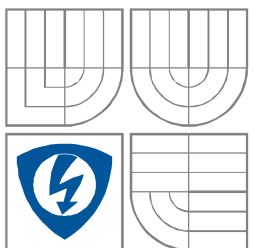


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

DETEKCE SRDEČNÍCH BUNĚK V MIKROSKOPICKÉM OBRAZU

DETECTION OF CARDIAC CELLS IN MICROSCOPIC IMAGE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Ksenia MUSIKHINA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Milan RYCHTÁŘIK

BRNO, 2009

Zde vložit originál zadání

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Ksenia Musikhina
Bytem: Dekabristov 6-2, Seltý, 427270
Narozen/a (datum a místo): 3. srpna 1987 v Seltach

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Jiří Jan, CSc, předseda rady oboru Biomedicínské a ekologické inženýrství
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
 - ☒ diplomová práce
 - ☐ bakalářská práce
 - ☐ jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Detekce srdečních buněk v mikroskopickém obrazu

Vedoucí/ školitel VŠKP: Ing. Milan Rychtárik

Ústav: Ústav biomedicínského inženýrství

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- ☒ v tištěné formě – počet exemplářů: 2
- ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☒ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 29. května 2009

.....
Nabyvatel

.....
Autor

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Detekce srdečních buněk v mikroskopickém obrazu“ jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení §11 a následujících autorského zákona č.121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení §152 trestního zákona č.140/1961 Sb.

V Brně dne 29. května 2009

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Milanu Rychtárikovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 29. května 2009

.....
podpis autora

Abstrakt

Tato diplomová práce se zabývá problematikou detekce srdečních buněk v mikroskopickém obraze. Byly vyzkoušeny různé způsoby předzpracování a segmentace obrazu za účelem určit nejvhodnější metody pro další klasifikaci. Pro samotnou klasifikaci pak byly otestovány různé metody: metoda příznaku objektu a klasifikátory založené na principu neuronových sítí. Výstupem pak byl počet živých a mrtvých buněk a jejich procentní poměr v původním mikroskopickém obraze. Míry účinnosti každého z klasifikátorů byly spočítány pomocí společného uvažování hodnot sensitivity a specificity. Pro zlepšení názornosti výsledků bylo vytvořeno grafické rozhraní v prostředí MATLAB.

Klíčová slova

Detekce objektů, zpracování obrazů, neuronové sítě, srdeční buňky.

Abstract

This work is devoted to problem of detection of cardiac cells in microscopic picture. All possible means of preprocessing and segmentation were considered with the aim to choose the most suitable method for further classification. Different methods of classification were be testing: method of objects attributes and classifier based on neural network. As a result was obtained the number of living and dead cardiac cells and percentage of them. The electivity of classification methods was calculated by sensitivity and specificity. The user's interface was created for improvement of clearness classification in MATLAB environment.

Keywords

Object detection, image processing, neural networks, cardiac cells.

MUSIKHINA, K.: *Detekce srdečních buněk v mikroskopickém obrazu: diplomová práce.*
Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
2009. 59 s., 3 přílohy. Vedoucí práce je Ing. Milan Rychtářík.

Obsah

Seznam obrázků	- 3 -
Seznam tabulek	- 4 -
1. Úvod	- 5 -
1.1 Struktura srdeční buňky – myocytu	- 5 -
2. Postup zpracování a rozpoznávání obrazu.....	- 7 -
2.1 Snímání obrazu a digitalizace.....	- 7 -
2.2 Předzpracování	- 8 -
2.2.1 Jasové transformace	- 8 -
2.2.2 Filtrace.....	- 8 -
2.3 Segmentace.....	- 9 -
2.3.1 Segmentace prahováním.....	- 9 -
2.3.2 Segmentace na základě detekce hran	- 10 -
2.4 Morfologické operace	- 12 -
2.5 Popis objektů	- 13 -
2.5.1 Měření geometrických rozměrů	- 13 -
2.5.2 Měření tvaru.....	- 14 -
2.5.3 Momenty.....	- 15 -
2.6 Rozpoznávání obrazů.....	- 17 -
2.6.1 Lineární klasifikátor	- 17 -
2.6.2 Neuronová síť jako klasifikátor	- 19 -
3. Návrh vlastního řešení.....	- 27 -
3.1 Snímání mikroskopického obrazu	- 27 -
3.2 Aplikace ekvalizace histogramu	- 27 -
3.3 Srovnávání různých typů filtru	- 29 -
3.4 Volba optimálního způsobu segmentace.....	- 29 -
3.5 Použití morfologických operace	- 33 -
3.6 Postup klasifikace	- 34 -
3.6.1 Klasifikátor založený na příznakovém popisu	- 36 -
3.6.2 Perceptronové síti s algoritmem učení backpropagation.....	- 37 -
3.6.3 RBF síť	- 41 -
3.6.4 Kohonenova síť	- 42 -
3.7 Porovnání účinnosti klasifikátorů	- 43 -

4. Realizace v prostředí MATLAB	- 46 -
5. Závěr	- 52 -
Literatura	- 53 -
Seznam příloh	- 55 -
Přílohy	- 56 -
Příloha A. Funkce pro adaptaci vah zpětným šířením chyby (backpropagation)	- 56 -
Příloha B. Funkce pro učení RBF sítě	- 57 -
Příloha C. Algoritmus Kohonenovy sítě	- 58 -

Seznam obrázků

Obr. 1 Izolovaná živá srdeční buňka	- 6 -
Obr. 2 Mrtvé srdeční buňky	- 6 -
Obr. 3 Příklad strukturních elementů	- 12 -
Obr. 4 Systém pro příznakové rozpoznávání objektu	- 18 -
Obr. 5 Klasifikace podle minima vzdálenosti	- 19 -
Obr. 6 Formální neuron	- 19 -
Obr. 7 Architektura sítě perceptronů	- 22 -
Obr. 8 Architektura RBF sítě	- 24 -
Obr. 9 Architektura Kohonenovy sítě	- 26 -
Obr. 10 Znázornění sousednosti	- 26 -
Obr. 11 Původní mikroskopický obrázek	- 27 -
Obr. 12 Histogram původního obrazu	- 28 -
Obr. 13 Ekvalizace histogramu	- 28 -
Obr. 14 Adaptivní ekvalizace histogramu	- 29 -
Obr. 15 Výsledky filtrace	- 30 -
Obr. 16 Výsledky segmentace prahováním	- 31 -
Obr. 17 Prahové hodnoty jednotlivých částí	- 31 -
Obr. 18 Různé typy detektorů hran	- 32 -
Obr. 19 Pozitivní a negativní vliv dilatace	- 33 -
Obr. 20 Pozitivní a negativní vliv binárního zavření obrazu	- 33 -
Obr. 21 Pozitivní a negativní vliv eroze	- 34 -
Obr. 22 Pozitivní a negativní vliv binárního otevření obrazu	- 34 -
Obr. 23 Odstranění nežádoucích objektů	- 35 -
Obr. 24 Testovací množina	- 36 -
Obr. 25 Rozložení hodnot excentricity objektů	- 36 -
Obr. 26 Výsledek příznakového klasifikátoru	- 37 -
Obr. 27 Zvolená trénovací množina	- 37 -
Obr. 28 Trénovací množina po zpracování	- 38 -
Obr. 29 Příklad trénovacích vzorků	- 38 -
Obr. 30 Struktura jednovrstvé sítě perceptronů	- 38 -
Obr. 31 Změna střední chyby v průběhu učení	- 39 -
Obr. 32 Výsledek klasifikace jednovrstvé sítě perceptronů	- 39 -
Obr. 33 Struktura dvouvrstvé perceptronové sítě	- 39 -
Obr. 34 Změna střední chyby v průběhu učení dvouvrstvé perceptronové sítě	- 40 -
Obr. 35 Výsledek klasifikace dvouvrstvou perceptronovou sítí	- 40 -
Obr. 36 Struktura RBF sítě	- 41 -
Obr. 37 Výsledek klasifikace RBF sítí	- 42 -
Obr. 38 Reprezentace center tříd	- 42 -
Obr. 39 Výsledek Kohonenovy sítě	- 43 -
Obr. 40 Dialog pro otevření obrázku	- 46 -
Obr. 41 Dialog pro zobrazení obrázku a příslušného histogramu	- 46 -

Obr. 42 Dialog pro porovnání operací transformace histogramu	- 47 -
Obr. 43 Dialog pro výběr typu filtru	- 47 -
Obr. 44 Dialog pro nastavení parametrů a implementaci filtru	- 48 -
Obr. 45 Dialog pro výběr metody prahování	- 48 -
Obr. 46 Dialogové okno pro nastavení prahové hodnoty a implementaci vybrané metody	- 49 -
Obr. 47. Dialog pro výběr způsobu segmentace	- 49 -
Obr. 48 Okno pro nastavení parametrů a realizaci zvoleného typu segmentace	- 50 -
Obr. 49 Dialog pro výběr typu klasifikátoru.....	- 50 -
Obr. 50 Dialog s výsledkem klasifikace	- 51 -

Seznam tabulek

Tabulka 1. Aktivační funkce neuronu	- 20 -
Tabulka 2. Úspěšnost klasifikátorů	- 44 -

1. Úvod

Technologií detekce objektu a celkové zpracování počítačem je známá jako úloha počítačového vidění. Při vyhodnocení obrazové informace hraje podstatnou roli člověk a jeho předchozí zkušenosti. Inteligence umožňuje reprezentovat nabyté znalosti či zkušenosti o okolním světě a využívat je pro řešení nových úloh. Počítačové vidění je v širším smyslu považováno za součást kybernetiky či umělé inteligence. Cílem technologie počítačového vidění je vytvářet stroje, které dokáží zpracovávat úlohy rychleji, přesněji a levněji než lidé.

Předložená práce se zabývá možnostmi předzpracování a segmentace mikroskopického obrazu za účelem detekce izolovaných kardiomyocytů v obraze. Cílem je prozkoumat různé způsoby a navrhnout klasifikátor vhodný pro rozpoznávání mrtvých a živých srdečních buněk v mikroskopickém obrazu podle jejich tvaru a struktury. Aby činnost klasifikátoru byla kvalitní, je třeba vybrat optimální způsoby předzpracování a segmentace podle kritéria spojitosti hran detekovaných objektů a jejich vzájemném nepřekrývání, které mohou být zlepšeny morfologickými operaci.

V první části jsou rozebrány existující metody realizace předzpracování a segmentace a popsány principy klasifikace založené: na příznakovém popisu objektu a neuronových sítí. Více podrobně byly popsány principy činnosti perceptronové sítě s algoritmem učení backpropagation, RBF sítě (sít' s radiální bází) a samoorganizační Kohonenové sítě. Druhá část pojednává o návrhu vlastního řešení, porovnání výsledků filtrace a segmentace, účinnosti binárních morfologických operací, srovnání úspěšností klasifikátorů. Další kapitola je věnována realizaci v prostředí MATLAB a rozebírání struktury grafické-uživatelského rozhraní. Výsledky mé práce a návrhy dalších řešení jsou shrnuty v kapitole č. 5. V příloze jsou uvedeny některé důležité zdrojové kódy.

Detekce objektů nachází uplatnění v lékařské diagnostice, analýze biologických obrazů, značných úspěchů bylo dosaženo při čtení tištěných i psaných znaků, v dálkovém průzkumu, při identifikaci lidských obličejů, identifikaci otisku prstů a v dalších oblastech [14]. Konkrétně detekce mrtvých a živých srdečních buněk v mikroskopickém snímku a určení jejich množství (případně procentního poměru) může být účinným nástrojem při hodnocení úspěšnosti izolace srdečních buněk z myokardu či může napomoci ke zkoumání životaschopnosti takto naizolovaných buněk.

1.1 Struktura srdeční buňky – myocytu

Srdeční tkáň svalová se skládá z buněk, zvaných *kardiomyocyty*, jež lze na základě strukturních znaků a funkce rozdělit na pracovní a vzrušivé.

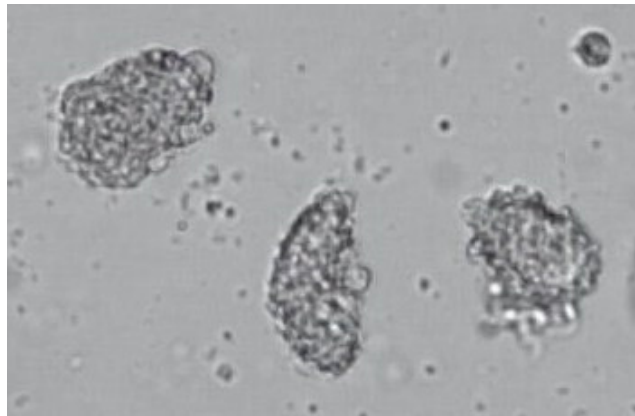
Pracovní (kontraktilní) kardiomyocyty jsou buňky válcovitého tvaru, tloušťky 10 – 15 a délky 85 – 100 μm . Prostor mezi jádrem a sarkolemou vyplňují žíhané myofibrily, probíhající od jednoho konce ke druhému, a jsou příčinou příčného pruhování a smršťování. Pracovní kardiomyocyty tvoří střední vrstvu stěny srdce (myokard) a jsou navzájem spojeny interkalárními disky. Vzniká tak prostorová síť, která umožňuje koordinovaný stah srdečního svalu. [4] Buňky nemají schopnost regenerace a jsou-li zničeny nebo poškozeny, jsou nahrazeny vazivem (jizvy na myokardu).

Vzrušivé (nonkontraktilní) kardiomyocyty jsou součástí převodního neboli excitomotorického aparátu srdce. Buňky se vyznačují schopností spontánně vytvářet a vest impulsy a tyto rozvádět. Strukturně vzrušivé kardiomyocyty charakterizuje celkově nízký počet myofibril a jejich víceméně náhodné uspořádání. [2]

V důsledku nedostatečného okysličení krve, například při ucpání koronárných tepen může dojít k buněčné smrti, následkem čeho se mění tvaru a struktury původní buňky (obr. 1 a 2), kterou jsme schopni vizuálně odlišit od buňky živé.



Obr. 1 Izolovaná živá srdeční buňka



Obr. 2 Mrtvé srdeční buňky

2. Postup zpracování a rozpoznávání obrazu

Obraz může být modelován matematicky pomocí spojitě skalární funkce f dvou nebo tří proměnných, které se říká obrazová funkce. V jednoduchém případě je statický obraz popsán obrazovou funkcí dvou souřadnic v rovině $f(x,y)$. Obrazová funkce tří proměnných se použije, když se obrazy mění v čase t nebo v případě objemových obrazů, např. tomografu. Hodnoty obrazové funkce odpovídají některé měřené fyzikální veličině, např. jas u obrazu.

Počítačové vidění je vědní obor, který se technickými prostředky snaží napodobit některé schopnosti lidského vidění. Interpretace obrazových dat je jádrem porozumění obrazu v počítačovém vidění. Snahou je napodobit proces vnímání u člověka a jemu podobný způsob rozhodování na základě informace obsažené v obrazech. [9]

Podle [14] se dá postup zpracovávání a rozpoznávání obrazu reálného světa obvykle rozložit do posloupnosti těchto základních kroků:

1. Snímání, digitalizace a uložení obrazu v počítači.
2. Předzpracování.
3. Segmentace obrazu na objekty.
4. Popis objektů.
5. Porozumění obsahu obrazu (často jen klasifikace objektů).

2.1 Snímání obrazu a digitalizace

Potřebujeme-li obraz zpracovat počítačem, musíme ho nejprve sejmout a digitalizovat. Snímáním rozumíme převedení některé optické veličiny na elektrickou. Otázce snímání je třeba při nasazení metod digitálního zpracování obrazu věnovat velkou pozornost. Snímací zařízení, uspořádání scény i příslušného obrazového čidla a vhodné osvětlení mají na další zpracování rozhodující vliv. Je důležité si uvědomit, že informace, kterou ztratíme špatným snímáním, v průběhu zpracování již nikdy nezískáme.

Příčiny ztráty informace obrazových dat v úlohách počítačového vidění lze shrnout do následujících bodů:

1. Ztráta informace při perspektivním zobrazení původně trojrozměrné scény do dvojrozměrné roviny čidla kamery. Všechny body na polopřímce dané bodem (x, y) v obrazové rovině ve směru od středu promítání se zobrazí právě do bodu (x, y) . Zpětná úloha, která se snaží odvodit trojrozměrné vlastnosti objektů z obrazu jediné kamery, má nekonečně mnoho řešení. Ve skutečnosti ji lze řešit jen tehdy, je-li možné využít, dalších dodatečných znalostí.

2. Komplikovaný a nejednoznačný vztah mezi jasnem, který měří kamera a tvarem povrchu 3D objektů ve scéně. Jas bodu závisí na mnoha vlivech

- odrazivosti povrchu pozorovaného předmětu,
- poloze a vlastnostech zdrojů světla,
- orientaci povrchu vzhledem k pozorovateli.

Úloha určení 3D vlastností pozorovaných objektů na základě radiometrických měření je také nedostatečně určená úloha.

3. Veliké množství obrazových dat.

4. Šum, který je v obraze reálné scény vždy přítomen, je důvodem, proč je při zpracování velmi často nutné použít pravděpodobnostní techniky. Často ovšem není k

dispozici tolik obrazů, aby bylo možné korektně odhadnout statistické vlastnosti obrazových signálů.

5. Vztah mezi pozorovaným detailem a zjišťovaným celkem. Algoritmy zpracování obrazu mohou obvykle analyzovat vlastnosti části obrazu prostřednictvím malého posuvného okna. Tímto způsobem se ovšem těžko zjišťují globální vlastnosti obrazu, o které většinou jde. [9]

2.2 Předzpracování

Metody předzpracování obrazu slouží k zlepšení obrazu z hlediska dalšího zpracování. Cílem předzpracování je potlačit šum vzniklý při digitalizaci a přenosu obrazu, odstranit zkreslení dané vlastnostmi snímacího zařízení nebo potlačit či zvýraznit jiné rysy obrazu důležité z hlediska dalšího zpracování.

2.2.1 Jasové transformace

Při bodové jasové transformaci dochází ke změně jasové stupnice. Změna intenzity obrazového bodu nezávisí na poloze bodu v obrazu. Transformace T výchozího jasu p na novou stupnici q je dána vztahem: [18]

$$q = T(p) \quad (1)$$

Obvykle transformace jasové stupnice jsou zesvětlení, ztmavení, zvýšení kontrastu a snížení kontrastu. Transformační vztah se nejčastěji pro zvyšování kontrastu hledá automaticky metodou **vyrovnání (ekvalizace) histogramu**.

Histogram jasu $h_f(z_i)$, $i = 0, \dots, L-1$ je vektor s počtem složek rovným počtu jasových úrovní L . Hodnota každé složky odpovídá četnosti bodů příslušného jasu v obraze, takže pomocí histogramu získáme představu o rozdělení jasových úrovní v digitálním obraze. Histogram lze chápat jako odhad hustoty pravděpodobnosti rozdělení jasu v obraze. [9]

Ve výsledném vyrovnaném histogramu jsou jednotlivé jasové úrovně zastoupeny zhruba stejně četně. Ekvalizací se zvýší kontrast pro úrovně jasu blízko maximu histogramu a sníží kontrast blízko minima histogramu. Nevýhodou metody je to, že je nerozlišující, což znamená, že může zvětšit kontrast šumu v pozadí, zatímco se sníží užitečný signál. Vyhnout tomuto problému je možné použitím **adaptivní ekvalizace histogramu**. Tato metoda spočívá v aplikaci ekvalizace histogramu na malé oblasti obrazu. [18]

2.2.2 Filtrace

Filtrace je název pro soubor transformací obrazu, které převádějí hodnoty jasu vstupního obrazu na hodnoty jasu výstupního obrazu s cílem zvýraznit nebo potlačit některé vlastnosti. Volba transformace je přitom závislá na velikosti objektů. Častým cílem filtrace je vyhlazování šumu v obraze.

Obyčejné průměrování filtruje obraz tím, že jako nový jas bodu přiřadí aritmetický průměr jasu bodů obdélníkového okolí. Potlačí se skvrny šumu menší než velikost okolí, které by mělo být menší než nejmenší významný detail v obraze. Nevýhodou obyčejného průměrování je rozmazávání hran. Je možné šum v obraze odstranit bez podstatného rozmazání objektů, je-li objekty budou poměrně velké a naopak šumu budou odpovídat malé oblasti lišící se od objektu jasně.

Kruhový filtr funguje stejně jako průměrovací filtr, ale v tomto případě je konvoluční maska ve „tvaru“ kruhu s různou vahou jednotlivých prvků masky. [8]

Často se používá filtrace **metodou mediánu**. Medián M je hodnota, pro kterou je pravděpodobnost jevu $x < M$ rovna jedné polovině. Výpočet mediánu je pro diskrétní obrazovou funkci jednoduchý. Stačí uspořádat vzestupně hodnoty jasu v lokálním okolí a medián určit jako prvek, který je uprostřed této posloupnosti. Metoda filtrace mediánem stanoví jas výsledného bodu jako medián určený z hodnot jasu bodů v lokálním okolí vstupního obrazu. Metoda redukuje stupeň rozmazání hran a dobře potlačuje impulsní šum. Tuto metodu filtrace lze použít iterativně. Hlavní nevýhodou filtrace mediánem v obdélníkovém okolí je to, že porušuje tenké čáry a ostré rohy v obraze. Tento nedostatek se dá obejít, když se použije jiný tvar okolí.

Filtrace Gaussiánem se používá pro vyhlazení (rozmazání) obrazu, odstranění detailů a šumu. Gaussián G je v 2D definován jako: [15]

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2)$$

kde σ je směrodatná odchylka. Předpokládá se, že střední hodnota rozložení je nulová. V případě diskrétního obrazu potřebujeme diskrétní aproximaci funkce Gaussiánu. Ideou vyhlazování Gaussiánem je použití 2D rozložení jako funkce pro výpočet vah koeficientů konvolučního jádra, se kterým se provede konvoluce s obrazem. Velikost konvolučního jádra je určena velikostí směrodatné odchylky σ a koeficienty jádra vzdálenější než σ jsou nulové. [15]

Filtrace **metodou rotující masky** se snaží podle homogenity jasu najít k filtrovanému bodu část jeho okolí, ke které pravděpodobně patří. Pro výpočet výsledné hodnoty jasu průměrováním je použita jen tato homogenní část. Do průměru jsou zahrnuty jen ty obrazové elementy, které nepředstavují hrany. Metoda nerozmazává hrany obrazu, má dokonce mírně ostří character. Homogenita oblasti se často měří rozptylem jasu. Velikost a tvar masek ovlivňuje rychlost konvergence. Čím jsou menší masky, tím jsou menší změny a tím více iterací je potřeba. [14]

2.3 Segmentace

Segmentace obrazu je jedním z nejdůležitějších kroků k analýze obsahu zpracovaných obrazových dat [14]. Segmentaci obrazu můžeme specifikovat jako proces, který člení obraz do nespojitých a nepřekrývajících se oblastí. Oblastí, do kterých je žádoucí rozdělit obraz, lze často odlišit na základě úrovně šedé nebo barvy jednotlivých obrazových bodů, ze kterých se tyto části skládají. [3]

2.3.1 Segmentace prahováním

Prahování je transformace vstupního obrazu f na výstupní (segmentovaný) binární obraz g podle vztahu: [14]

$$g(i, j) = \begin{cases} 1 & \text{pro } f(i, j) \geq T, \\ 0 & \text{pro } f(i, j) < T, \end{cases} \quad (3)$$

kde T je předem určená konstanta nazvaná práh a $g(i, j)=1$ pro obrazové elementy náležející po segmentaci objektům a $g(i, j)=0$ pro elementy pozadí (nebo naopak).

Správná volba prahu je pro úspěšný výsledek prahování zásadní. Hodnotu prahu lze určovat interaktivně nebo pomocí některé z metod automatického určování prahu. [14]

Globální prahování

Metoda je charakteristická tím, že do ní vstupuje jediný parametr známý jako globální prahová hodnota. Zda může být obraz správně segmentován tímto způsobem nebo ne, může být zjištěno vyšetřením jeho histogramu intenzity světla (jasu). Typ histogramu musí být bimodální, tj. intenzity jednotlivých bodů budou seskupeny okolo dvou dobře separovaných hodnot. Vhodná prahová hodnota pro oddělení těchto dvou skupin je někde mezi těmi dvěma vrcholy v histogramu. Pokud ovšem rozdělení intenzit nemá tuto charakteristiku, pak je nepravděpodobně, že globální prahování bude dávat pro zpracováváný obraz přijatelný výsledek. [3]

Adaptivní prahování

Pokud se v obraze vyskytují šum a různé stíny, musí být pro každý obraz určeno více prahových hodnot. Obecně je práh počítán pro každý pixel originálního obrazu, tato hodnota je poté použita pro porovnávání s hodnotou daného pixelu a vzniká nový obraz. Tato metoda je tedy typická tím, že mění velikost prahové hodnoty dynamicky nad obrázkem, který zpracovává. Pro hledání prahové hodnoty se používá několik postupů.

1. *Metoda Chow a Kaneka* funguje na principu rozdělení obrazu do pole překrývajících se pod obrazů a následném nalezení vhodné prahové hodnoty pro každý tento podobraz vyšetřením jeho histogramu. Nevýhoda tohoto přístupu je výpočetní náročnost a proto není vhodná pro aplikace pracující v reálném čase.

2. *Algoritmus Otsu* (metoda optimálního prahování) vychází z relativního histogramu a je založen na statistických výpočtech. Předpokládá existenci dvou tříd C_0 a C_1 (body objektů a body pozadí) rozdělené prahem v hodnotě k , což znamená, že body patřící do třídy C_0 mají hodnotu jasu v intervalu $\langle 0, k-1 \rangle$ a body ze třídy C_1 v intervalu $\langle k, J-1 \rangle$. Hodnota J odpovídá počtu jasových úrovní obrazu. Jedná se tedy o nalezení prahu v takovém místě histogramu, kde vzdálenost středních hodnot obou dvou tříd je maximální a dochází tak k optimální separaci objektů a pozadí.

3. *Metoda lokálního prahování* hledá prahovou hodnotu pro každý pixel statistickým vyšetřením intenzit jasu v lokálním okolí bodu. Hodnotu prahu T možno chápat třemi způsoby. Prvním přístupem je určení střední hodnoty z intenzit jasu v lokálním okolí bodu ($T = \text{střední hodnota}$). Dalším je určení hodnoty zvané medián, což je prostřední hodnota z intenzit jasu v okolí ($T = \text{medián}$). Posledním a nejjednodušším způsobem je vypočtení T jako $T = (\max - \min)/2$, kde $\max$ je rovno maximální hodnotě jasu v okolí bodu a $\min$ je rovno minimální hodnotě jasu v daném okolí.

Velikost okolí musí být tak rozlehlá, aby pokryla dostatečně množství pixelů, které jsou součástí objektů a zároveň pixelů pozadí. V opačném případě je zvolena špatná prahová hodnota. Na druhou stranu, pokud vybereme příliš velké okolí bodu, může to vést k porušení předpokladu, že osvětlení v lokálním okolí bodu se příliš nemění. Tato metoda je méně výpočetní náročná než algoritmus Otsu a produkuje dobré výsledky pro mnohé aplikace. [3]

2.3.2 Segmentace na základě detekce hran

Velkou skupinu segmentačních metod tvoří metody založené na informaci o hranách obrazu. Hrana je vektorová veličina a je určena velikostí a směrem. Tyto veličiny vycházejí z gradientu (označujeme ∇) obrazové funkce v obrazovém elementu. Hrana indikuje body obrazu, ve kterých dochází ke změnám obrazové funkce. Segmentace na základě detekce hran vycházejí ze skutečnosti, že hranice oblastí obrazu sestávají z hran, které jsou v obraze nalezeny aplikací některého z hranových operátorů.

Hranové operátory lze rozdělit do dvou skupin:

1. *Operátory aproximace derivace obrazové funkce pomocí diferencí realizovaných diskretní konvolucí.*

Nejstarší a velmi jednoduchý je **Robertsův operátor**, který používá jen okolí 2×2 obrazového bodu. Jeho konvoluční masky jsou

$$h_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (4)$$

Příkladem lepší aproximace digitálního gradientu je **Sobelův operátor**, který opět aproximuje první parciální derivace. Proto je směrově závislý. Sobelův operátor lze vytvořit pro různé veliké rozměry masek. Konvoluční masky Sobelova operátoru pro dva z osmi směrů jsou ve vztahu:

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}. \quad (5)$$

Masky pro ostatní směry vzniknou jen pootočením.

Existují další gradientní operátory, jejich myšlenka je stejná jako u Sobelova operátoru. Liší jen koeficienty v konvolučních maskách. Takovými operátory jsou např. Robinsonův, Kirchův a Prewittové.

2. *Operátory hledání hran v místech, kde druhá derivace obrazové funkce prochází nulou.*

Hranový detektor tohoto typu využívá skutečnosti, že první derivace obrazové funkce nabývá svého maxima v místě hrany. Druhá derivace protíná v místě hrany nulovou hodnotu. Určení druhé derivace lze úspěšně provést pomocí vychlazovacího filtru s Gaussovým (normálním) rozložením jak je patrné ze vzorce: [14]

$$G(i, j) = e^{-\frac{i^2 + j^2}{2\sigma^2}}, \quad (6)$$

kde i, j jsou souřadnice v obrazu a σ je střední kvadratická odchylka, která je přímo úměrná okolí bodu, na kterém se filtrace uplatní. Body vzdálenější od středu rozdělení mají menší vliv. Druhou derivaci nám poskytne Laplaceův operátor ∇^2 . Díky linearitě použitých operací lze zaměnit pořadí vychlazování Gaussovým filtrem G pomocí konvoluce (označena $*$) a druhé derivace. Pro průchody nulou potom platí [16]:

$$\nabla^2(G * g) = (\nabla^2 G) * g = 0. \quad (7)$$

Derivace Gaussova filtru $\nabla^2 G$ se dá spočítat předem analyticky, a tak lze snížit náročnost výpočtů. **Laplaceově gradientní operátor** ∇^2 , který aproximuje druhou derivaci, je necitlivý vůči otočení a udává je velikost hrany a ne její směr. V digitálním obraze je také aproximován diskretní konvolucí [6]. Dvě používaná konvoluční jádra (pro 4-sousedství a 8-sousedství) rozměru 3×3 jsou

$$h_{4-soused} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad h_{8-soused} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (8)$$

Tradiční operátory $\nabla^2 G$ a následné hledání průchodů nulou má také nevýhody [9]:

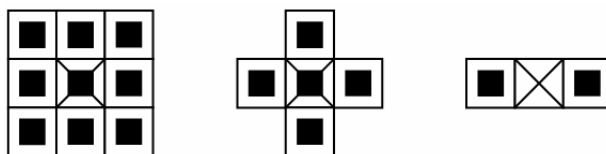
1. Příliš vyhlazují ostré tvary, například ostré rohy se ztrácejí.
2. Snaží se spojovat hrany do uzavřených křivek.

Princip různých rozlišení a hledání nejlepšího z nich realizuje **Cannyho hranový detektor**. Detektor je optimální pro skokové hrany vzhledem ke třem kritériím. Detekční kritérium zajišťuje, aby nebyly opomenuty významné hrany. Lokalizační kritérium se snaží, aby rozdíl mezi skutečnou a nalezenou pozicí hrany byl minimální. Třetí kritérium zajišťuje, aby detektor nereagoval na jednu hranu vícenásobně. Výsledkem Cannyho hranového detektoru jsou velikosti i směry hran [14].

2.4 Morfologické operace

Morfologické operace se používají při dodatečném zpracování binárních snímků, odstranění šumových objektů a defektů v objektech, úpravě pozadí, ztenčování a propojení hranic.

Morfologická transformace Ψ je relace mezi obrazem X s bodovou množinou (strukturním elementem B (viz obr. 3) vztaženým k „lokálnímu“ počátku O - reprezentativnímu bodu). Morfologické transformace $\Psi(X)$ je prováděna systematickým posouváním strukturního elementu B po obraze, výsledek relace (0 nebo 1) se zapíše do výstupního obrazu v reprezentativním pixelu [5]. Morfologické operace mění tvar a velikost geometrických objektů v obraze. Cílem transformací je kvantitativní popis objektů. Morfologické transformace neposkytují funkcionální předpis pro popis jedné charakteristiky původního objektu, ale nový transformovaný objekt. [9]



Obr. 3 Příklad strukturních elementů

Binární dilatace podle [7] je formulována otázkou: Kryje-li maska B o počátku v O některý pixel objektu v X ? Jestli je aspoň pod jednou jedničkou masky prvek objektu obrazu (tj. také jednička), pak vložíme jedničku do výstupu na polohu počátku B ve výstupním obraze. Účinky dilatace:

- dilatuje (přidává „pruhy“) podle hranic objektu;
- zvětšuje objekty;
- odstraňuje malé otvory a úzké zálivy v objektech;
- může propojit objekty (například část hran);
- změny nejsou reversibilní (nelinearita).

Při provádění *binární erose* podle [7] odpovídáme na otázku: Souhlasí-li maska B o počátku v O s objektem v X ? Jestli pod všemi jedničkami masky prvky objektu obrazu (tj. také jedničky), pak vložíme jedničku do výstupu na polohu počátku B ve výstupním obraze. Působení erose:

- eroduje (odstraňuje „pruhy“) podle hranic objektu;

- zmenšuje objekty;
- odstraňuje výčnělky a izolované malé objekty;
- může rozdělit propojené objekty (nežádoucí můstky apod.);
- změny nejsou reverzibilní (nelinearita).

Binárním otevřením podle [7] je eroze následovaná dilatací. Souhlasí-li maska B o počátku v O s objektem v X , tj. jestli pod všemi jedničkami masky prvky objektu obrazu, pak zapíšeme jedničky všude pod aktivní maskou, jinde zůstanou nuly z počátečního stavu výstupu. Účinkem je:

- filtrace objektu „zevnitř“;
- odstranění světlých stop v tmavé oblasti;
- vyhlazení hranic.

Binárním uzavřením podle [7] je dilatace následovaná erozí. Souhlasí-li maska B o počátku v O s pozadím v X , tj. jestli pod všemi jedničkami masky prvky pozadí obrazu (tj. nuly), pak zapíšeme nuly všude pod aktivní maskou, jinde zůstanou jedničky z počátečního stavu výstupu. Účinkem je:

- filtrace objektu zvenčí (z pozadí);
- odstranění tmavé stopy ve světlé oblasti;
- vyhlazení hranic.

2.5 Popis objektů

2.5.1 Měření geometrických rozměrů

Geometrické vlastnosti jednotlivých segmentovaných částí obrazu jsou velmi často důležité pro popis objektů, tj. pro vyjádření příznaků.

Plocha

Měření plochy znamená sečtení počtu obrazových bodů, které se nacházejí uvnitř hraniční čáry objektu. Přesněji plocha odpovídá součtu uvažovaných obrazových bodů, vynásobených plochou jednoho obrazového bodu. Plocha se používá pro měření celkové velikosti, je závislá jen na hranici objektu a nepřihlíží na změnu úrovně šedé uvnitř hrance a objektu.

Obvod a délka oblouku

V mnohých případech je výhodné pro klasifikaci použít délku hraniční čáry, tj. obvodu extrahovaného objektu (segmentované oblasti). Obvod lze stanovit jako počet obrazových bodů tvořící hraniční čáru objektu.

Přímost hranice

Jinou skalární vlastností je přímost hranice, která se určuje poměrem mezi celkovým počtem buněk hranice a počtem hraničních buněk, ve kterých hranice mění směr. Čím více hraničních buněk je bez změny směru, tím je hranice přímější.

Délka a šířka

Pokud je objekt extrahován z obrazu, je relativně jednoduché stanovit jeho horizontální a vertikální rozměr. Tyto parametry lze stanovit sčítáním obrazových bodů objektu v jednotlivých řádcích. Vycházíme z řádků s minimálním a maximálním počtem obrazových bodů. Pro objekt s obecnou orientací nemusí být délka a šířka vertikálně a horizontálně orientována. [12]

2.5.2 Měření tvaru

Mnohdy se pro rozlišení jednotlivých objektů používá tvar. Tvarové příznaky mohou být použity samostatně nebo ve spojení s příznaky velikosti.

Pravoúhlost

Pravoúhlost se stanovuje jako poměr mezi délkou a šířkou pravoúhelníku opsaného oblasti. Měříme ji pomocí faktoru: [14]

$$R = \frac{A_0}{A_I}, \quad (9)$$

kde A_0 je plocha objektu a A_I je plocha opsaného obdélníka, který má minimální plochu. Maximální hodnota faktoru R je jedna pro obdélník nebo čtverec. Ve spojitém prostoru pro kruh $R = \pi/4$. Čím více je předmět štíhlý a zakřivený, tím menší hodnota R ; minimální hodnota R je 0.

Kruhovitost

Je několik příznaků tvaru vyjadřujících kruhovitost. Velikost těchto koeficientů vyjadřuje členitost hraniční čáry. Kruhovitost jim říkáme proto, že dosahují minima pro kruh. Často je vyjadřována kruhovitost poměrem čtverce obvodů objektů k jeho ploše, tj. podle vztahu: [14]

$$K = \frac{P^2}{A_0}, \quad (10)$$

kde P je obvod a A_0 plocha. Pro kruh $K = 4\pi$.

Protáhlost

Měříme ji faktorem: [14]

$$P = \frac{\check{S}}{D}, \quad (11)$$

kde $\check{S}$ je šířka a D délka opsaného obdélníka, který má minimální plochu. Tato jednoduchá míra nevyhoví pro podlouhlé, zakřivené objekty, typickým příkladem je podkovy.

Výstřednost

Nejjednodušší mírou výstřednosti V je poměr délek nejdelších na sebe kolmých tětiv. Jiná možnost určení výstřednosti je založena na poměru velikostí hlavních setrvačných os oblasti a vychází z určení centrálních momentů oblasti (viz kap. 2.5.3)

Nekompaktnost

Nekompaktnost je další často používaná číselná charakteristika oblasti nezávislá na lineárních transformacích. Nekompaktnost vyčíslíme podle vztahu: [14]

$$\text{nekompaktnost} = \frac{D^2}{A}, \quad (12)$$

kde D je délka hranice oblasti a A je plocha objektu. V Euklidově prostoru je nejkompaktnější kruh.

2.5.3 Momenty

Momentový popis oblastí interpretuje normalizovanou jasovou funkci obrazu jako hustotu pravděpodobnosti dvojrozměrné náhodné veličiny. Vlastnosti této veličiny lze vyjádřit prostřednictvím statistických charakteristik – momentů, které lze užít k popisu binárních i šedotónových oblastí. Obecný moment stupně (řádu) $p+q$ není invariantní vůči změně měřítka, posunutí, natočení ani vůči šedotónovým transformacím a je definován: [14]

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy, \quad (13)$$

v digitálních obrazech počítáme

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p j^q f(i, j), \quad (14)$$

kde x, y, i, j jsou souřadnice bodů oblasti. Invariantnosti vzhledem k posuvu lze docílit použitím *centrálních momentů*

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_t)^p (y - y_t)^q f(x, y) dx dy, \quad (15)$$

V digitálních obrazech

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_t)^p (j - y_t)^q f(i, j), \quad (16)$$

kde x_t a y_t jsou souřadnice těžiště oblasti, lze je získat

$$x_t = \frac{m_{10}}{m_{00}}, \quad (17)$$

$$y_t = \frac{m_{01}}{m_{00}}, \quad (18)$$

kde m_{00} udává v případě binárního obrazu velikost. Požadavek invariantnosti vůči změně měřítka vykazuje *normované centrální momenty*

$$\vartheta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^\gamma}, \quad (19)$$

$$\gamma = TRUNC\left(\frac{p+q}{2}\right) + 1, \quad (20)$$

kde $TRUNC(x)$ značí celou část x . Nezávislost momentů na otočení zajistí vhodnou volbou souřadné soustavy tak, aby centrální moment $\mu_{11} = 0$.

Potřebné hodnoty se určí podle vzorců uvedených výše. Uvedené charakteristiky jsou samozřejmě závislé na lineární šedotónové transformaci oblastí. Při popisu tvarových vlastností však vycházíme z binárních obrazů, pro obrazové elementy oblastí platí $f(i, j) = 1$, a proto tyto problémy odpadají. To znamená, že jsou tedy vhodnými charakteristikami tvaru objektu.

Poznamenejme, že momentové charakteristiky lze užít i pro oblasti reprezentované hranicí (v oblasti pak neuvažujeme díry). [14]

Hlavní momenty φ_{pq} objektu jsou momenty vztaženy k tzv. hlavní soustavě souřadné objektu.

Hlavní momenty jsou invariantní vůči posunutí a otočení objektu i vůči změně měřítka. Jsou závislé pouze na tvaru objektu. Jsou tedy vhodnými charakteristikami tvaru objektu.

Dalšími invariantními příznaky jsou pak dále elongace, disperze a extenze. Pro jejich definici je nutné definovat Legendreovu elipsu.

Legendreova elipsa

Zobrazme objekt v jeho hlavní soustavě souřadné. Hledejme elipsu E , která by měla s objektem stejné geometrické momenty až do 2. řádu včetně. Lze ukázat, že tato úloha není v obecném případě řešitelná. Řešitelnou se stane, nebudeme-li uvažovat geometrické momenty elipsy, ale její fyzikální momenty, tj. budeme chápat elipsu jako homogenní tenkou desku s konstantní hustotou, která však nemusí být 1. Geometrické momenty jsou speciálním případem momentů fyzikálních, pro hustotu rovnu 1, a proto lze požadovanou elipsu definovat takto.

Předpokládejme, že objekt G má hustotu 1. Legendreovou elipsou E objektu G nazveme elipsu s hustotou menší nebo rovnou 1, která aproximuje objekt tak, že s ním má stejné fyzikální momenty až do 2. řádu včetně. Pro poloosy této elipsy platí podle: [15]

$$A = 2\sqrt{\varphi_{20}}, \quad (21)$$

$$B = 2\sqrt{\varphi_{02}}, \quad (22)$$

kde A je velká poloosa a B je malá poloosa elipsy.

Z toho, že hustota elipsy E je menší nebo rovna 1, plyne, že Legendreova elipsa E má plochu vždy větší nebo rovnou ploše objektu G . Protože plocha objektu G v jeho hlavní soustavě souřadné je 1, je plocha E v této soustavě vždy větší nebo rovna 1. Rovnost nastane jen tehdy, je-li objekt elipsou.

Legendreova elipsa objektu v hlavní soustavě souřadné objektu má plochu větší nebo rovnou 1, přičemž rovnost nastane pouze v případě, že objektem je elipsa. Plocha Legendreovy elipsy objektu je tedy charakteristika, která udává, jak mnoho se liší objekt od elipsy. Plocha Legendreovy elipsy A_L je dána vztahem: [15]

$$A_L = \pi AB. \quad (23)$$

Elongace

Elongace EL je definována vztahem: [15]

$$DP = \log_2 \left(\frac{A}{B} \right). \quad (24)$$

Tato charakteristika udává „štíhlost“ objektu. Kruh má $EL = 0$, elipsa s poměrem poloos 1:2 má $EL = 1$ atd. Je to vhodná charakteristika tehdy, jestliže je potřebné charakterizovat tvar objektu jediným číslem, přičemž se spokojíme s aproximací objektu Legendreovou elipsou.

Disperze

Disperzi DP objektu definujeme jako: [15]

$$DP = \log_2(\pi AB). \quad (25)$$

Pro objekt, který je elipsou, pak platí $DP = 0$. Čím více se objekt liší od elipsy, tím vyšší je hodnota disperze.

Extenze

Extenze EX je součtem elongace a disperze: [15]

$$EX = EL + DP. \quad (26)$$

Extenze je charakteristika kompaktnosti objektu. Extenzi rovnu 0 má pouze kruh. Čím více se objekt liší od kruhu, tím je extenze vyšší.

Excentricita

Pro vyhodnocení excentricity objektů počítáme excentricitu odpovídající elipsy, která má stejný moment druhého řádu, tj. legendreové elipsy. Excentricita vyjadřuje vzdálenost ohniska od středu elipsy, leží v rozsahu mezi 0 a 1 (pro kružnice nabývá hodnoty 0).

2.6 Rozpoznávání obrazů

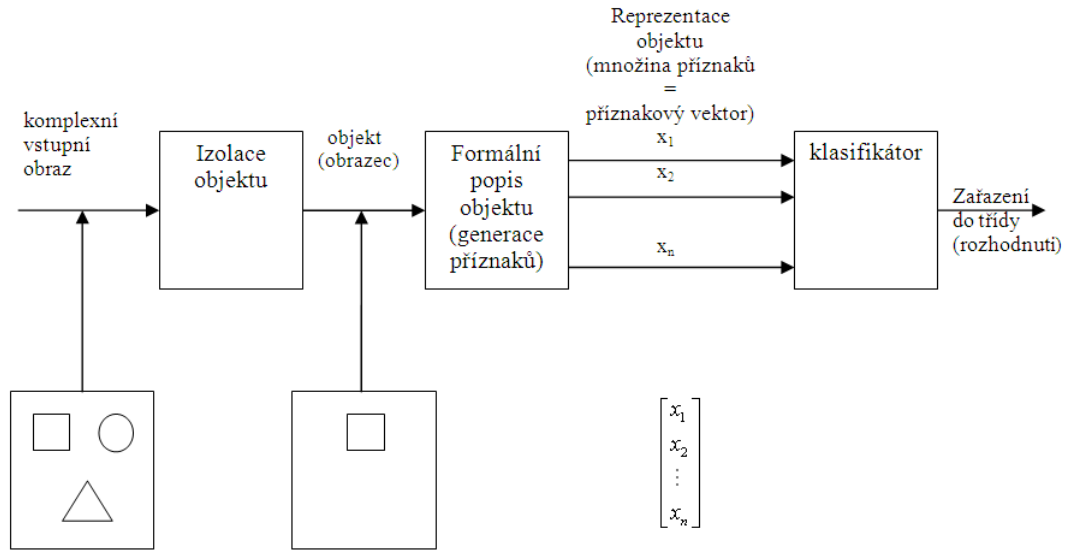
2.6.1 Lineární klasifikátor

Při rozpoznávání objektů nejprve vyhledáme objekty uvnitř obrazu a pak tyto objekty identifikujeme (klasifikujeme) podle určitých parametrů. Tyto měřené parametry nazýváme *příznaky* a značíme $x_1, \dots, x_n$. Při vhodném výběru příznaků je podobnost objektů v každé třídě vyjádřena geometrickou blízkostí jejich obrazů v obrazovém prostoru. Jednotlivým třídám odpovídají shluky obrazů, které lze zhruba oddělit vhodnou křivkou nebo u vícerozměrných obrazů nepochlou nazývanou *rozdělující nadplocha*. [14]

Zařízení, které provádí rozpoznávání (tj. klasifikaci) objektů, nazýváme klasifikátor je stroj s n vstupy a jedním výstupem.

Postup při rozpoznávání obrazců je znázorněn na obr. 4 [12]. Vstupní objekt, který máme zařadit do zvolených tříd, je nejprve vhodně popsán. Tento popis je prováděn pomocí množiny základních vlastností objektů určité třídy. Pomocí těchto základních vlastností se získá prezentace daného vstupního objektu. Takto reprezentovaný objekt je zkoumán klasifikátorem, který provede zařazení do některé z předem zvolených tříd.

Vztahy mezi vstupy a výstupem klasifikátoru popisuje funkce $\omega = d(x)$, kterou nazýváme rozhodovacím pravidlem. Rozhodovací pravidlo rozděluje obrazový prostor na R disjunktních podmnožin K_r , $r = 1, \dots, R$ tak, že podmnožina K_r obsahuje všechny obrazy x , pro které platí $\omega_r = d(x)$. Rozhraní mezi podmnožinami K_r , $r = 1, \dots, R$ tvoří již výše uvedené rozdělující nadplocha. V jejich určení spočívá návrh klasifikátoru.



Obr. 4 Systém pro příznakové rozpoznávání objektu

Rozděľující nadplocha můžeme určit R skalárními funkcemi $g_1(x), \dots, g_R(x)$ zvanými *diskriminační funkce*. Vybírají se tak, aby pro všechna $x \in K_r$ platilo podle [14]:

$$g_r(x) > g_s(x), \quad s = 1, \dots, R, \quad s \neq r. \quad (27)$$

Potom rozděľující nadplocha mezi sousedními množinami K_r a K_s má rovnici

$$g_r(x) - g_s(x) = 0. \quad (28)$$

Rozhodovací pravidlo realizované tímto klasifikátorem zní: Obraz x klasifikuje do té třídy, jejíž diskriminační funkce nabývá pro dané x maxima:

$$x: \omega_r = d(x) \iff g_r(x) = \max g_s(x), \quad s = 1, \dots, R. \quad (29)$$

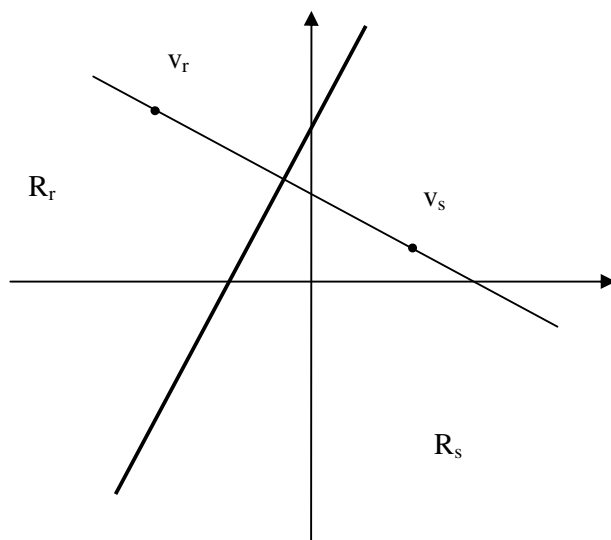
Lineární diskriminační funkce jsou nejjednodušší a často používané. Mají tvar

$$g_r(x) = q_{r,0} + q_{r,1}x_1 + \dots + q_{r,n}x_n, \quad r = 1, \dots, R. \quad (30)$$

Klasifikátor, jehož všechny diskriminační funkce jsou lineární, se nazývá *lineární klasifikátor*. Rozděľující nadplocha lineárního klasifikátoru se skládají z částí nadrovin. Jinou možností je konstruovat *klasifikátory pracující na principu minimální vzdálenosti*. Předpokládejme, že v obrazovém prostoru je zadáno R bodů $v_1, v_2, \dots, v_R$, které se nazývají etalony neboli vzorové obrazy tříd $\omega_1, \dots, \omega_R$. Klasifikátor podle minima vzdálenosti zařadí klasifikovaný obraz x do té třídy, jejíž etalon má od bodu nejmenší vzdálenost. Přítomnost ke třídě ω_r je určena

$$x: \omega_r \iff |v_r - x| = \min |v_s - x_s|. \quad (31)$$

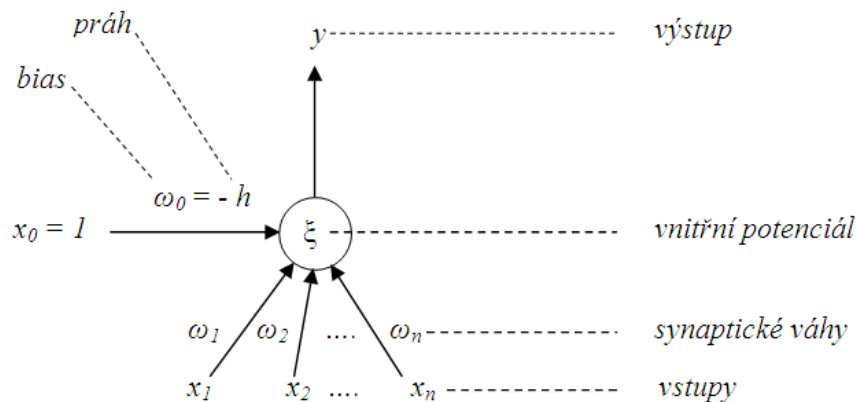
Rozděľující nadplochy tohoto klasifikátoru jsou kolmé na úsečku $v_s v_r$ a půlí ji (obr. 5). Pokud každé třídě odpovídá jeden etalon, je výsledkem lineární klasifikátor. Odpovídá-li jedné třídě více etalonů, výsledný klasifikátor má po částech lineární rozděľující nadplochy. [14]



Obr. 5 Klasifikace podle minima vzdálenosti

2.6.2 Neuronová síť jako klasifikátor

Neuronová síť se skládá z formálních neuronů (obr. 6), které jsou vzájemně propojené tak, že výstup neuronu je vstupem obecně více neuronů. Neuron má n reálných vstupů $x_1, x_2, \dots, x_n$, které ohodnoceny odpovídajícími obecně reálnými synaptickými váhami $\omega_1, \omega_2, \dots, \omega_n$, které určují jejich propustnost.



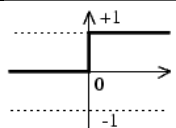
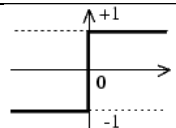
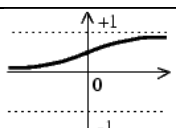
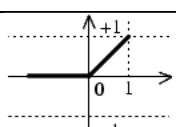
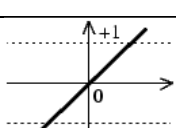
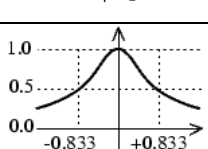
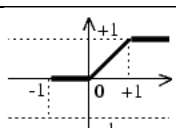
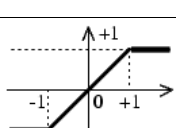
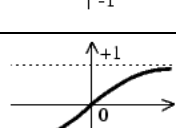
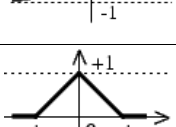
Obr. 6 Formální neuron

Zvážená suma vstupních hodnot představuje *vnitřní potenciál* neuronu podle [13]:

$$\xi = \sum_{i=1}^n \omega_i x_i, \quad (32)$$

Hodnota vnitřního potenciálu po dosažení tzv. *prahové* hodnoty h indikuje *výstup* (*stav*) neuronu y . Nelineární nárůst výstupní hodnoty při dosažení prahové hodnoty potenciálu h je dán tzv. *aktivační (přenosovou) funkcí* (Tabulka 1). Přenosová funkce udává, jaká bude odezva na výstupu na vstupní podnět. Volba funkce závisí na problému, který chceme řešit [17]. Nejjednodušším typem aktivační funkce je tzv. *ostrá nelinearita*.

Tabulka 1. Aktivační funkce neuronu

Název aktivační funkce	Matematické vyjádření	Oblast hodnot	Graf
Ostrá nelinearita	$\sigma(\xi) = \begin{cases} 1, & \xi \geq h, \\ 0, & \xi \leq h \end{cases}$	0, 1	
Znaménková funkce	$\sigma(\xi) = \begin{cases} -1, & \xi \geq h, \\ 0, & \xi = 0, \\ 1, & \xi \leq h \end{cases}$	-1, 1	
Standardní (logistická) sigmoida	$\sigma(\xi) = \frac{1}{1 + e^{-\xi}}$	(0, 1)	
Pololineární funkce	$\sigma(\xi) = \begin{cases} \xi, & \xi > 0, \\ 0, & \xi \leq 0 \end{cases}$	(0, ∞)	
Lineární	$\sigma(\xi) = \xi$	(-∞, ∞)	
Radiální bazická (Gaussova) funkce	$\sigma(\xi) = e^{-\xi^2}$	(0, 1)	
Saturovaná pololineární funkce	$\sigma(\xi) = \begin{cases} 0, & \xi \leq 0, \\ \xi, & 0 < \xi < 1, \\ 1, & \xi \geq 1 \end{cases}$	(0, 1)	
Saturovaná lineární funkce	$\sigma(\xi) = \begin{cases} -1, & \xi \leq -1, \\ \xi, & -1 < \xi < 1, \\ 1, & \xi \geq 1 \end{cases}$	(-1, 1)	
Hyperbolický tangens	$\sigma(\xi) = \frac{1 - e^{-\xi}}{1 + e^{-\xi}}$	(-1, 1)	
Trojúhelníková funkce	$\sigma(\xi) = \begin{cases} 1 - \xi , & \xi \leq 1, \\ 0, & \xi \geq 1 \end{cases}$	(0, 1)	

Učení neuronových sítí

Algoritmus se obvykle dělí na dvě fáze a to na fázi aktivační (vybavovací) a adaptační (učící), které ke své činnosti potřebují trénovací množinu. Trénovací množina je skupina vektorů obsahujících informace o daném problému pro učení [17]. Cílem učení je nastavit váhy spojení w_i tak, aby síť vytvářela správnou odezvu na vstupní signál. Základní způsoby učení:

1. Učení s učitelem.

Neuronová síť se učí srovnáním aktuálního výstupu s výstupem požadovaným (učitel) a nastavováním vah synapsí tak, aby se snížil rozdíl mezi skutečným a požadovaným výstupem. Trénovací množina se skládá ze dvou vektorů vstup – výstup.

2. Učení bez učitele.

Váhy spojení se nastavují tak, aby byl výstup sítě konzistentní, tj. aby síť poskytovala stejnou odezvu při stejných, popř. podobných vstupních vektorech. Trénovací množina obsahuje jen vstupní vektory. [16]

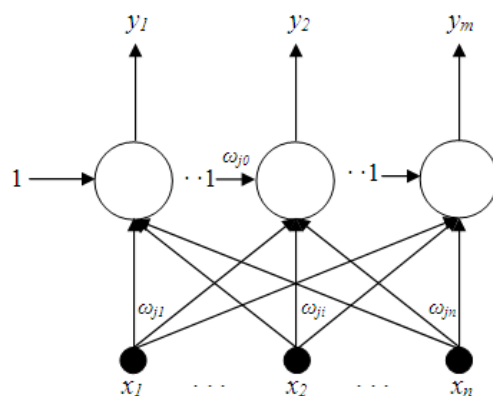
Aktivační fáze je proces, při kterém se předložený vektor informací na vstup sítě přepočítá přes všechny spoje včetně jejich ohodnocení vahami až na výstup, kde se objeví odezva sítě na tento vektor ve formě výstupního vektoru. Při učení se tento vektor se porovná s vektorem originálním (požadovaným, výstupním) a rozdíl mezi oběma vektory (*lokální odchylka* - chyba) uloží do paměťové proměnné.

Adaptační fáze je proces, při kterém je minimalizovaná lokální chyba sítě tak, že se přepočítávají váhy jednotlivých spojů směrem z výstupu na vstup za účelem co největší podobnosti výstupní odezvy s originálním vektorem. [17]

Hlavní výhodou neuronových sítí je schopnost generalizace, tzn. schopnost správně rozhodovat i v případech, které jsou pro síť nové, a není na ně přímo naučena. To je výhodné právě v případě rozpoznávání objektů v obraze, protože každý obraz může být nějakým způsobem degradován od vzoru, určeného za typického reprezentanta třídy. [15]

Síť perceptronů

Organizační dynamika této sítě specifikuje na začátku pevnou architekturu jednovrstvé sítě $n-m$. To znamená, že síť se skládá z n vstupních neuronů, z nichž každý je vstupem každého z m výstupních neuronů, jak je naznačeno na obrázku 7. Označme $x_1, \dots, x_n$ reálné stavy vstupních neuronů, tj. $\mathbf{x} = (x_1, \dots, x_n) \in R^n$ je vstup sítě, a $y_1, \dots, y_m$ binární stavy výstupních neuronů, tj. $\mathbf{y} = (y_1, \dots, y_m) \in \{0, 1\}^m$ je výstup sítě. Dále ω_{ji} představuje reálnou synaptickou váhu spoje od i -tého vstupního k j -tému výstupnímu neuronu a $\omega_{j0} = -h_j$ je bias (práh h_j s opačným znaménkem) j -tého výstupního neuronu odpovídající formálnímu jednotkovému vstupu $x_0 = 1$.



Obr. 7 Architektura sítě perceptronů

Aktivní dynamika sítě perceptronů určuje způsob výpočtu funkce sítě. V tomto případě se reálné stavy neuronů ve vstupní vrstvě nastaví na vstup sítě a výstupní neurony počítají svůj binární stav, který určuje výstup sítě, stejným způsobem jako formální neuron (viz rovnice (33)). Stav perceptronu se pak určí z jeho vnitřního potenciálu aplikací aktivační funkce σ , která má tvar ostré nelinearity.

V adaptivním režimu je požadovaná funkce sítě perceptronů zadána tréninkovou množinou: [13]

$$T = \left\{ (x_k, d_k) \left| \begin{array}{l} x_k = (x_{k1}, \dots, x_{kn}) \in R^n \\ d_k = (d_{k1}, \dots, d_{km}) \in \{0,1\}^m \end{array} \right. \quad k = 1, \dots, p \right\}, \quad (33)$$

kde $\mathbf{x}_k$ je reálný vstup k -tého tréninkového vzoru a $\mathbf{d}_k$ je odpovídající požadovaný binární výstup (daný učitelem). Cílem adaptace je, aby síť pro každý vstup $\mathbf{x}_k$ z tréninkové množiny odpovídala v aktivním režimu požadovaným výstupem $\mathbf{d}_k$, tj. aby platilo: [13]

$$y(\omega, x_k) = d_k \quad k = 1, \dots, p \quad (34)$$

Generalizační schopnost tohoto modelu není velká, protože síť perceptronů lze použít jen v případě, kdy klasifikované objekty jsou ve vstupním prostoru oddělené nadrovinou. Tento jednoduchý model je však základem složitějších modelů jako obecná vícevrstvá síť s učícím algoritmem backpropagation.

Vícevrstvá síť perceptronů a backpropagation

Tento model je zobecněním sítě perceptronů pro architekturu se skrytými vrstvami (tzv. vícevrstvý perceptron). Backpropagation často využívá především metody nejstrmějšího klesání gradientu chybové funkce.

Algoritmus trénování vícevrstvého perceptronu podle [16].

1. Počáteční inicializace vah $\omega_{ji}^{(0)} \in (-1, 1)$.
2. Přivedení vstupního vektoru $x = [x_1, \dots, x_n]^T$ a definice požadované výstupní odezvy $d = [d_1, \dots, d_m]^T$.
3. Pro každý tréninkový $q = 1, \dots, p$ vzor děláme:
 - a) Výpočet aktuálního výstupu podle následujících vztahů (platí pro třívrstvý perceptron, $j = 1, 2, 3$)

$$y_l(t) = \sigma \left(\sum_{k=1}^{n_2} \omega_{kl}''(t) x_k''(t) \right), \quad 1 \leq l \leq m, \text{ výstupní vrstva (j = 3)} \quad (35)$$

$$x_k''(t) = \sigma \left(\sum_{j=1}^{n_1} \omega_{jk}'(t) x_j'(t) \right), \quad 1 \leq k \leq n_2, \text{ 2. skrytá vrstva (j = 2)} \quad (36)$$

$$x_i'(t) = \sigma \left(\sum_{r=1}^n \omega_{ri}(t) x_r(t) \right), \quad 1 \leq i \leq n_1 \text{ vstupní vrstva (j = 1), } 1 \leq r \leq n \quad (37)$$

b) Adaptace vah podle vztahu

$$\omega_{ji}(t+1) = \omega_{ji}(t) + \eta \delta_j x_i. \quad (38)$$

Nastavení vah začíná u výstupních uzlů a postupuje rekurzivně směrem ke vstupům. V uvedeném vztahu jsou ω_{ji} váhy mezi j -tým skrytým uzlem (popř. vstupním uzlem) a uzlem i -tým v čase t , x_j je výstup j -tého, popř. vstupního uzlu, η je koeficient učení, δ_i je chyba, pro kterou platí následující vztahy:

$$\delta_j = \frac{1}{2} \sum_{q=1}^3 (y_j(w, x_q) - d_{qj})^2. \quad (39)$$

Opakování kroků 3a) a 3b), dokud chyba není menší než předem stanovená hodnota.

RBF síť

RBF síť si můžeme představit jako třívrstvou síť, kde vstupní vrstva neuronů slouží pouze k přenosu vstupních hodnot (obr. 8). Druhá (skrytá) vrstva sestává z tzv. *RBF jednotek*, které realizují jednotkové radiální funkce. Třetí, výstupní vrstva je lineární. RBF jednotka je podobná perceptronu – má také n reálných vstupů $\mathbf{x} = (x_1, \dots, x_n)$, z nichž každý má přiřazen parametr (váhu) c_i , jeden reálný výstup y a může mít další parametr b , kterému budeme říkat *šířka*. Ovšem přechodová funkce RBF jednotky je odlišná: vnitřní potenciál ξ se nepočítá jako skalární součin $\mathbf{w} \cdot \mathbf{x}$, ale je to vzdálenost vstupního vektoru $\mathbf{x}$ od středu c (případně ještě dělena šířkou b) [13]:

$$\xi = \frac{\|\mathbf{x} - c\|}{b}. \quad (40)$$

Výstupní hodnotu y získáme aplikováním aktivační funkce φ na potenciál ξ :

$$y = \varphi(\xi). \quad (41)$$

Nejčastěji používaná aktivační funkce je monotónně klesající směrem od středu Gaussova funkce

$$\varphi(x) = e^{-\left(\frac{\|\mathbf{x} - c\|}{b}\right)^2}. \quad (42)$$

Parametr b pak určuje její strmost. Táto funkce v tomto případě vyjadřuje míru příslušnosti vzoru k prototypu.

Neurony ve výstupní vrstvě jsou perceptronového typu a definice jejich výstupní funkce je následující: [13]

$$f(x) = \sum_{i=1}^n w_i h_i(x) \quad (43)$$

Učící algoritmus sestává ze tří fází, z nichž v každé se určují hodnoty jiné skupiny parametrů. V průběhu učení postupujeme v síti vzhůru a postupně se zabýváme souřadnicemi středů jednotek, jejich šířkami a naposledy koeficienty lineární kombinace.

Tréninkovou množinu učení RBF sítě tvoří páry vektorů $\{(x^{(t)}, d^{(t)}); t = 1, \dots, k\}$ sestávající ze vstupů $x^{(t)} \in R^n$ a požadovaných výstupů $d^{(t)} \in R^m$. První krok určuje pozice středů RBF jednotek, jež jsou reprezentovány vahami mezi vstupní a skrytou vrstvou. Jelikož v této fázi jde o aproximaci hustoty výskytu vzorů, používají se různé techniky samoorganizačního učení.

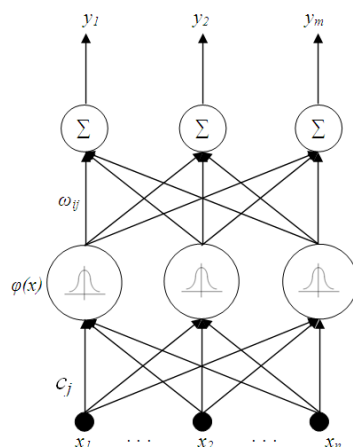
Druhá fáze nastavuje hodnoty případných dalších parametrů u RBF jednotek. Obvykle mají jednotky nastavitelný parametr, který určuje šířku oblasti kolem středu, v níž má jednotka relevantní výstup.

Třetí krok je obvyklé učení s učitelem, které adaptuje váhy mezi skrytou a výstupní vrstvou. Jelikož jde o koeficienty lineární kombinace, je učení lineární a není také nutné šířit chybu do nižších vrstev, protože jejich váhy jsou už fixovány. Toto učení je tedy mnohem jednodušší než například u standardního zpětného šíření, ale jeho úspěch závisí na předchozích krocích [13]. Optimální váhy w_i této sítě se určují pomocí minimalizace nějaké kritériální funkce. Typický příklad je součet kvadrátů odchylek skutečné hodnoty od požadované. Minimalizace kvadrátů odchylek vede na následující rovnici, pomocí níž lze optimální váhy spočítat: [15]

$$w = (H^T H)^{-1} H^T y \quad (44)$$

$$H = \begin{pmatrix} h_1(x_1) & h_2(x_1) & \dots & h_m(x_1) \\ h_1(x_2) & h_2(x_2) & \dots & h_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_p) & h_2(x_p) & \dots & h_m(x_p) \end{pmatrix} \quad (45)$$

kde y je vektor velikosti m , který obsahuje m výstupů z trénovací množiny a x je n vstupních vektorů z trénovací množiny velikosti rovné počtu vstupů.



Obr. 8 Architektura RBF sítě

Kohonenova síť

Kohonenova síť patří k samoorganizujícím neuronovým sítím, což znamená, že nepotřebují ke svému trénování učitele. Její funkce je založena na principu shlukové analýzy (nalezení určitých vlastností a závislostí přímo v předkládaných trénovacích datech). Tento model obsahuje jen vstupní a výstupní vrstvu, která se prezentuje plošným uspořádání neuronů.

Výpočetní rychlost této sítě je dána množstvím neuronů. Pokud budeme mít velké množství neuronů, pak dosáhneme toho, že vstupní prostor bude velmi dobře pokryt a naopak.

Označme vstupní vektor neuronu $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, $\mathbf{x}_i \in R$. Zvláštností u této sítě je, že neurony nemají přenosovou funkci, a proto výstupní hodnota neuronů je definována jako vzdálenost mezi vstupním a váhovým vektorem, tj.

$$y_i(t) = \sum_{j=1}^n (x_j(t) - \omega_{ij}(t))^2. \quad (46)$$

Algoritmus Kohonenove sítě [16].

1. Počáteční inicializace vah ω_{ij} a množiny $Ne_i(t)$, která definuje sousednost okolo každého uzlu $i = 1, 2, \dots, n$. Dále se nastaví parametr učení na libovolnou hodnotu v intervalu od 0 do 1. Parametr učení ovlivňuje rychlost učení (adaptace vah). Vzhledem k tomu, že učení spočívá ve hledání vhodného neuronu a jeho okolí, musí se nastavit maximální a minimální hranice okolí neuronu. Ta se obvykle volí pro maximum - počet neuronů na straně plochy a pro minimum - jeden neuron. [17]

2. Přivedení vektoru $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ na vstup sítě a výpočet vzdálenosti mezi vstupním vektorem a váhovým vektorem každého neuronu podle vztahu

$$d_i = \sum_{j=1}^n (x_j(t) - \omega_{ij}(t))^2. \quad (47)$$

3. Výběr i -tého neuronu s nejmenší vzdáleností d_i (tzv. vítězného neuronu)

4. Adaptace vah vítězného neuronu (viz rovnice 49) a všech neuronů k (viz rovnice 50 a obr. 9), pro které platí $k \in Ne_i(t)$ (sousedí s i -tým neuronem) podle vztahu: [16]

$$\omega_{ki}(t+1) = \omega_{ki}(t) + \alpha(t)(x_i(t) - \omega_{ki}(t)), \quad \forall k \in Ne_i(t), \quad i = 1, 2, \dots, n, \quad (48)$$

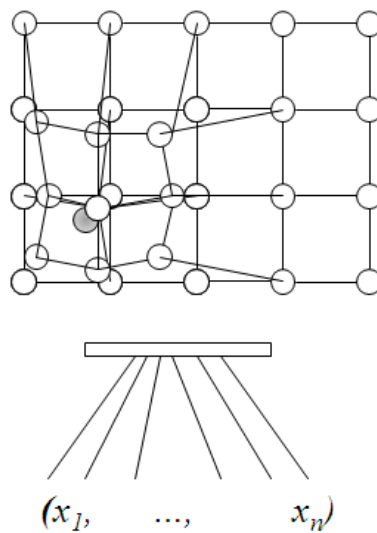
$$\omega_{ki}(t+1) = \omega_{ki}(t) + \eta(t)(x_i(t) - \omega_{ki}(t)), \quad \forall k \in Ne_i(t), \quad i = 1, 2, \dots, n, \quad (49)$$

kde $\eta(t)$ ($0 < \eta(t) < 1$) je koeficienty učení, jehož hodnota s rostoucím časem klesá:

$$\alpha(t+1) = 0,5\alpha(t), \quad (50)$$

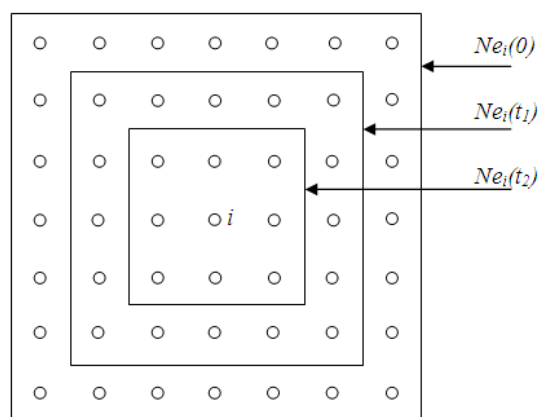
$$\eta(t+1) = 0,5\eta(t), \quad (51)$$

Neuron, jehož váhy byly adaptovány, se automaticky stává citlivějším na vektory, jenž jsou podobné vektoru, na nějž se právě adaptoval. Jak už bylo na začátku řečeno, vytváří se shluky – třídy, které odpovídají průměru všech vektorů, které do této třídy patří [17].



Obr. 9 Architektura Kohonenove sítě

5. Změna sousednosti $Ne_i(t)$ pro $i = 1, 2, \dots, n$ (viz obr. 10)



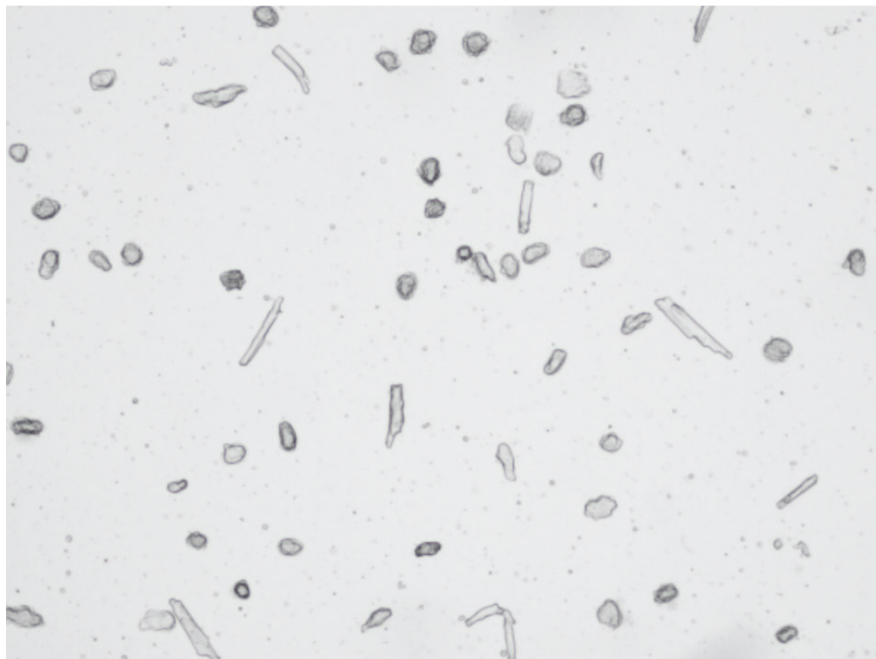
Obr. 10 Znažornění sousednosti

6. Opakování kroků 2-5 pro všechny vstupní vektory, dokud dochází ke změně vah.

3. Návrh vlastního řešení

3.1 Snímání mikroskopického obrazu

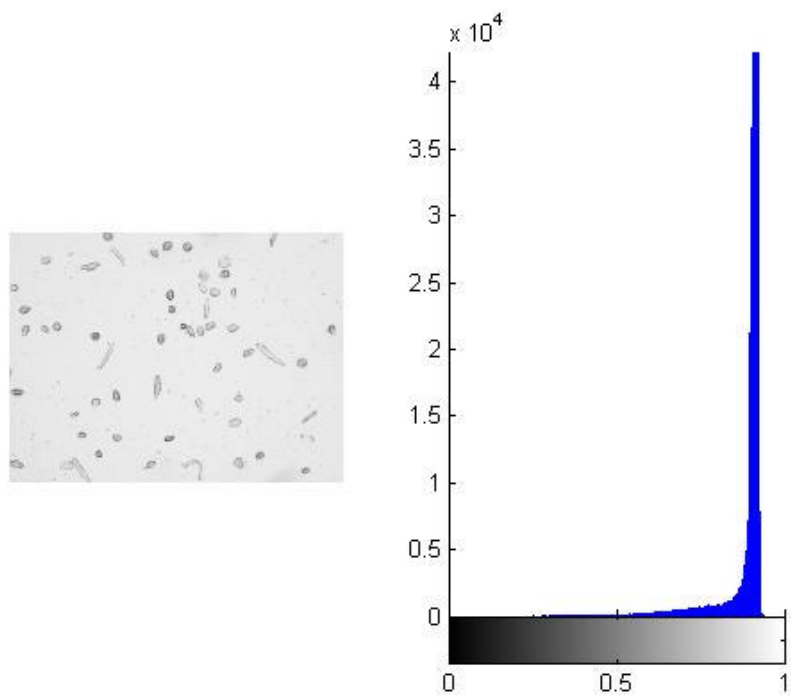
Snímky použité v této práci byly získány kamerou A622 PixeLINK v rozlišení 1024*768 pixelů při stonásobném zvětšení. Cílem snímání je získání obrázků, které mají požadované vlastnosti: rovnoměrné osvětlení, zaostření snímku. Na kvalitu původního obrázku má také vliv rozmístění buněk: musí jich být dostatečné množství a zároveň musím být docíleno toho, aby se vzájemně nepřekrývaly. Na obr. 11 je zobrazen příklad původního mikroskopického snímku, který obsahuje jak živé, tak i mrtvé buňky.



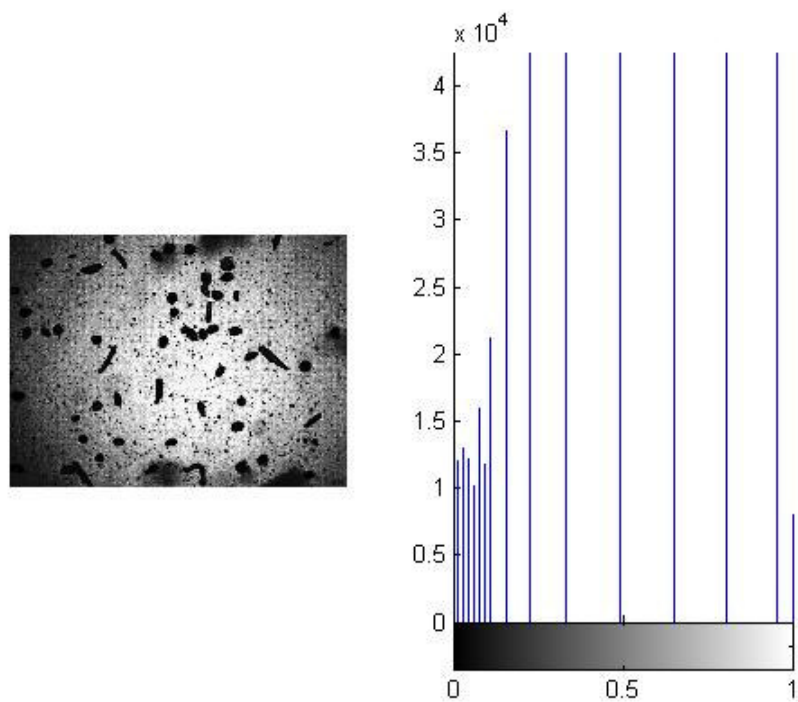
Obr. 11 Původní mikroskopický obrázek

3.2 Aplikace ekvalizace histogramu

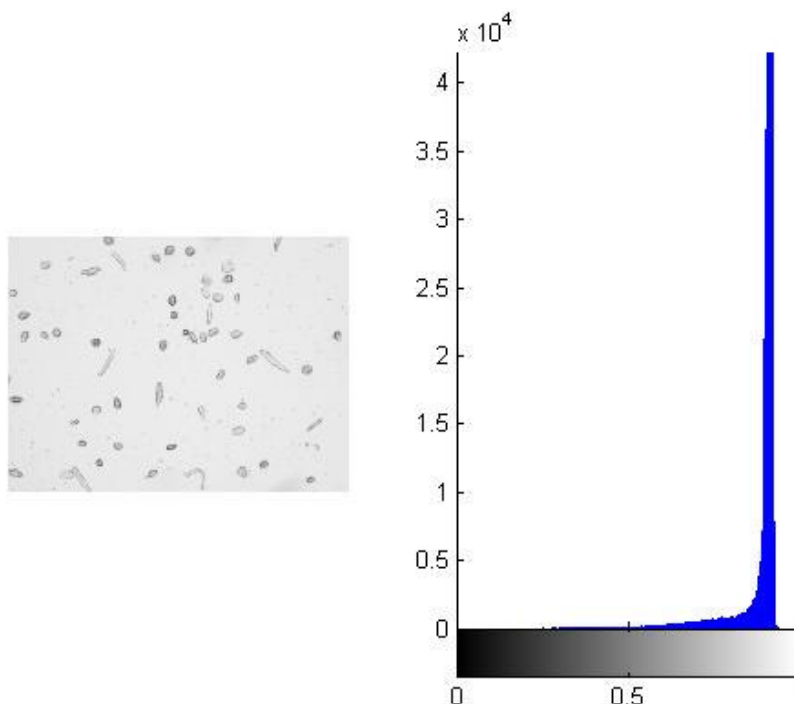
Ekvalizace histogramu zvýší kontrast pro úrovně jasu blízko maxima histogramu a sníží kontrast blízko minima histogramu. V daném případě ekvalizace histogramu (viz obr. 12 a obr. 13) přivádí ke zvyšování kontrastu, ale zároveň i k zvětšení kontrastu šumu v pozadí, což může mít negativní vliv na další segmentaci. Mnohem lepšího výsledku, který je vhodný pro další zpracování, můžeme dosáhnout použitím adaptivní ekvalizace histogramu (viz obr. 12 a obr. 14). Velikost jednotlivých částí obrazu, na které působíme adaptivní ekvalizací histogramu je 8×8 .



Obr. 12 Histogram původního obrazu



Obr. 13 Ekvalizace histogramu



Obr. 14 Adaptivní ekvalizace histogramu

3.3 Srovnávání různých typů filtrů

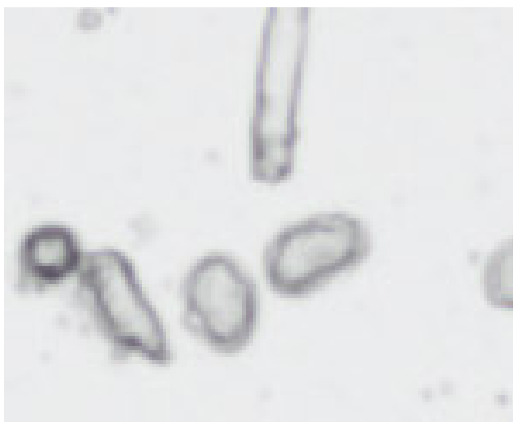
V prostředí MATLAB [19] byly pomocí existujících funkcí vyzkoušeny různé typy filtrace (vstupní obrázek na obr. 15a). Po filtraci Gaussianem (obr. 15b) má obrázek dostatečně ostré hrany, ale zůstaly zvýrazněnými elementy pozadí. Nevýhodou průměrovacích filtrů (obr. 15d, 15e) je rozmazávání hran. Ostřicí filtr dobře zvýrazňuje hrany, ale zároveň i malé nežádoucí details (viz obr. 15f). Kompromisu mezi vyhlazováním pozadí a ostrostí hran lze dosáhnout použitím mediánového filtru (viz obr 15c).

3.4 Volba optimálního způsobu segmentace

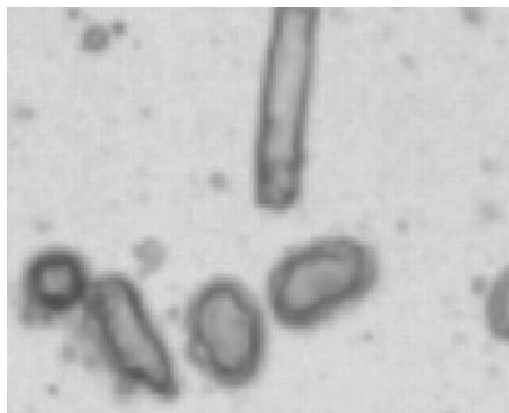
Segmentace prahováním

Při segmentaci po mediánové filtraci (vstupní obrázek na obr. 16a) metodou globálního prahování (obr. 16b) byly vyzkoušeny různé hodnoty prahu z toho důvodu, že histogram nemá bimodální povahu a proto nemůžeme ho jednoznačně stanovit (obr. 12). Při hodnotě prahu 0,635 byl získán binární obrázek, přičemž některé objekty byly „roztrhány“, tím se zvětšuje jejich celkové množství. Jako výhodu tohoto prahování je možné označit dostatečné odstranění elementů pozadí.

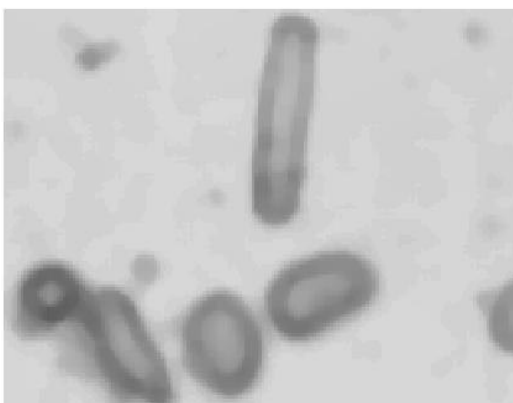
Další vyzkoušenou metodou bylo globální prahování. Hodnota prahu je vypočítána pomocí algoritmu Otsu a je rovná 0,6902 (obr. 16b). Hranice objektů ve výsledném obrazu mají spojitý charakter. Nedostatkem metody je propojení objektů, které se ve skutečnosti nepřekrývají.



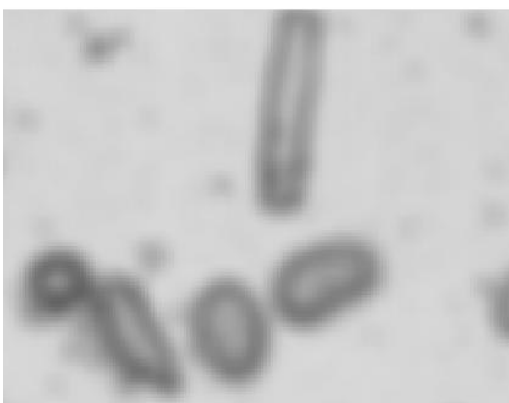
a) Vstupní obrázek



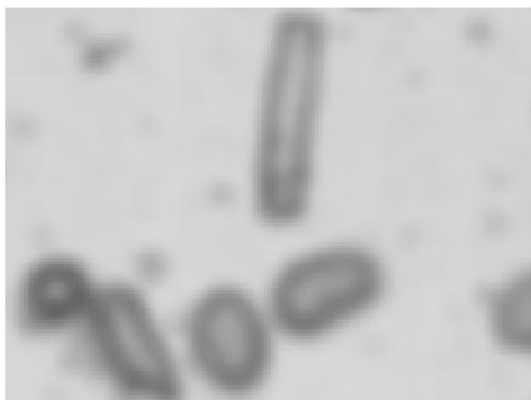
b) Filtrace Gaussianem s parametrem $\sigma = 0,3$



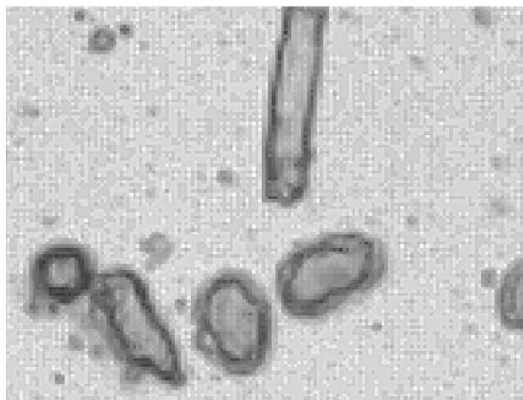
c) Mediánová filtrace
(rozměr masky 7x7)



d) Filtrace průměrovacím filtrem
(rozměr masky 7x7)

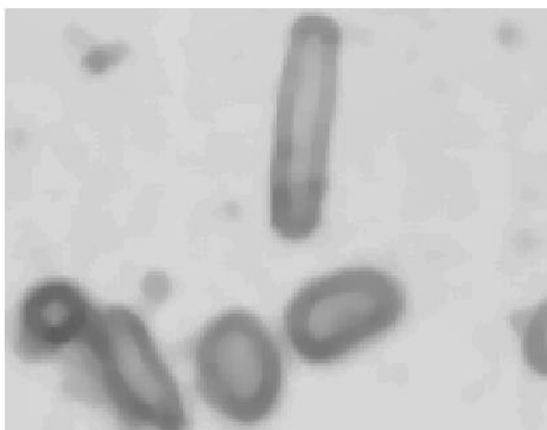


e) Filtrace průměrovacím filtrem
s konvoluční maskou ve tvaru kruhu
s hodnotou poloměru 3 a různou vahou
jednotlivých prvků masky



f) Filtrace ostřícím filtrem
(rozměr masky 3x3)

Obr. 15 Výsledky filtrace



a) Vstupní obrázek



b) Globální prahování ($T=0,635$)



c) Prahování metodou Otsu ($T = 0,6902$)



d) Lokální prahování

Obr. 16 Výsledky segmentace prahováním

Při použití metody lokálního prahování (obr. 16d) byl získán přijatelně kvalitní výsledek, protože nebyly propojeny objekty a také nedošlo k přerušení hranic. Jedna nevýhoda této metody spočívá v zachování malých detailů pozadí, ale tento nedostatek je odstranitelný při dalším zpracování. Celý obrázek byl rozdělen na části o stejné velikosti 256×256 a pro každou část byla vypočítána prahová hodnota metodou Otsu (viz obr. 17).

0,6824	0,6627	0,6980	0,7216
0,6706	0,6863	0,6941	0,6980
0,6745	0,6784	0,7020	0,7098

Obr. 17 Prahové hodnoty jednotlivých částí

Segmentace detekcí hran

Prostředí MATLAB [19] má k dispozici funkce pomocí kterých je možno realizovat různé typy detektorů hran. Operace segmentace se působí na vstupní snímek po mediánové filtraci (obr. 18a). Robertsův operátor je dost citlivý na šum, protože využívá pouze malé lokální okolí obrazového elementu (obr. 18c). Odsud vyplývá nespojitost hran a zařazení pixelu do hrany při malou změně obrazové funkce. Po aplikace Sobelova a Prewittové operatorů byly obdrženy přibližně stejné výsledky, ale s větší spojitostí hran (obr. 18a, 18b).



a) Vstupní obrázek



b) Sobelův detektor hran



c) Prewittové detektor hran



d) Robertsův detektor hran



e) Detekce po Laplace-Gaussoé filtraci



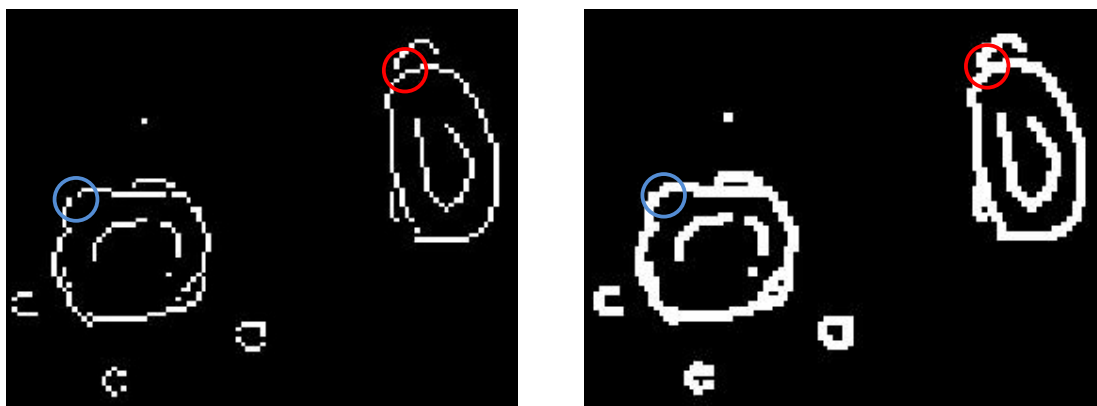
f) Canny detektor hran

Obr. 18 Různé typy detektorů hran

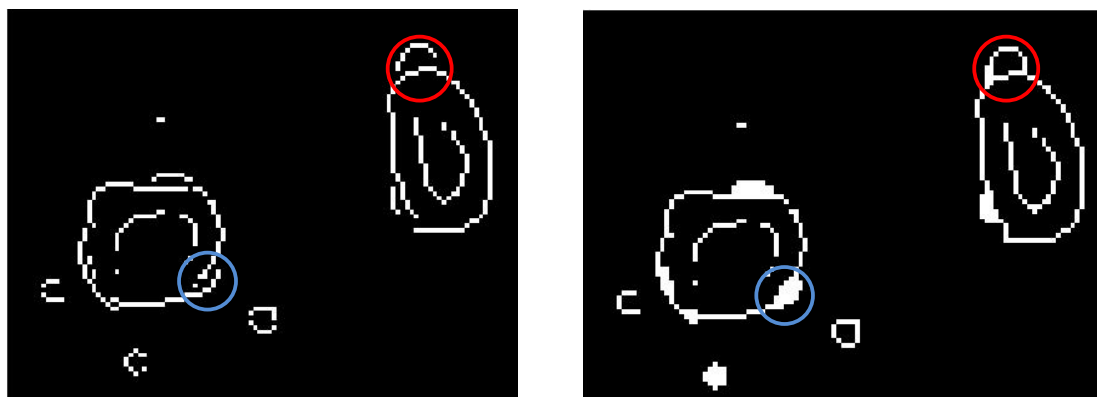
Kvalitnější výsledky byly dosaženy při použití Cannyho detektoru hran (viz obr. 18f) a detekce hran založené na průchodu druhé derivace obrazové funkce nulou po filtraci Laplace-Gaussovým filtrem (viz obr. 18d). Hrany elementů obrazu mají spojitý charakter, ale v případě Cannyho detektoru byli detekovány dokonce i nepatrné změny intenzity pozadí, které pak mohou být těžko odstranitelné při dalším zpracování, protože jejich velikost je srovnatelná s velikostí buňky.

3.5 Použití morfologických operace

Aplikace morfologických operací binární dilatace a binárního zavření obrazů dává dobré výsledky pro zpracování segmentovaných hranovými detektory obrazů. Po jejich použití došlo ke spojení přerušených hran (modrá čára na obr. 19 a na obr. 20), ale zároveň i k připojení předmětů okolí do těla buňky (červená čára na obr. 19 a na obr. 20).

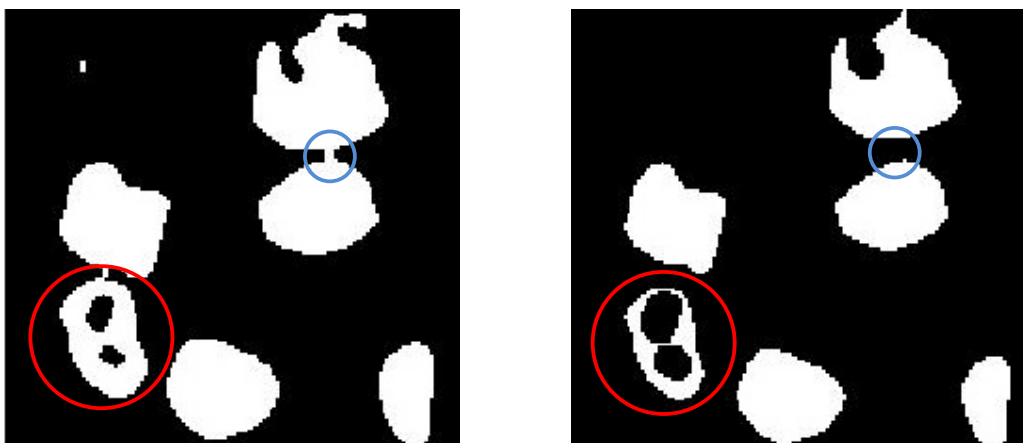


Obr. 19 Pozitivní (—) a negativní (—) vliv dilatace



Obr. 20 Pozitivní (—) a negativní (—) vliv binárního zavření obrazu

Morfologické operce binární erose a binárního otevření obrazu je vhodnými pro úpravu snímků po lokálním respektive globálním prahování. Jejich pozitivní vliv spočívá v rozpojení buněk (modrá čára na obr. 21 a na obr. 22), ale existuje i negativní vliv – zvětšení vnitřních „děr“ objektu, které pak mohou být srovnatelné s rozměry buněk (červená čára na obr. 21) nebo porušení celistvosti objektu (červená čára na obr. 22).



Obr. 21 Pozitivní (—) a negativní (—) vliv erose



Obr. 22 Pozitivní (—) a negativní (—) vliv binárního otevření obrazu

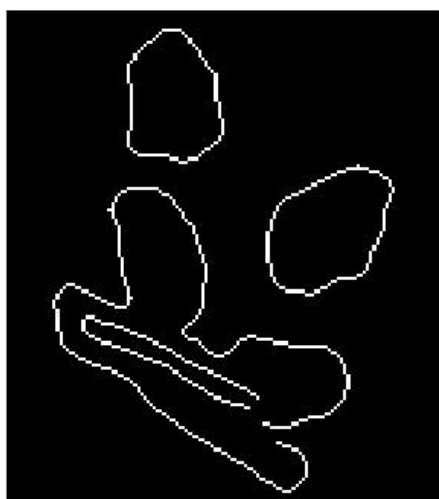
3.6 Postup klasifikace

Před samotnou klasifikací je třeba provést odstranění nežádoucích objektů, které zůstali po zpracování.

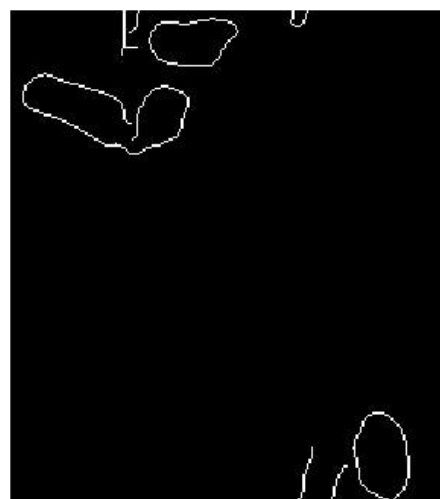
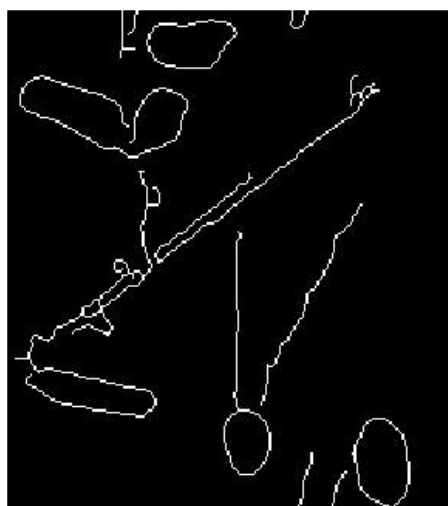
1. Odstranění příliš malých objektů. Vypočítáme plochu objektu jako součet odpovídajících pixelů a vybíráme pouze objekty s plochou více než nastavená prahová hodnota $T = 70$ (viz obr. 23a).
2. Odstranění příliš tenkých objektů. Tato operace je realizována změřením malé osy odpovídající legendreové elipsy a pak eliminace těch objektů, které mají hodnotu tohoto příznaku menší než určitá experimentálně stanovená hranice (viz obr. 23b).
3. Odstranění příliš dlouhých objektů. Postup je analogický, ale vypočítáme velkou osu odpovídající legendreové elipsy a odstraňujeme objekty s větší osou než zkušebně nalezená mez (viz obr. 23c).



a) Odstranění malých objektů



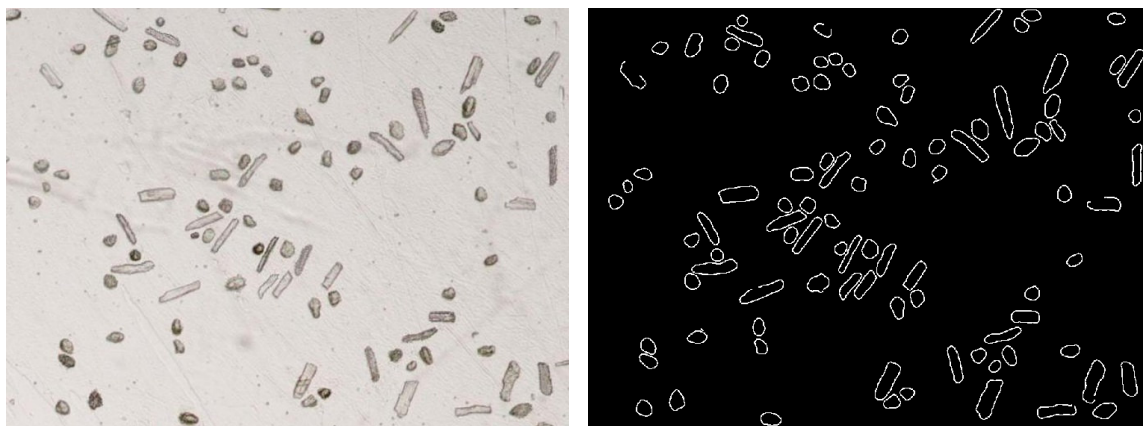
b) Odstranění malých objektů



c) Odstranění tenkých objektů

Obr. 23 Odstranění nežádoucích objektů

Pro ověření činností klasifikátorů byl jeden obrázek byl upraven tak aby bylo možné co nejpřesněji definovat typ buněk, tzv. testovací množina (viz obr. 24). Tato trénovací množina obsahuje 31 živých a 68 mrtvých buněk, což v procentním obsahu je 31,31 % a 68,68 %.

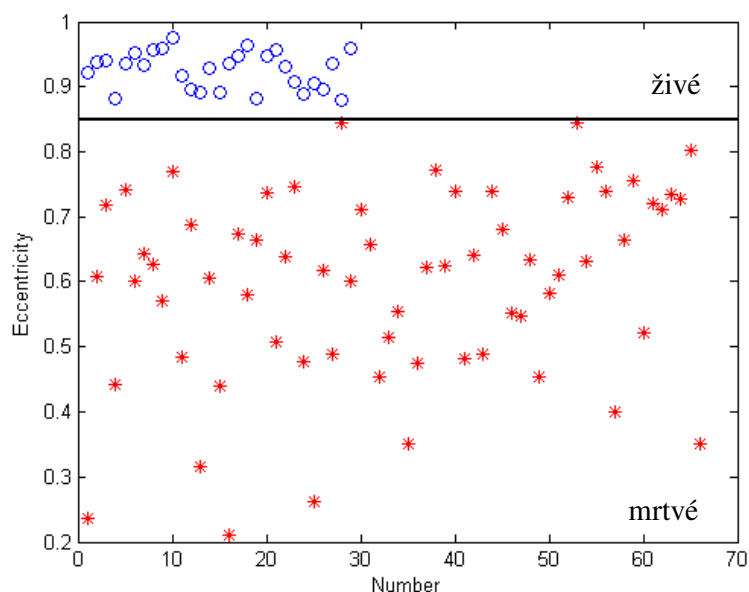


Obr. 24 Testovací množina

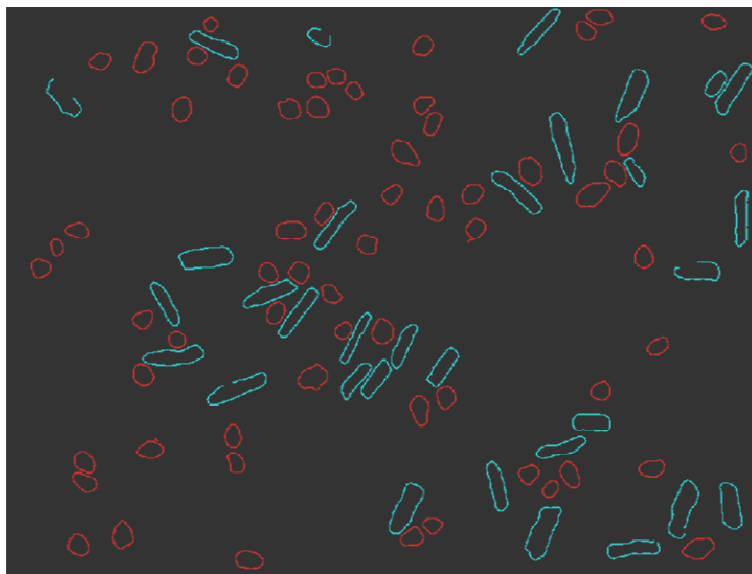
3.6.1 Klasifikátor založený na příznakovém popisu.

Rozhodujícím příznakem pro rozlišení mrtvých a živých buněk podle tvaru byla zvolena excentricita odpovídající elipsy. Prahová hodnota excentricity se rovná v tomto případě 0,85. To znamená, že je-li hodnota excentricity více, než 0,85 odpovídající objekt se zařadí do třídy živých buněk (modrá čára na obr. 25) a když je méně, než 0,85 daný objekt je mrtvou buňkou (červená čára na obr. 25).

V důsledku této klasifikace je získán počet živých a mrtvých buněk v daném mikroskopickém obrázku (obr. 26): 29 živých a 66 mrtvých v procentuálním vyjádření 30,53 % a 69,47 %. To se od skutečných hodnot liší chybou 6,45 % a 1,47 %. Tato chyba vzniká důsledkem falešně negativního vyhodnocení dvou mrtvých buněk.



Obr. 25 Rozložení hodnot excentricity objektů



Obr. 26 Výsledek příznakového klasifikátoru

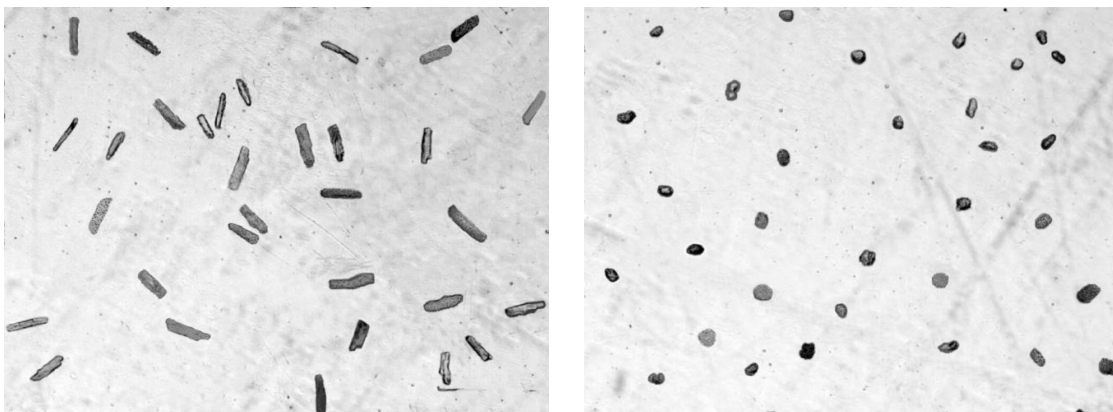
3.6.2 Perceptronové sítě s algoritmem učení backpropagation

Jednovrstvá perceptronová síť

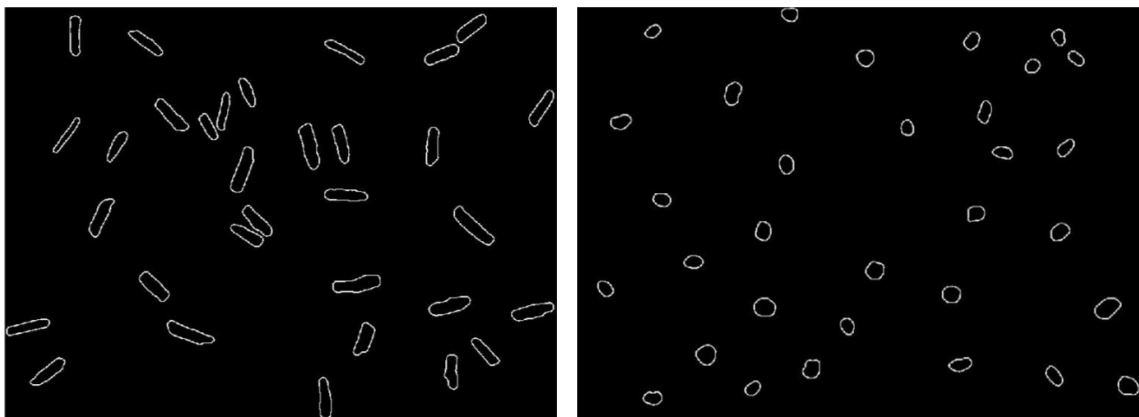
Pro trénování sítě jsou vytvořeny obrázky, které obsahují jenom živé nebo jenom mrtvé buňky, tzv. etalony (viz obr. 27), které byly zpracovány následujícím způsobem:

- filtrace mediánovým filtrem s maskou 8×8 ,
- dilatace se čtvercovým strukturním elementem 4×4 ,
- segmentace založena na průchodu druhé derivace obrazové funkce nulou po filtraci Laplace-Gaussovým filtrem,
- odstranění nežádoucích objektů.

Obdržíme obrázky, v nichž je stejný počet objektů – 32 (viz obr. 28). Jednotlivé trénovací vzory mají různou velikost a orientace v prostoru. Toho se využije při učení sítě, v důsledku toho síť bude mít větší generalizační schopnost.



Obr. 27 Zvolená trénovací množina



Obr. 28 Trénovací množina po zpracování

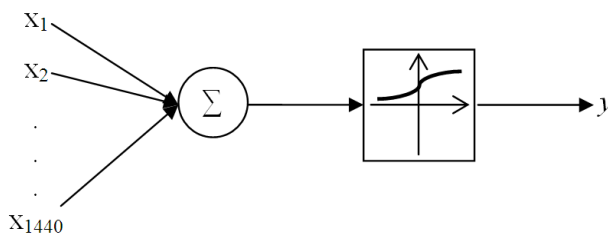


Obr. 29 Příklad trénovacích vzorků

Aby každý trénovací vzorek měl stejnou velikost, buňky jsou izolovaně umístěny uprostřed čtverce velikosti 120×120 (obr. 29). Čtvercová obrazová matice, je předělena na jeden sloupec rozměrností 14400×1 , který je pak přiváděn na vstup a který představuje binární obrazovou datu. Na výstupu je hodnota y , která je v rozmezí mezi 0 a 1. Pravidlem pro stanovení příslušenství objektu k určité třídě je:

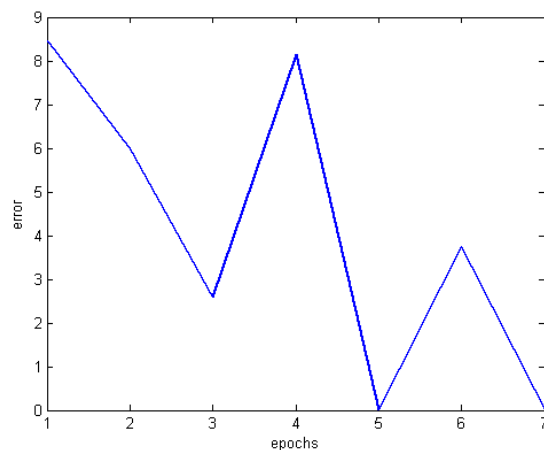
- $0,5 \leq y \leq 1 \Rightarrow$ živá buňka,
- $0 \leq y \leq 0,6 \Rightarrow$ mrtvá buňka,
- $0,5 < y < 0,6 \Rightarrow$ síť „pochybuje“.

Struktura sítě je znázorněna na obrázku 30, aktivační funkce neuronu je logistická.

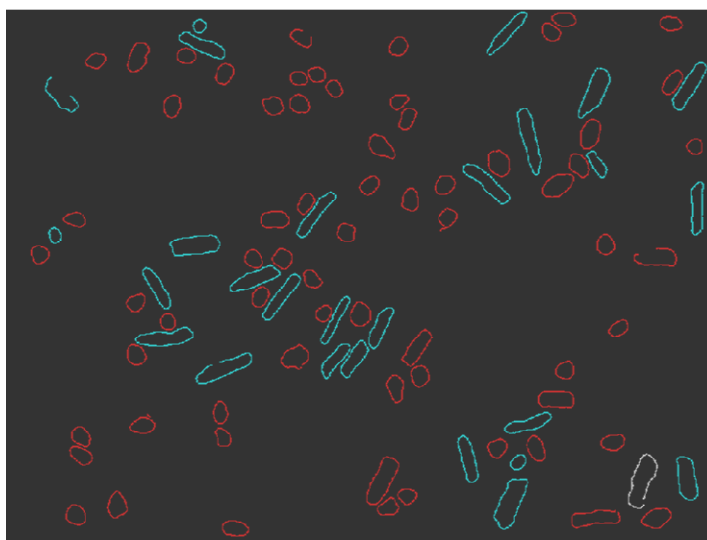


Obr. 30 Struktura jednovrstvé sítě perceptronů

Proces učení, tj. úprava vah podle algoritmu backpropagation (viz Příloha 1), trvá 7 epoch a končí po dosažení požadované hodnoty chyby 10^{-4} (viz obr. 31). Touto sítí bylo správně rozpoznáno 23 živých a 62 mrtvých buněk, což v procentním poměru je 23,23% a 62,62%. Sít' nebyla schopna rozpoznat jednu buňku (viz obr. 32 – bílá buňka). Při porovnání se skutečnými hodnotami 31,31 % a 68,68 % můžeme konstatovat, že účinnost tohoto klasifikátoru je horší než lineárního klasifikátoru. Větší chyba 25,8 % a 8,82 % je důsledkem falešné detekce: čtyři mrtvé buňky byly rozpoznány jako živé a pět živých jako mrtvé.



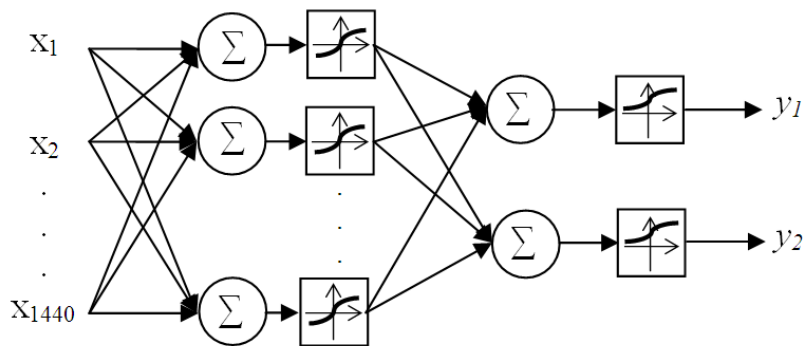
Obr. 31 Změna střední chyby v průběhu učení



Obr. 32 Výsledek klasifikace jednovrstvé sítě perceptronů

Dvouvrstvá perceptronová síť

Trénovací množina je analogická s předchozím případem. Síť se skládá ze dvou vrstev, neurony první vrstvy, jejich počet je 500, mají tangentovou aktivační funkci, neurony druhé (výstupní vrstvy) mají logistickou aktivační funkci [10]. Struktura sítě je na obr. 33.

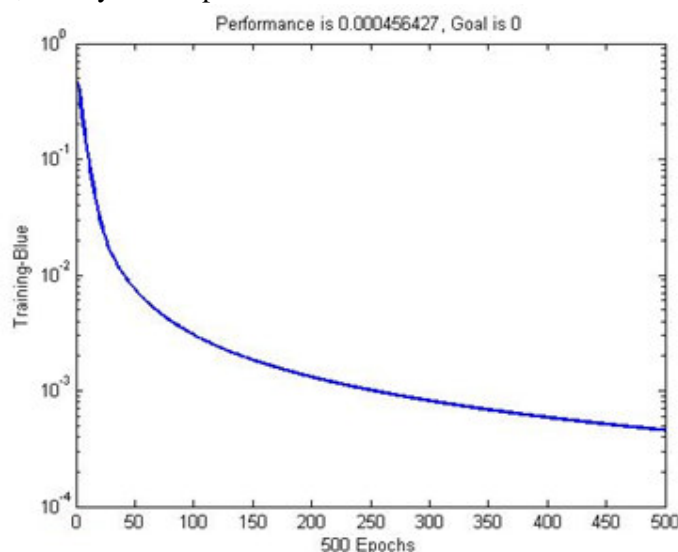


Obr. 33 Struktura dvouvrstvé perceptronové sítě

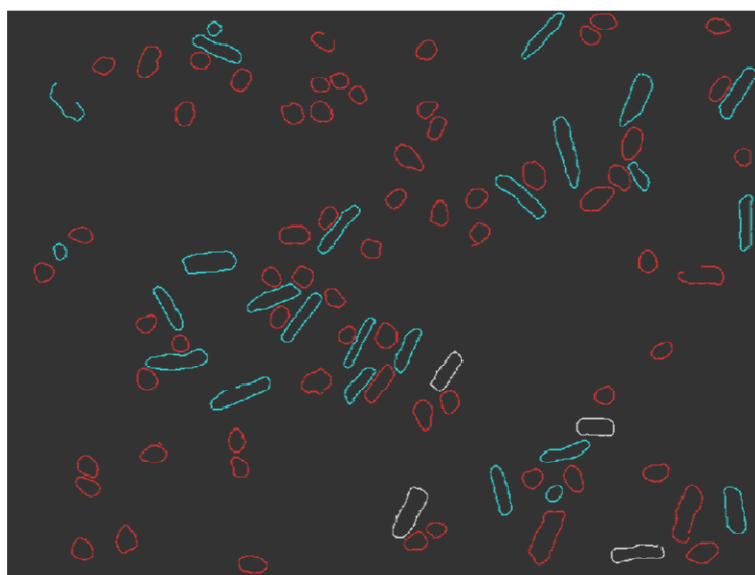
Na výstupu obdržíme dvě hodnoty y_1 a y_2 v rozsahu od 0 do 1. Rozhodovací pravidlo pro určení typu buňky je:

1. výpočet rozdílu $d = y_1 - y_2$
2. určení typu buňky:
 $d > 0$ (na prvním výstupu je větší hodnota) a $|d| \geq 0,3$ (dostatečně velký) $\Rightarrow$ mrtvá buňka,
 $d < 0$ (na druhém výstupu je větší hodnota) a $|d| \geq 0,1 \Rightarrow$ živá buňka.
 $d > 0$ a $|d| \geq 0,3$ nebo $d < 0$ a $|d| < 0,1 \Rightarrow$ síť „pochybuje“.

Signálem pro skončení trénování je dosažení požadovaného počtu epoch, v daném případě 500 (viz obr. 34). Dosažená hodnota chyby je řádově 10^{-3} , což je dostatečně malá hodnota pro úspěšnou klasifikaci. Výsledek rozhodování sítě je znázorněn na obr. 35. Počet správně detekovaných živých buněk je 21, správně detekovaných mrtvých je 62, falešně detekovaných buněk je 8 (4 živých a 4 mrtvých), síť nedokázala rozpoznat 4 buňky (bílé buňky na obr. 35). V procentním obsahu jsou na snímku 23,08 % živých, 68,13 % mrtvých, 8,79%. Chyba je způsobena zařazením do třídy mrtvých buněk živých a naopak a pochybováním sítě tam, kde by mohla přesně klasifikovat.



Obr. 34 Změna střední chyby v průběhu učení dvouvrstvé perceptronové sítě



Obr. 35 Výsledek klasifikace dvouvrstvou perceptronovou sítí

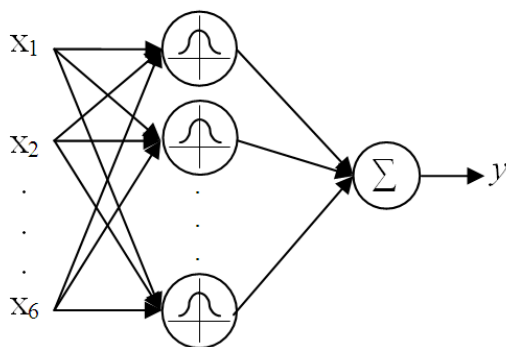
3.6.3 RBF síť

Trénovací množina je tvořena vypočítanými hodnotami příznaků objektu:

- normovaná plocha,
- excentricita,
- pravoúhlost,
- disperze,
- elongace,
- extenze,

a představena jako matice rozměrnosti, $6 \times n$; n je počet trénovacích vzorků. Struktura sítě je zobrazena na obr. 36. Učení se provádí ve třech fázích:

1. určení středů živých a mrtvých buněk c_1 a c_2 shlukováním,
2. určení střední vzdálenosti vektorů trénovací množiny od těchto středů (šířky Gaussovy funkce),
3. výpočet koeficientů ω_{ij} výstupní vrstvy (viz rovnice 45 a 46).



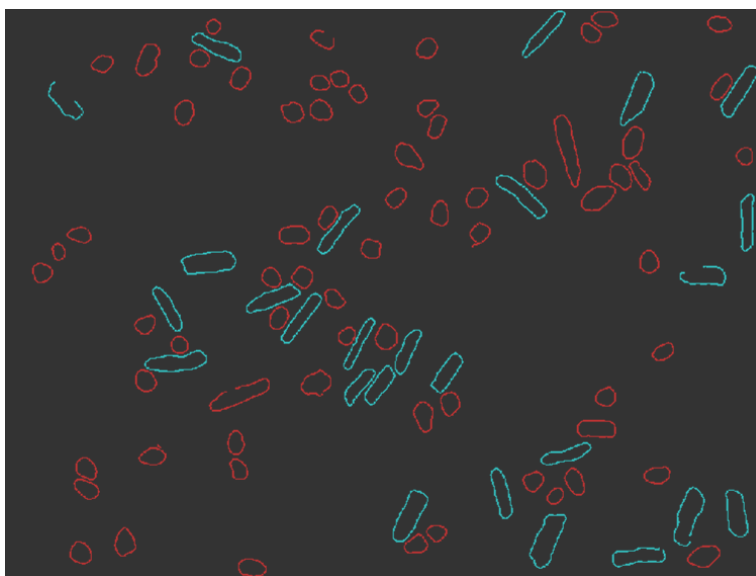
Obr. 36 Struktura RBF sítě

Při dodání testovací množiny do sítě je na výstupu y promítnutá vzdálenost d vstupního vektoru od centra c_1 reprezentujícího mrtvé mušky, hodnota této vzdálenosti určí příslušnost vzoru do třídy mrtvých mušek.

Rozhodovacím pravidlem je:

- $d \leq y < -d \Rightarrow$ živá buňka,
- $-d < y \leq d \Rightarrow$ mrtvá buňka.

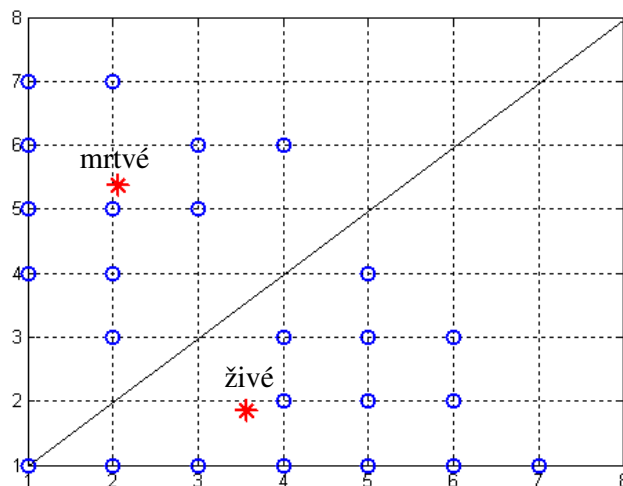
Výsledek klasifikace (viz obr. 37): bylo správně detekováno 26 živých, 66 mrtvých, což je v procentuálním vyjádření je 27,37% a 69,47%. Přičemž tři živé buňky byly zařazeny do mrtvých buněk. Na výsledek klasifikace mají vliv prahové hodnoty rozhodovacího pravidla.



Obr. 37 Výsledek klasifikace RBF sítě

3.6.4 Kohonenova síť

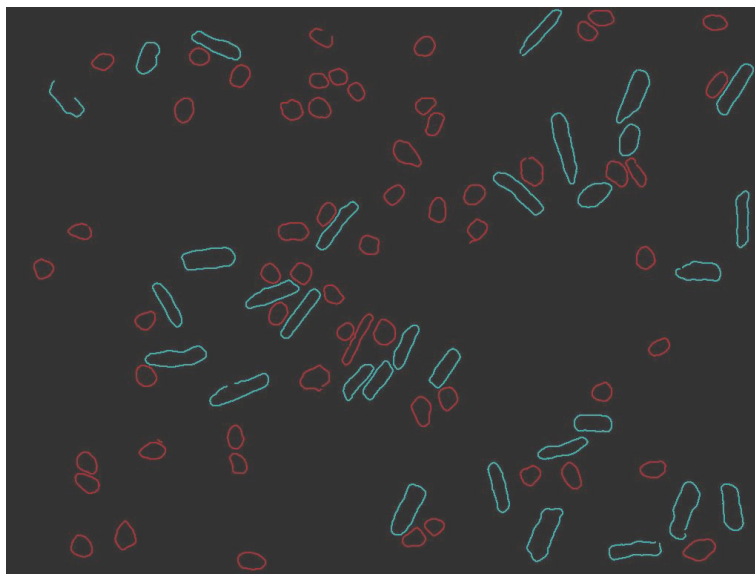
Síť obsahuje 64 neuronů plošně uspořádaných v uzlech matice velikosti 8×8 . Každý neuron má definované váhy ω_{ij} . Při přivádění jednotlivého vzoru na vstup sítě se vyhledá „vítězný“ neuron pomocí nejmenší euklidovské vzdálenosti. Vítězný neuron může být stejný pro různé vzory. Na základě toho, kterou třídu reprezentuje „vítězný“ neuron, určíme typ vybraného objektu (viz obr. 38). Trénovací množina je analogická jako v předchozím klasifikátoru.



Obr. 38 Reprezentace center tříd

Výsledek získaný pomocí Kohonenovy sítě se může lišit při každém spouštění. Na obr. 39 je zobrazen jeden z výsledků. Počet správně detekovaných živých a mrtvých buněk je 29 a 56, což v procentuálním vyjádření je 33,33% a 64,37%, přičemž falešně detekovány byly tři mrtvé a jedna živá buňka. Při porovnání se skutečnými hodnotami můžeme konstatovat docela vysokou úspěšnost tohoto klasifikátoru.

Jedna nevýhoda Kohonenovy sítě spočívá v tom že, máme-li obrázek, který obsahuje jenom buňky jednoho typu, síť se je snaží rozdělit do dvou tříd. V tom to případě nemůžeme očekávat větší úspěšnost než 50 %.



Obr. 39 Výsledek Kohonenové sítě

3.7 Porovnání účinnosti klasifikátorů

Míra účinnosti navržených klasifikátorů byla testována na dvaceti dvou snímcích s různým množstvím kardiomyocitů, úrovní jasu a zaostření, ale při výběru snímků byla věnována pozornost, aby buňky byly izolovány od sebe a vzájemně se nepřekrývaly. Mezní hodnoty rozhodovacích pravidel klasifikátorů jsou stejné pro všechny obrázky. Účinnost budeme hodnotit podle následujících kritérií za předpokladu úspěšného zpracování a segmentace snímku:

1. Sensitivita
2. Specifita
3. Výpočetní rychlost
4. Obtížnost implementace.

Pod pozitivním výsledkem testu budeme rozumět odpověď: „ano, buňka je mrtvá“, pod negativním výsledkem: „ne, buňka je živá“. Pro plné posouzení reálného testu (výpočet sensitivity a specificity) musíme použít následujících pojmů:

TP (true positiv, pravdivě pozitivní) – pozitivní výsledek testu pro buňku, která je mrtvá (korektní klasifikace).

TN (true negative, pravdivě negativní) – negativní výsledek testu pro buňku, která je živá (korektní klasifikace).

FP (false positive, falešně pozitivní) – pozitivní výsledek testu pro buňku, která je živá (nekorektní klasifikace).

FN (false negative, falešně negativní) – negativní výsledek testu pro buňku, která je mrtvá (nekorektní klasifikace).

1. Sensitivita (true-positive rate – TPR, míra pravdivé positivity) je pravděpodobnost, že mrtvá buňka má pozitivní test. Sensitivita se počítá podle vzoru: [11]

$$TPR = \frac{TP}{TP + FN} \quad (52)$$

Výsledky výpočtu jsou shrnuty v tabulce 2 a na jejich základě můžeme konstatovat, že nejúspěšnějšími detektory mrtvých buněk jsou RBF síť a klasifikátor založený na příznakovém popisu. Sensitivita Kohonenovy sítě má také vysokou hodnotu, ale tato hodnota se může měnit kvůli náhodnému nastavení vah neuronů.

2. Specificita (true-negative rate – TNR, míra pravdivě negativity) je pravděpodobnost, že živá buňka má negativní test a počítá se podle vzoru: [11]

$$TNR = \frac{TN}{TN + FP} \quad (53)$$

Nejúčinnějším detektorem živých buněk podle tabulky je dvouvrstvá perceptronová síť, ale nemůžeme tento výsledek uvažovat jako spolehlivý, protože ve 108 případech síť nebyla schopna přesně detekovat, to znamená „pochybovala“. V důsledku toho, mezi celkovým množstvím detekovaných živých buněk je menší hodnota falešně pozitivních výsledků.

Tabulka 2. Úspěšnost klasifikátorů

Typ klasifikátoru	TP	TN	FP	FN	Sensitivita, %	Specificita, %	Střední hodnota
Klasifikátor založený na příznakovém popisu	666	235	27	59	91,86	89,69	90,76
Jednovrstvá síť perceptronů	665	206	32	86	88,54	86,55	87,54
Dvouvrstvá síť perceptronů	575	201	12	91	86	94,36	90,18
RBF síť	708	173	92	28	96,19	65,28	80,73
Kohonenova síť	664	212	51	60	91,71	80,60	86,15

3. Z hlediska rychlosti učení (doba trvání adaptační fáze) a doby aktivační fáze při podání nového snímku sítě, jsou nejvhodnějšími klasifikátory založené na příznakovém popisu a Kohonenova síť, protože nepotřebují žádnou adaptaci vah. Za nimi následuje RBF síť, jelikož doba jejího učení je krátká. Nejdelší dobu učení mají perceptronové sítě, to je způsobeno velikostí trénovací množiny.

4. S ohledem na obtížnost implementace jako nejvhodnější jeví klasifikátor založený na příznakovém popisu, který spočívá pouze ve vyhodnocení excentricity a porovnání s mezní hodnotou. Kohonenova a RBF síť mají ve svém algoritmu shlukování, což také zvyšuje obtížnost jejich implementace. Nejsložitější algoritmy implementace mají sítě, které mají složitější trénovací množinu a to jsou v našem případě perceptronové sítě.

Při výběru vhodného klasifikátoru musíme vycházet z toho, jaký je stanoven cíl. Jestli je nejdůležitějším parametrem počet mrtvých buněk, musíme vybrat test s větší hodnotou sensitivity, jestliže je důležitější, kolik je na obrázku živých buněk, musíme zvolit více specifický klasifikátor. Pokud měříme jich procentní poměr, při výběru vycházíme

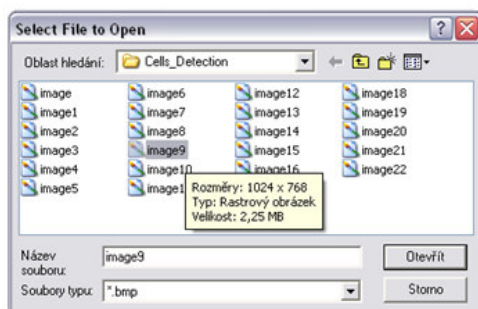
z maximální střední hodnoty sensitivity a specificity. Při kladení zvláštních požadavků na vysokou rychlost nebo na snadnější implementaci, musíme vybírat klasifikátory mezi těmi, které nepotřebují učení nebo mají jednodušší, ale dostačující trénovací množinu.

4. Realizace v prostředí MATLAB

V prostředí MATLAB bylo vyvinuto grafické uživatelské rozhraní „Cells_Detection“ pro zpracování snímků a klasifikaci izolovaných kardiomyocytů [1]. Toto rozhraní se ovládá především pomocí menu, jednotlivé položky kterého odpovídají etapám zpracování a klasifikace obrazu.

1. Nejprve uživatel vybere obrázek, který chce zpracovávat v původním formátu .bmp.

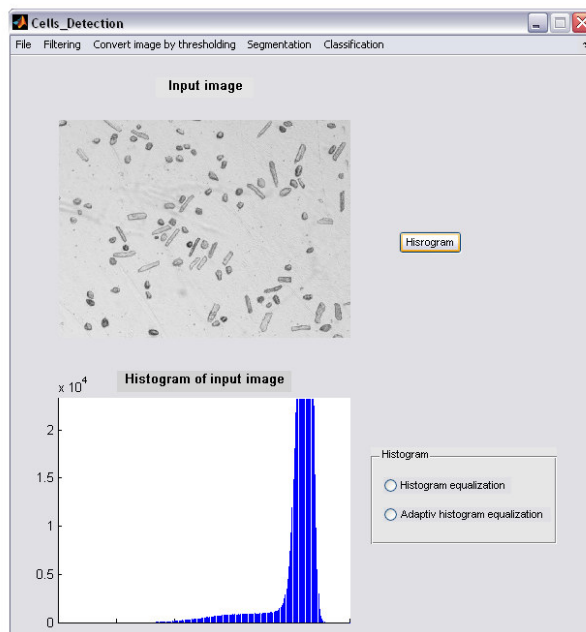
- File - obsahuje položky určené především pro otevření obrázku:
 - Open – zobrazí dialog pro otevření snímku (obr. 40)
 - Exit – ukončí aplikaci



Obr. 40 Dialog pro otevření obrázku

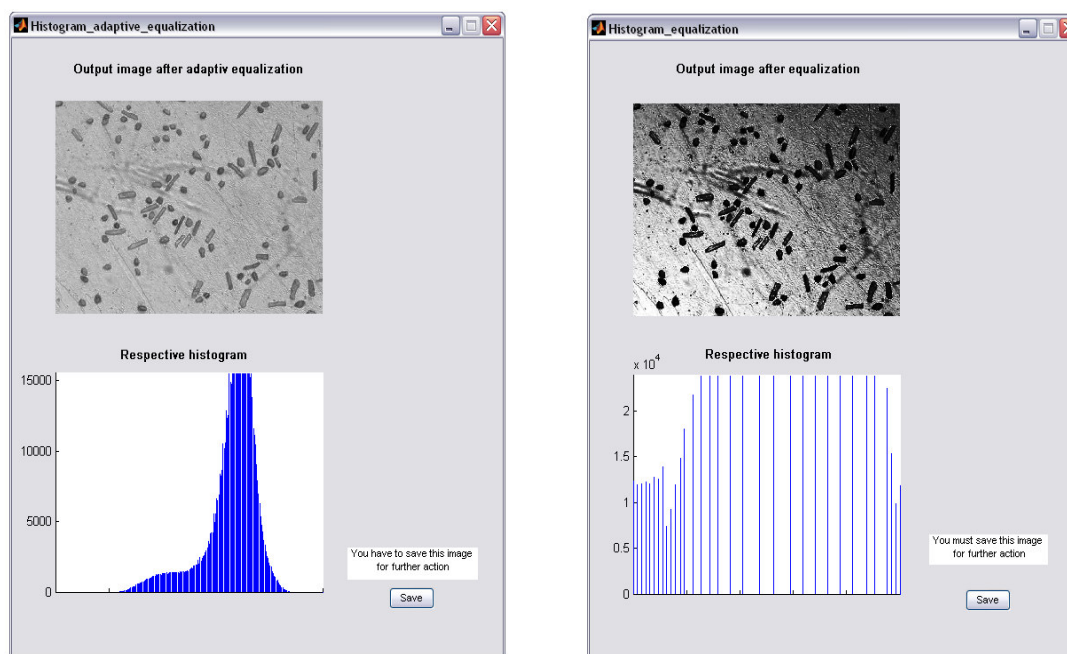
Vybraný barevný snímek je automaticky převáděn na šedotonový, protože filtry informace o jednotlivých barevných složkách nevyužívají a práce se snímky ve stupních šedi je jednodušší: algoritmy, které snímek zpracovávají, vyžadují méně strojového času, než kdyby se zpracovávala každá barevná složka zvlášť a interpretace výsledků by byla komplikovanější.

Původní obrázek se zobrazí v samostatném okně (obr. 41), stisknutím tlačítka „Histogram“ je možné zobrazení histogramu obrázku.



Obr. 41 Dialog pro zobrazení obrázku a příslušného histogramu

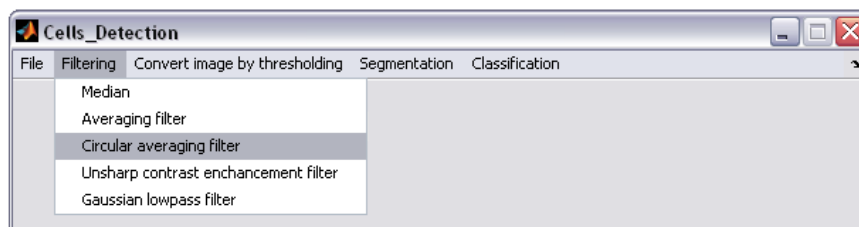
2. Jako druhý krok můžeme aplikovat ekvalizaci nebo adaptivní ekvalizaci histogramu (obr. 42). Po porovnání obou výsledků, musíme zachovat ten obrázek, se kterým chceme dále pracovat (stisknutím tlačítka „Save“), jinak budeme v dalším kroku zpracovávat původní snímek.



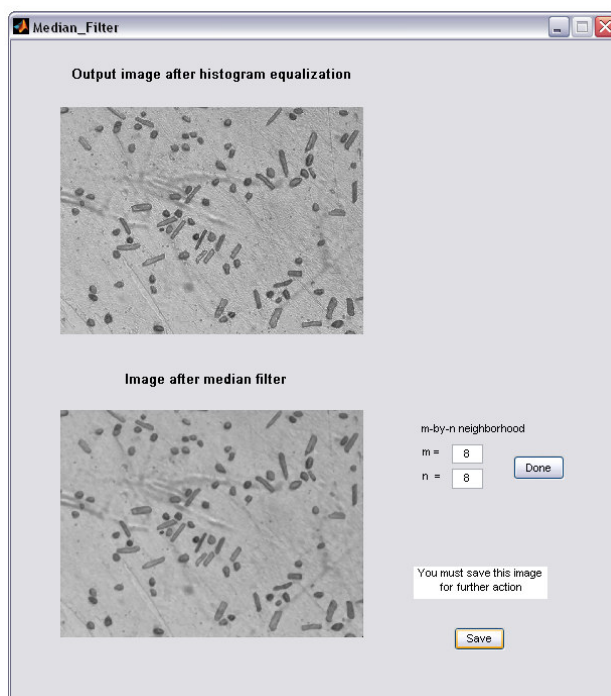
Obr. 42 Dialog pro porovnání operací transformace histogramu

3. Dalším krokem je filtrace. Na výběr je pět typů filtru (obr. 43), které uživatel může zvolit na základě vlastností snímku nebo stanoveného účelu. V dialogovém okně filtrace je možno zadat velikost okolí případně jiné parametry filtrů nebo použít defaultní (automatická) nastavení (obr. 44). Při spokojenosti s výsledkem je třeba obrázek zachovat pro další zpracování (stejně jako v minulém kroku pomocí "Save").

- Filtering
 - Median – filtrace medianovým filtrem
 - Averaging filter – filtrace průměrovacím filtrem
 - Circular filter – filtrace průměrovacím filtrem s kruhovou maskou
 - Unsharp contrast enhancement filter – filtrace ostřicím filtrem
 - Gaussian lowpass filter – filtrace Gaussovou dolní propustí



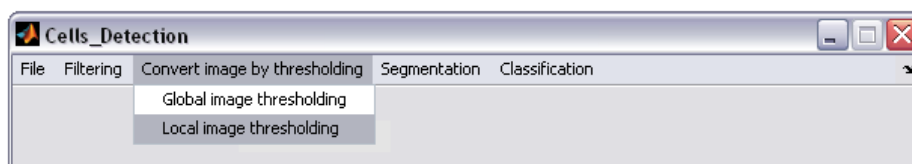
Obr. 43 Dialog pro výběr typu filtru



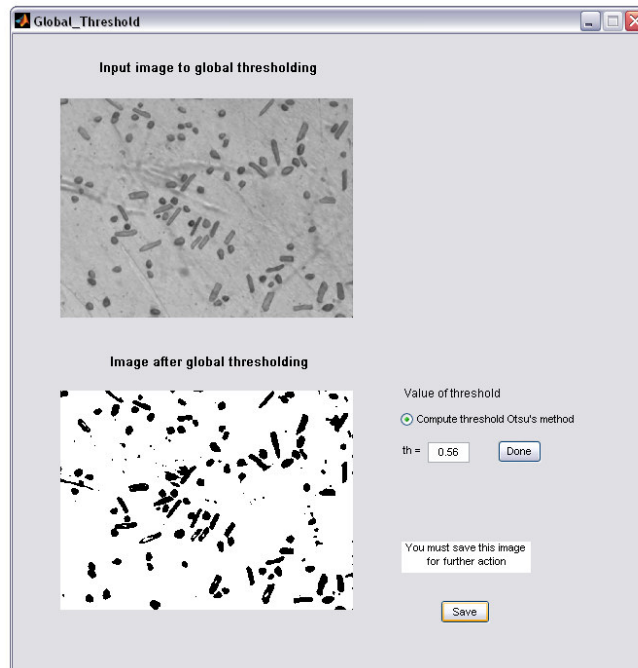
Obr. 44 Dialog pro nastavení parametrů a implementaci filtru

4. Dále může uživatel použít metody globálního a lokálního prahování (obr. 45). Hodnotu globálního prahu může uživatel nastavit sám nebo vypočítat metodou Otsu (obr. 46). Při výběru metody lokálního prahování se obraz automaticky rozdělí na sedm částí o stejné velikosti 256×256 a pro každou část bude prahová hodnota vypočítána metodou Otsu (viz obr. 17).

- Convert image by thresholding
 - Global image thresholding – metoda globálního prahování
 - Local image thresholding – metoda lokálního prahování



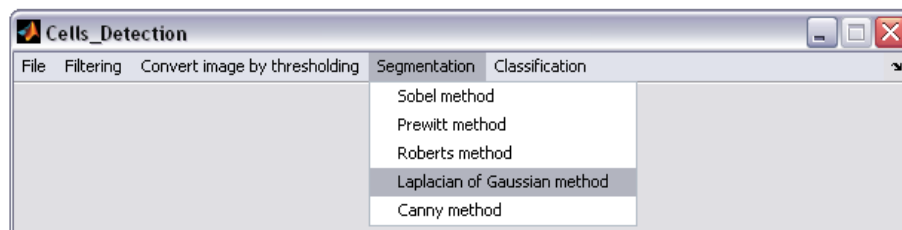
Obr. 45 Dialog pro výběr metody prahování



Obr. 46 Dialogové okno pro nastavení prahové hodnoty a implementaci vybrané metody

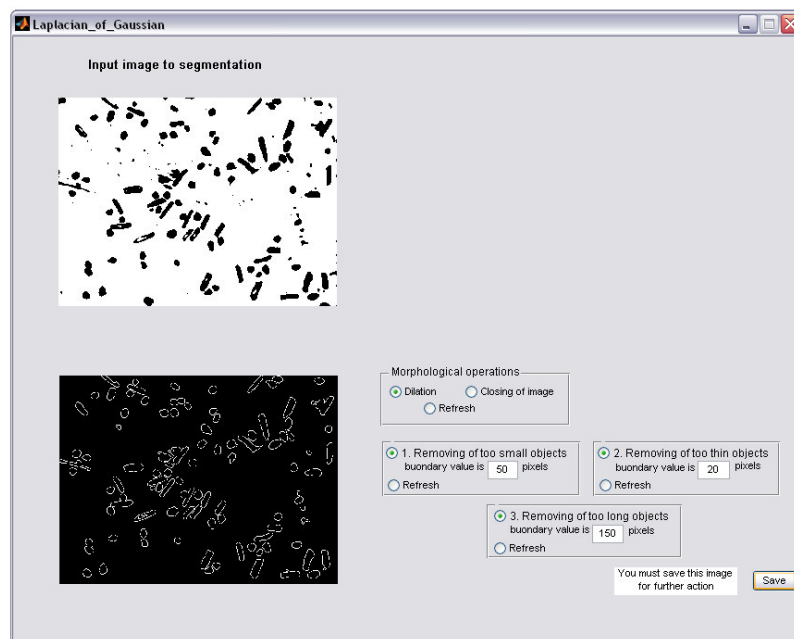
5. Finální etapou zpracování je segmentace. Potřebný typ segmentace uživatel může zvolit ve hlavním menu (obr. 47).

- Segmentation
 - Sobel method – Sobelův detektor hran
 - Prewitt method – Prewittův detektor hran
 - Roberts method – Robertsův detektor hran
 - Laplacian of Gaussian method – Laplacian po Gaussove filtraci
 - Canny method – Cannyho detektor hran



Obr. 47. Dialog pro výběr způsobu segmentace

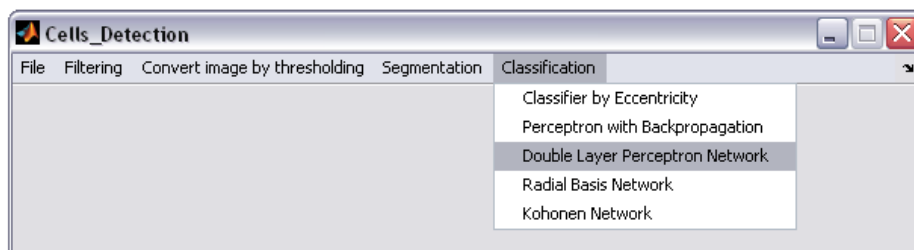
V dialogovém okně segmentace můžeme provést morfologické operace: binární dilatace a binární zavření, za účelem spojování hranic objektu a odstranit nežádoucí objekty nastavením hraničních mezí pro velikost parametrů (obr. 48).



Obr. 48 Okno pro nastavení parametrů a realizaci zvoleného typu segmentace

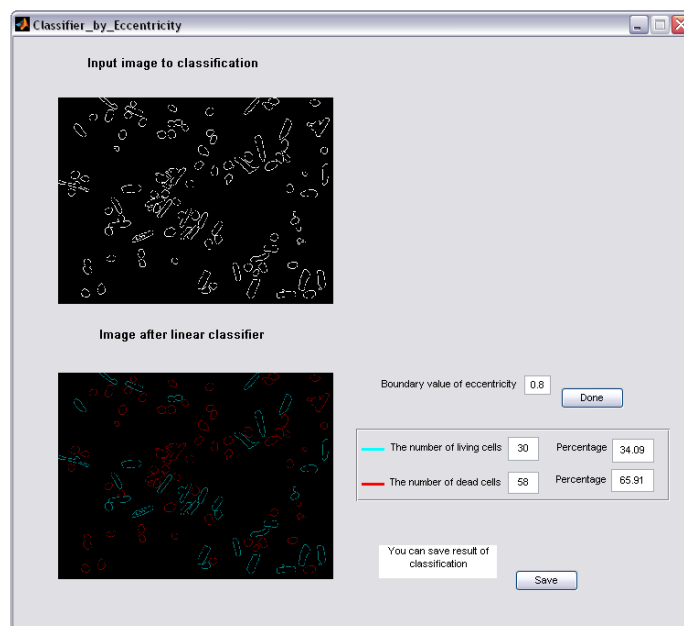
6. Způsob klasifikace můžeme vybrat v položce hlavního menu (obr. 49):

- Classification
 - Classifier by Eccentricity – klasifikátor založený na příznakovém popisu,
 - Perceptron with Backpropagation – perceptronová síť s algoritmem učení backpropagation
 - Double Layer Perceptron Network – dvouvrstvá perceptronová síť s algoritmem učení backpropagation,
 - Radial Basis Network – RBF síť,
 - Kohonen Network – Kohonenova síť.



Obr. 49 Dialog pro výběr typu klasifikátoru

Po zvolení klasifikátoru zobrazí další okno s výsledkem v grafické a číselné podobě. Rozpoznané živé a mrtvé buňky jsou zobrazeny odlišnou barvou, je uveden jejich počet a procentuální poměr (obr. 50).



Obr. 50 Dialog s výsledkem klasifikace

5. Závěr

Cílem předložené diplomové práce byly, kromě prostudování metod zpracování obrazů (filtrace, segmentace, morfologické operace), zejména návrh a realizace těchto metod v prostředí MATLAB. Dalším cílem práce byl návrh a realizace různých klasifikátorů a vyhodnocení úspěšností klasifikace.

Vhodným postupem pro zpracování bylo zvoleno:

- adaptivní ekvalizace histogramu;
- mediánová filtrace (rozměr masky 7×7);
- segmentace na základě průchodu druhé derivace obrazové funkce nulou po Gaussově filtraci;
- dilatace obrazu za účelem spojování hran.

Pro určení množství mrtvých a živých buněk byly navrženy čtyři typy klasifikátorů: klasifikátor založený na příznakovém popisu objektu, jednovrstvá a dvouvrstvá perceptronová síť, RBF síť a Kohonenova síť. Rozhodujícím příznakem pro první typ klasifikátoru byla vybrána excentricita objektu. Vstupními daty pro perceptronové sítě jsou binární obrazová data. Vstupními daty pro RBF a Kohonenovu síť jsou jednodušší matice, která obsahuje hodnoty šesti příznaku objektů (viz kapitola 3.6.3).

Účinnost klasifikátorů byla testována na dvaceti dvou snímcích s různou úrovní jasu a zaostření (viz kapitola 3.7). Na základě obdržení hodnot sensitivity a specificity můžeme konstatovat, že složitost klasifikátoru neovlivní žádným způsobem jejich účinnost. Funkčnost klasifikátorů realizovaných v tomto projektu záleží na charakteru zpracovaného obrazu, segmentaci a na použitých způsobech předzpracování. Proto pro zlepšení výsledků klasifikace je třeba věnovat pozornost co nejvyšší kvalitě snímání obrazu (co nejmenší ztráta informace) a algoritmu zpracování.

Aplikace navržená pro kvalitní analýzu mikroskopického a vytvořená v programovém prostředí MATLAB by měla nalézt uplatnění zejména při hodnocení kvality izolace srdečních buněk.

Literatura

- [1] ANUFRIEV, I.: *Priloženija s GUI i diskriptornaja grafika*. Sankt – Peterburg, 2008 [cit. 2009-04-23]. Elektronická skripta. Dostupné z WWW: <<http://matlab.exponenta.ru/gui/index.php>>
- [2] ČECH, S., HORKÝ, D.: *Přehled obecné histologie*. 1. vyd., Brno: Masarykova univerzita v Brně, 2005. 140 s. ISBN 80-210-3854-3.
- [3] GENČÚR, M.: *Nástroje na zpracování fotografovaného textu*. Bakalářská práce. Brno, 2007.
- [4] HONZÍKOVÁ, N.: *Biologie člověka*. Skripta FEI VUT v Brně, 2003.
- [5] CHALUPKA, J.: *Binární matematická morfologie*. [cit. 2009-05-10]. Elektronická skripta. Dostupné z WWW: <<http://itakura.kes.tul.cz/pv/t08/P08.pdf>>.
- [6] JAN, J.: *Číslíková filtrace, analýza a restaurace signálů*. VUTIUM, Brno 2002.
- [7] JAN, J.: *Analýza signálů a obrazu*. Elektronická skripta, 2007.
- [8] KYSELÝ, F.: *Metody maskování chyb pro Motion JPEG2000*. Brno 2007 [cit. 2008-04-12]. Dostupné z WWW: <www.elektrorevue.cz/cz/download/metody-maskovani-chyb-pro-motion-jpeg2000/>. ISSN 1213-1539.
- [9] LINKA, A., VOLF, P., KOŠEK, M.: *Zpracování obrazu a jeho statistická analýza*. [cit. 2009-05-15]. Dostupný z WWW: <<http://e-learning.tul.cz/cgi-bin/elearning/elearning.fcgi?podsekce=34&stranka=publ>>.
- [10] POTEMKIN, V.: *Nejronnyje seti*. [cit. 2009-03-15]. Dostupné z WWW: <<http://matlab.exponenta.ru/neuralnetwork/index.php>>.
- [11] PROVAZNIK, I.: *Počítačová podpora lékařské diagnostiky*. Elektronická skripta, 2008.
- [12] SOBOTKA, Z., SOBOTKA, M.: *Počítačová analýza a rozpoznávání obrazu*. Praha: Dům techniky ČSVTS, 1990. 277 s. ISBN 80-02-00739-5.
- [13] ŠIMA, J., NERUDA, R.: *Teoretické otázky neuronových sítí*. Elektronická skripta.
- [14] ŠONKA, M., HLAVÁČ, V.: *Počítačové vidění*. Praha: Grada, 1992. 252 s. ISBN 80-85424-67-3.
- [15] ŠTRAFELDA, P.: *Detekce objektů*. Diplomová práce. Brno, 2005.
- [16] VESELOVSKY, M.: *Neuronové sítě*. Elektronická skripta. Dostupné z WWW: <<http://avari.cz/uir/index.php>>
- [17] ZILNIKA, I.: *Umělá inteligence*. Elektronická skripta. Zlín, 1997. Dostupné z <<http://vonmarcus.ic.cz/uploads/Staniik/MUI.pdf>>

- [18] ŽELEZNÝ, M.: *Bodové jasové transformace* [cit 2008-03-27], Dostupné z WWW: <<http://www.kky.zcu.cz/cs/courses/zdo/lesson2>>.
- [19] ŽURAVLEV, I.: *Kratky kurs teorii obrabotki izobraženija*. Kiev, 2008. Dostupné z WWW: <<http://matlab.exponenta.ru/imageprocess/index.php>>.

Seznam příloh

Příloha A. Funkce pro adaptaci vah zpětným šířením chyby (backpropagation)	- 56 -
Příloha B. Funkce pro učení RBF sítě	- 57 -
Příloha C. Algoritmus Kohonenove sítě	- 58 -

Přílohy

Příloha A. Funkce pro adaptaci vah zpětným šířením chyby (backpropagation)

```
%algoritmus backpropagation
% cells - trenovací množina, d-vektor etalonu
function [w, error] = Weigth_BP(cells, d);
[s q] = size(cells);
k = 1;
e = 0.1;
w(1:s,1) = 0.5 - rand(s,1); %pocatecni nahodne inicializace vah

%aktivacni faze
for j = 1:q
    y(j) = 1/(1+exp(-(sum(w(1:s,k).*cells(1:s,j)))));
    error(j) = (0.5)*(d(j)-y(j))^2;
end
error_sum(1) = sum(error); %vypocet celkove chyby

%stridani adaptacni a aktivacni faze pokud celkova chyba je vetsi
nez 0.0001
while error_sum(k) > 0.0001
    for j = 1:q
        w(1:s,1) = w(1:s,1)-e*(y(j)-d(j))*cells(1:s,j);
        y(j) = 1/(1+exp(-(sum(w(1:s,1).*cells(1:s,j)))));
        error(j) = (0.5)*(d(j)-y(j))^2;
    end;
    error_sum(k+1) = sum(error);
    k = k+1;
end;
```

Příloha B. Funkce pro učení RBF sítě

```
%uceni RBF site
%P - trenovaci mnozina
function w = Weigth_RBF(P)
P_p = P';
[center, U, obj_fcn] = fcm(P_p,2); %nalezeni centru klasteru
center = center';
ed = dist(P_p,center); %vypocet euklidovske vydalenosti
sr = mean2(ed);
fi = zeros(s,2*t);
r1 = zeros(s,1); r2 = zeros(s,1);
pr1 = zeros(s,1); pr2 = zeros(s,1);
%vystup RBF jednotek
for j = 1:t
    r1(1:s,1) = P(1:s,2*j-1)-center(1:s,1)/sr;
    r2(1:s,1) = P(1:s,2*j)-center(1:s,2)/sr;
    pr1(1:s,1) = r1(1:s,1).*r1(1:s,1);
    pr2(1:s,1) = r2(1:s,1).*r2(1:s,1);
    fi(1:s,2*j-1) = exp(-(pr1(1:s,1)));
    fi(1:s,2*j) = exp(-(pr2(1:s,1)));
end
[s q] = size(fi);
%pocatecni inicializace vah vystupni vrstvy
w(1:s,1) = 0.5 - rand(s,1);
%aktivacni faze
for j = 1:q
    y(j) = sum(w(1:s,1).*fi(1:s,j));
end
%adaptace vah minimalizaci kvadratu odchylek
wpr = (fi'*fi)^(-1)*fi';
w = wpr'*y';
%zapis promennych do .txt souboru
dlmwrite('weigth_RBF.txt', w);
dlmwrite('center.txt',c);
dlmwrite('sr.txt', sr);
```


Příloha C. Algoritmus Kohonenove sítě

```
%Algoritmus Kohonenove site
%P - vstupni mnozina, dead - pocet mrtvych bunek, live - pocet zivych
function [dead, live]= Kohonen(P);
%plosne usporadani neurunu a nastaveni vah
n = 8;
[s t] = size(P);
r = cell(n, n);
A = ones(1, t);
Pc = mat2cell(P,[s], A);
w1 = rand(s*n, n);
wc = mat2cell(w1, [6 6 6 6 6 6 6 6], [1 1 1 1 1 1 1 1]);
nul = zeros(s*(3*n-2), (3*n-2));
rn = zeros(s*(3*n-2), (3*n-2));
r = mat2cell(rn, [6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6],...
    [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]);
w = mat2cell(nul,[6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6],...
    [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]);
wr = repmat(r, 1, 1);
rt = repmat(r, 1, 1);
ch = 1;
for i = n:2*n-1
    for j = n:2*n-1
        w{i,j}=wc{i-n+1, j-n+1};
    end
end
alfa = 0.5;
%aktivacni faze
for k = 1:t
    beta = 0.5;
    for i = 1:n
        for j = 1:n
            r{n+i-1,n+j-1} = Pc{k}-w{n+i-1, n+j-1};
            rum{i, j} = r{n+i-1,n+j-1}.*r{n+i-1,n+j-1};
            d(i, j) = sum(rum{i, j});
        end;
    end;
    %nalezeni minimalni euklidovske vzdalenosti
    dm = min(min(d));
    [x, y] = find(d == dm);
    xb = x + 7; yb = y + 7;
    w{xb, yb} = w{xb, yb} + alfa*(Pc{k} - w{xb,yb});
    ch = 1
    %adaptace vah - posunuti vitezneho neuronu
    while ch~=0
        for l = 1:n-1
            for p = 1:2*l+1
                wr{xb-l+p-1,yb-l}=w{xb-l+p-1,yb-l}+beta*r{xb-l+p-1,yb-l};
                wr{xb-l+p-1,yb+l}=w{xb-l+p-1,yb+l}+beta*r{xb-l+p-1,yb+l};
                wr{xb+l,yb-l+p-1}=w{xb+l,yb-l+p-1}+beta*r{xb+l,yb-l+p-1};
                wr{xb-l,yb-l+p-1}=w{xb-l,yb-l+p-1}+beta*r{xb-l,yb-l+p-1};
                beta = (0.5)*beta;
            end
        end
        for i = 1:(3*n-2)
            for j = 1:(3*n-2)
                ch = sum(wr{i, j} - w{i, j});
            end
        end
    end
end
```

```

        end
        w = wr;
    end
    alfa = 0.5*alfa
end
%nalezni viteznych neuronu
for k = 1:t
    for i = 1:n
        for j = 1:n
            rt{n+i-1,n+j-1} = Pc{k}-wr{n+i-1, n+j-1};
            rumt{i,j} = rt{n+i-1,n+j-1}.*rt{n+i-1,n+j-1};
            dt(i,j) = sum(rumt{i,j});
        end;
    end;
    dmt(k) = min(min(dt));
    [x(k), y(k)] = find(dt == dmt(k));
end
%nalezni stredu klasteru
xy = cat(1, x, y);
xyt = xy';
[kohonen, U, obj_fcn] = fcm(xyt, 2);
%urceni vydalenosti vitezneho neuronu od stredu klasteru
for j=1:t
    r1(j) = (xy(1,j)-kohonen(1,1))^2 + (xy(2,j)-kohonen(1,2))^2;
    r2(j) = (xy(1,j)-kohonen(2,1))^2 + (xy(2,j)-kohonen(2,2))^2;
    if r1(j) > r2(j)
        live(j)=j;
    else
        dead(j)=j;
    end
end
end
end

```