotypes. *Journal of cell science.* 2013, 126, 24, s. 5529–5539.
- [33] spring. Spirng Framework. [software] <<http://projects.spring.io/spring-framework/>>, 2012-2015.
- [34] SUZUKI, S. – ABE, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing.* 1985, 30, 1, s. 32–46.
- [35] TEK, F. – DEMPSTER, A. – KALE, I. Blood Cell Segmentation Using Minimum Area Watershed and Circle Radon Transformations. *Math. Morphology: 40 Years On.* 2005, 30, s. 441–454.

- [36] WÄHLBY, C. et al. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. *Journal of Microscopy*. 2004, 215, 1, s. 67–76.
- [37] WEXBRIDGE, J. – NYLAND, W. *Netbeans Palatform for Beginners*. Berkeley, CA, USA : Leanpub, 1st edition, 2014.
- [38] WORTIS, M. Chemistry and Physics of Solid Surfaces. 1988.
- [39] YANG, Z. R. – CHOU, K. C. Bio-support vector machines for computational proteomics. 2004, s. 735–741.
- [40] ZHANG, G. P. Neural networks for classification: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*. 2000, 30, 4, s. 451–462.

Příloha A

Seznam použitých zkratek

API - Application Programming Interface

NP - Nanočástice, z aglického *nanoparticles*

TEM - Transmisní elektronová mikroskopie

SEM - Rastrovací elektronový mikroskop

UMG - Ústav molekulární genetiky, Ústav Akademie Věd České Republiky

SVM - Support Vector Machines

Příloha B

Obsah přiloženého CD

V této kapitole naleznete obsah CD přiloženého k této diplomové práci.

- readme.txt - popis obsahu CD s dalšími informacemi
- pattern/ - zdrojové kódy aplikace v jazyce Java
- data/ - vzorky obrázků
- text/
 - src/ - zdrojový text v LaTeXu, včetně obrázků a šablon
 - palasjir_2015dip.pdf - text diplomové práce ve formátu PDF

Příloha C

Snímky obrazovky

Obrázek 1: Analýza data pomocí dendrogramu.

Obrázek 2: Ukázka rozčlenění do shluků po řezu v histogramu.

Obrázek 3: Ukázka aplikace při provedeném výběru částic.

Obrázek 4: Ukázka okna s aplikací.

