

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Automatická detekce rakoviny

Místo této strany bude
zadání práce.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 5. května 2016

Michal Medek

Abstract

This bachelor thesis deals with an automatic cancer detection using image descriptors. The work briefly summarizes available image databases. It analyses the LBP, LDP and POEM image descriptors that are used for feature extraction, and the SVM and MLP classifiers necessary for cancer identification. Within the thesis, user manual and result analysis of the particular methods are provided.

Abstrakt

Tato bakalářská práce se zabývá automatickou detekcí rakoviny s použitím obrazových deskriptorů. V práci jsou stručně shrnuty dostupné databáze se snímků. Následně jsou rozebrány obrazové deskriptory LBP, LDP a POEM, které slouží pro extrakci příznaku a poté klasifikátory SVM a MLP pro identifikaci rakoviny. Práce dále popisuje ovládání vzniklé aplikace a rozebírá dosažené výsledky.

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce doc. Ing. Pavlu Královi, Ph.D. za jeho vedení, trpělivost a cenné rady v průběhu vypracování celé této práce.

Obsah

1	Úvod	8
2	Mamografický screening	9
2.1	Hledané abnormality	9
3	Computer aided detection systems (CAD)	11
3.1	Detekce podezřelé tkáně	11
3.1.1	Detekce	11
3.1.2	Klasifikace	12
3.2	Úspěšnost jiných aplikací	13
3.3	Databáze mamogramů	13
3.3.1	Digital Database for Screening Mammography (DDSM)	13
3.3.2	MiniMammographic Database (MIAS)	14
3.3.3	Mammographic patches	14
4	Relevantní metody z oblasti počítačového vidění	15
4.1	Předzpracování	15
4.2	Lokální obrazové deskriptory	16
4.2.1	Local Binary Pattern (LBP)	16
4.2.2	Local Derivate Pattern (LDP)	17
4.2.3	Patterns of Oriented Edge Magnitudes (POEM) . . .	18
5	Klasifikace	20
5.1	Metoda podpůrných vektorů	20
5.2	Vícevrstvý perceptron	22
6	Implementace	24
6.1	Vybrané metody	24
6.2	Použité knihovny	24
6.2.1	OpenCV	24
6.2.2	Instalace OpenCV na OS Ubuntu 14.04 LTS	25
6.2.3	Boost C++ libraries	26
6.2.4	Instalace Boost C++ na OS Ubuntu 14.04 LTS . . .	26
6.3	Knihovna pro rozpoznávání obličejů	26
6.4	Navržený systém pro automatickou identifikaci rakoviny . . .	27
6.4.1	Architektura programu	27

6.4.2	Nahrávání obrázků	28
6.4.3	Předzpracování	29
6.4.4	Extrakce příznaku	29
6.4.5	Klasifikace	30
6.4.6	Konfigurační soubor	30
6.4.7	Trénovací a testovací soubory	33
7	Provedené experimenty	34
7.1	Nastavení parametrů obrazových deskriptorů	35
7.1.1	Nastavení parametrů LBP	36
7.1.2	Nastavení parametrů LDP	38
7.2	Optimalizace klasifikátorů	39
7.2.1	Nastavení parametrů MLP	39
7.2.2	Nastavení parametrů SVM	44
7.3	Experimentální použití k-means	46
7.4	Shrnutí	47
8	Další možná rozšíření	49
9	Závěr	52
	Literatura	53
	Přílohy	55
A	Obsah přiloženého CD	56

1 Úvod

Cílem této práce je vytvořit systém, který bude s co největší úspěšností rozpoznávat rakovinu na snímcích pořízených lékařskými diagnostickými přístroji, které pracují na bázi rentgenového záření. My budeme pracovat se snímky z mamografu, přístroje, který je určen pro diagnostiku nádorových onemocnění prsu.

Právě nádorové onemocnění prsu je u žen nejběžnější formou rakoviny po rakovině plic a druhou nejčastější formou rakoviny způsobující úmrtí [11]. Zasažena je v průměru jedna z osmi až dvanácti žen [8]. Odborníci navíc předpovídají, že v následujících letech bude počet žen zasažených tímto onemocněním a dalšími civilizačními chorobami stoupat.

Mamografie jako metoda diagnostiky karcinomu prsu je jednoznačně nejrozšířenější metodou sloužící k diagnostice a to zejména proto, že umožňuje včasné rozpoznání ložiska karcinomu. Platí totiž, že čím menší je v době nálezu nádor, tím je větší pravděpodobnost úspěšné léčby. Samotné mamografické snímky může klasifikovat pouze zkušený radiolog, který už se setkal s mnoha snímky. Právě tuto úlohu budeme v práci řešit pomocí metod strojového učení. Vzniknout by tak měl funkční prototyp programu, který by mohl svým rozhodováním pomáhat radiologovi se správným určením, zda je na konkrétním snímku rakovina či ne.

Vize je taková, že při vhodném natrénování větší databáze několika skupin snímků budeme následně nově rozpoznávané snímky do těchto skupin řadit. Na základě této myšlenky vznikl dříve různý software. My se pokusíme pro tuto úlohu využít lokálních obrazových deskriptorů jako například v [8].

2 Mamografický screening

Jedná se o nejčastěji používanou metodu pro diagnostiku karcinomu prsu. Pro získávání snímků používáme mamograf, což je přístroj, který funguje na základě speciálního měkkého rentgenového záření, které umožňuje zobrazení i méně kontrastní komponenty prsu. Obvykle se provádí screening obou prsou, přičemž je v průběhu snímání prso mírně stlačeno z důvodu zdůraznění kontrastu a zamezení pohybového rozostření [14]. Snímkování provádíme ze dvou směrů - od hlavy dolů (kраниokaudální) a šikmo zhruba pod 45 stupni (mediolaterální). Následně jsou snímky, neboli mamogramy, důkladně prozkoumány zkušeným radiologem, který provede jejich vyhodnocení.

Mnoho rozsáhlých screeningových studií uvedlo, že ve věkové kategorii 50 až 65 let přináší zavedení mamografického snímkování s intervalem jeden až dva roky po dobu nejméně sedmi let snížení úmrtnosti z důvodu karcinomu až o 30 %. Přínos je zřejmý, i když ne tak znatelný, i pro věkovou skupinu 40 až 50 let. V České republice je preventivní mamografické vyšetření doporučováno po dosažení 45 let.

2.1 Hledané abnormality

Mezi základní znamení, že něco není v pořádku, patří nejčastěji výraznější oválné, kulaté, lalůčkovité nebo nepravidelné útvary kontrastně vystupující nad okolní tkáň prsu. Co se týká jejich hranic, tak pokud jsou nevýrazné a neurčité, je větší pravděpodobnost zhoubnosti než hranice výrazně vymezené. Tato skutečnost souvisí s faktem, že rozrůstající se nádor většinou své pevné hranice postupně ztrácí.

Dalším identifikátorem možného rakovinotvorného bujení jsou malé kousky vápníku vyskytující se v prsou, zvané kalcifikace. Na mamogramu jsou vidět jako drobné světlé tečky různých velikostí a tvarů. Rozlišovacím znakem je tvar, počet, velikost a rozprostření po mamogramu. Hlavním identifikátorem je, že jsou obvykle velmi malé 0.1 až 1 mm a v průměru okolo 0.3 mm. Proto mohou být snadno přehlédnuté z důvodu překryvu tkání. Pokud je pozadím právě hutná tkáň je velmi obtížné tyto částičky detekovat. Někdy jsou také

odstraněny jako šum, pokud se málo liší od podkladu.



Obrázek 2.1: Ukázka označeného nádoru na mamogramu z databáze DDSM

3 Computer aided detection systems (CAD)

CAD systémy byly vyvinuty na pomoc radiologům při detekci rakoviny prsu ze snímků získaných například z mamografu. Tyto systémy jsou schopny odhalit abnormality v mamogramu, které často nemusí být zachytitelné lidským okem. Fungují jako tzv. druhý čtenář, přičemž finální rozhodnutí je stále na radiologovi, který se analýze věnuje.

Dalším důvodem pro zavedení CAD systémů je fakt, že z tisíce snímků analyzovaných radiologem jsou karcinogenní pouze tři až čtyři. Tato abnormalita může být proto snadno opomenuta. Důvodem neúspěšného rozhodnutí radiologa bývá přehlédnutí nebezpečného snímku v 10 % až 30 % případů [11].

3.1 Detekce podezřelé tkáně

Radiologové charakterizují nádor dle tvaru a rozpětí. Podezřelá tkáň (spiculated mass) je část tkáně s hroty nebo body na povrchu, která ale není diagnostikována jako zhoubná, ale pouze podezřelá. Je charakteristická čárami táhnoucími se po okrajích této tkáně.

Většina programů na určení podezřelých tkání je složena ze dvou částí:

1. detekce podezřelých tkání,
2. klasifikace.

3.1.1 Detekce

První fáze je připravena tak, aby zachytila všechny podezřelé snímky i za cenu velkého množství falešně pozitivních nálezů – tedy nálezů, které označíme mylně jako rakovinu. Automaticky počítáme s tím, že klasifikace bude pracovat právě na selekci těchto falešně pozitivních nálezů.

Rozlišujeme dva základní algoritmy pro detekci podezřelých oblastí, Pixel-based a region based detection.

Pixel-based detection V těchto metodách jsou příznaky získávány ze sousedství každého pixelu a následně klasifikovány. Toto může být provedeno za pomoci jednoduchého prahování či složitějších metod. Následně jsou podezřelé pixely seskupovány do spolu propojených regionálních skupin. V zásadě pouze zdůrazníme podezřelé oblasti, klasifikace probíhá v další fázi. Mezi tyto metody patří například algoritmus ALOE [4], který je založený na lokální orientaci hran. Jedná se o využití informace, že v normálním případě jsou struktury vedeny od bradavky k hrudníku, ale podezřelé oblasti mohou tento trend měnit.

Nad touto skupinou vzniklo několik modifikací. Příkladem je algoritmus, který se snažil detekovat hvězdicovité útvary (víme, že nejčastějším tvarem zhoubného nádoru je právě hvězdicovitý tvar). Pro klasifikaci se následně používají velmi jednoduché algoritmy. Hlavní výhodou je počet vzorků pro trénování, který není příliš vysoký.

Region based detection Nejdříve extrahujeme podezřelé části mamogramů tzv. ROI pomocí segmentace nebo filtrování (DWCE [9], DoG [10]). Příznaky jsou následně vybrány pro každý region zvlášť a klasifikovány. Dané příznaky jsou navrženy tak, aby zaznamenaly důležité informace jako tvar nebo texturu. Výhodou je znalost prostorové informace a menší výpočetní náročnost [11].

Nevýhodou zmíněných algoritmů je nemožnost zacílit na všechny velikosti podezřelých oblastí.

3.1.2 Klasifikace

Klasifikační algoritmus je schopen při správném nastavení rozdělit dané množiny dle jejich charakteristických parametrů. V tomto případě bude klasifikátor rozhodovat, zda je příznakovým vektorem, získaným z mamogramu metodou pro detekci, zachycena rakovina. Pro úlohu mohou být použity rozmanité klasifikátory jako SVM klasifikátor nebo hluboká neuronová síť. Podrobněji se konkrétním klasifikačním algoritmům budeme věnovat v kapitole 5.

3.2 Úspěšnost jiných aplikací

Práci lze nejlépe srovnávat s podobnou aplikací jako je to v [8], kde byly použity metody LBP a klasifikační algoritmus SVM.

Úspěšnost komerčních CAD systémů podle [11]:

1. **R2 Technology** je první komerční firma vyvinuvší CAD systém, který vznikl v tomto odvětví. Podezřelou tkáň označuje hvězdičkou a kalcifikace trojúhelníkem. Detekce kalcifikací je prováděna s 98,5% úspěšností. U podezřelé tkáně je úspěšnost 85,7 %.
2. **Intelligent Systems Software** a její MammoReader dosahuje úspěšností okolo 89,3 %.
3. Firma **CADx Medical Systems** pracuje na nástroji SecondLook, který má sensitivitu 85 % (úspěšnost odhalení karcinomu/celkový počet podezřelých lézí) a je schopen odhalit 26,2 % případů 24 měsíců před diagnostikování rakoviny.

3.3 Databáze mamogramů

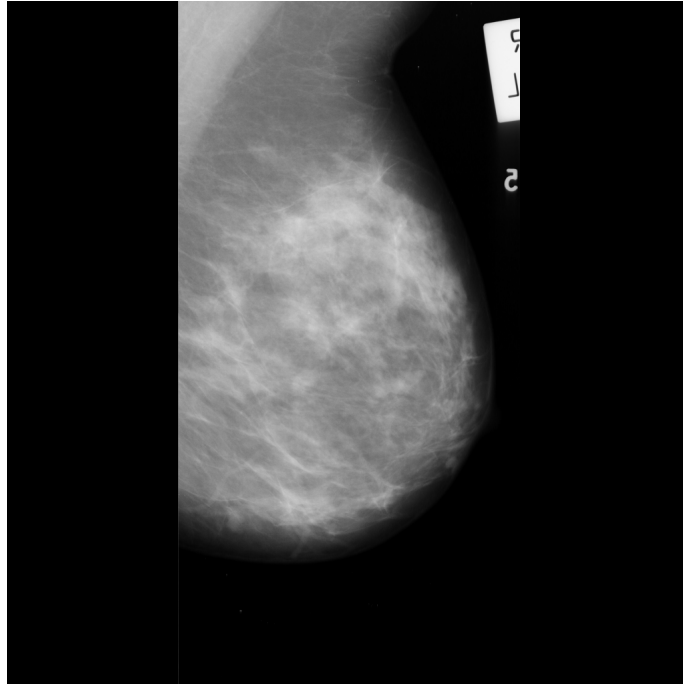
K dispozici je několik datových množin obsahujících snímky získané jako výsledek různých výzkumných projektů.

3.3.1 Digital Database for Screening Mammography (DDSM)

DDSM [2, 3] je největší známá databáze snímků obsahující aktuálně 2620 mamogramů z několika institucí. Snímky jsou rozdělené do tří základních skupin – zhoubné nádory (maligní nádory), nezhooubné nádory (benigní nádory) a normální snímky. Normálních snímků je v databázi 695, benigních 1011 a maligních 914. Obrázky jsou ve formátu `.jpeg`, který je nízkoztrátový, k dostání je však i databáze převedená do formátu `.png`. Dataset vznikl přímo pro výzkumné aktivity v této oblasti.

3.3.2 MiniMammographic Database (MIAS)

MIAS [12] je další rozšířená mamografická databáze v které můžeme najít 322 mamogramů o velikosti 1024 x 1024 pixelů ve formátu .pgm. Tento formát je určen pro rychlou práci s obrázky ve stupních šedi. Snímky jsou rozdělené na sedm skupin, dle typu diagnózy [12].



Obrázek 3.1: Ukázka obrázku z databáze MIAS

3.3.3 Mammographic patches

Mammographic patches je datová množina pro nekomerční použití, která obsahuje 2796 snímků 128 x 128 pixelů ve formátu .png. Obrázky byly vybrány z různých databází jako je DDSM, MIAS, apod. a mají zachycovat potenciálně nebezpečné snímky.

4 Relevantní metody z oblasti počítačového vidění

Při zpracování obrázků postupujeme nejčastěji tímto způsobem:

- předzpracování,
- extrakce příznaku,
- klasifikace.

V následujícím textu si tento postup rozebereme v souvislosti s naší úlohou.

4.1 Předzpracování

Předzpracování se v oblasti strojového učení používá pro přípravu daného objektu, v našem případě snímku, pro co nejsnazší a nejpřesnější extrakci příznaků (viz následující kapitola 4.2). Předzpracování může mít podobu filtrace parazitního šumu z obrázků, prahování obrázků, nebo například různých operací se signálem pro jeho lepší zpracování (vzorkování, různé posuny apod.).

Pro rentgenové obrázky se při předzpracování může použít metoda hledání největší kontury, protože jak můžeme vidět na obrázku 3.3.2, který pochází z mamografu se jako na většině rentgenových snímků vyskytuje destička. Destička podává radiologovi informaci o tom, jaké prso bylo snímáno a další doplňující informace. Tato destička však pro další zpracování snímku není potřebná a může negativně ovlivnit výsledky klasifikace. Na snímku je totiž navíc a pro každý obrázek bude v jiné pozici. Pro další práci s celými snímky tak bude muset být informační tabulka odstraněna.

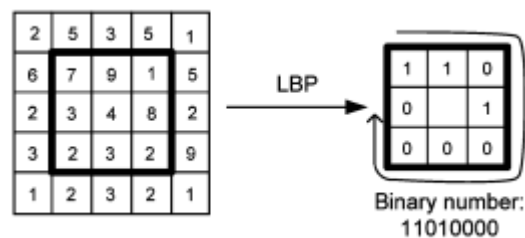
4.2 Lokální obrazové deskriptory

Reprezentaci získanou pomocí lokálních obrazových deskriptorů nazýváme příznakem a celou množinu metod zabývajících se tímto problémem metodami extrakce příznaků. Příznak může reprezentovat širokou škálu objektů jako je signál a nebo v našem případě obrázek. Samotný příznak si pak můžeme představit jako například vektor reprezentující určitý signál, podstatné je, že s tímto vektorem můžeme dále velmi jednoduše pracovat a také ho vhodně klasifikovat. Čím vhodnější extrakci příznaku zvolíme, tím pravděpodobněji dosáhneme při klasifikaci lepších výsledků. U lokálních obrazových deskriptorů se zpravidla jako příznak použije histogram výskytů detekovaných vzorů.

4.2.1 Local Binary Pattern (LBP)

Metoda lokálních binárních vzorů je deskriptor s nízkou výpočetní náročností a nízkou citlivostí na změny v osvětlení. Byl testován v mnoha různých odvětvích, například jako deskriptor pro obrazy mozku získané magnetickou rezonancí, deskriptor v oblasti umění, rozpoznávání obličejů nebo rozpoznávání otisků prstů. Díky svým vlastnostem je to jeden z nejčastěji používaných lokálních deskriptorů v medicínské informatice [8].

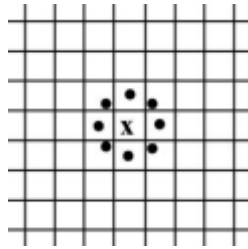
Základní **LBP(8,2)** operátor získáme porovnáváním jasové intenzity sousedních pixelů na mřížce 3 x 3, kde procházíme každý pixel v mřížce a porovnáváme ho s pixelem centrálním Z_0 . Při porovnávání se budeme řídit rovnicí 4.1. Výsledkem je binární reprezentace - vzor, přičemž stejný postup opakujeme pro celý obrázek. Poté je spočítán histogram četností výskytů vzorů, který je dále využit jako příznakový vektor.



Obrázek 4.1: Ukázka získání LBP operátoru [13]

$$Z(i) = \begin{cases} 1 & \text{pro } Z(i) \geq Z_0 \\ 0 & \text{pro } Z(i) < Z_0 \end{cases} \quad (4.1)$$

Důležitou vlastností LBP je rotační invariance - to znamená, že se nemusíme zabývat úhlem natočení obrázku, jak je právě obrázek natočený. Díky této konstrukci se umíme velmi dobře soustředit na hrany, určitá místa a body. Další zajímavou vlastností je možnost zacílit jen na některé prvky histogramu a ušetřit tak paměť. Selektci provedeme tak, že všechny vzory obsahující více než dva přechody z 0 na 1 nebo opačně vyřadíme [7].



Obrázek 4.2: LBP(8,2) s porovnáváním sousedů po kružnici [6]

Další modifikace můžou vzniknout použitím vhodného předzpracování, například pomocí Gáborova filtru.

4.2.2 Local Derivate Pattern (LDP)

Metoda LDP by měla být na rozdíl od LBP zaměřena na větší detaily, a také zachycovat směrovou informaci. Stejně jako LBP je pak tato metoda hojně používána na rozpoznávání obličejů.

Konstrukci vzoru pro LDP prvního řádu provádíme pomocí čtyř rovnic (4.2, 4.3, 4.4, 4.5). Každé pro jiné natočení.

$$I'_{0^\circ}(Z_0) = I(Z_0) - I(Z_4) \quad (4.2)$$

$$I'_{45^\circ}(Z_0) = I(Z_0) - I(Z_3) \quad (4.3)$$

$$I'_{90^\circ}(Z_0) = I(Z_0) - I(Z_2) \quad (4.4)$$

$$I'_{135^\circ}(Z_0) = I(Z_0) - I(Z_1) \quad (4.5)$$

LDP druhého řádu je definováno dle rovnice 4.6.

$$LDP_{\alpha}^2(Z_0) = [f(I'_{\alpha}(Z_0), I'_{\alpha}(Z_1)), f(I'_{\alpha}(Z_0), I'_{\alpha}(Z_2)) \dots f(I'_{\alpha}(Z_0), I'_{\alpha}(Z_8))] \quad (4.6)$$

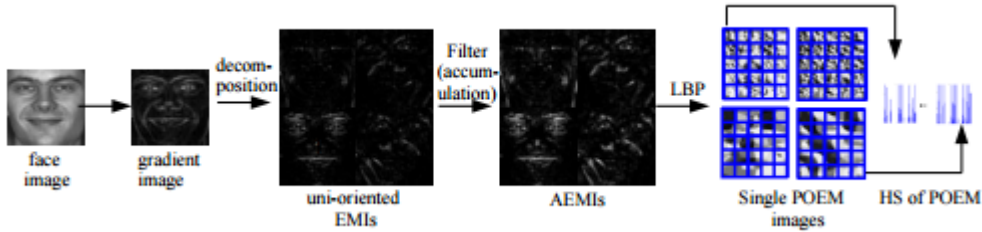
Přičemž $f(\cdot, \cdot)$ považujeme za binární kódovací funkci, která určí konkrétní bity výsledného vzoru pomocí rovnice 4.7 sestaveného na základě rovnice 4.8.

$$f(I'_{\alpha}(Z_0), I'_{\alpha}(Z_i)) = \begin{cases} 0 & \text{pro } I'_{\alpha}(Z_0) \cdot I'_{\alpha}(Z_i) > I'_{\alpha}(Z_0) \\ 1 & \text{pro } I'_{\alpha}(Z_0) \cdot I'_{\alpha}(Z_i) \leq I'_{\alpha}(Z_0) \end{cases} \quad (4.7)$$

$$LDP^2(Z) = [LDP_{\alpha}^2(Z) | \alpha = 0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}] \quad (4.8)$$

4.2.3 Patterns of Oriented Edge Magnitudes (POEM)

POEM je obrazový deskriptor používaný ve velké míře především pro rozpoznávání obličejů z důvodů jeho robustnosti.

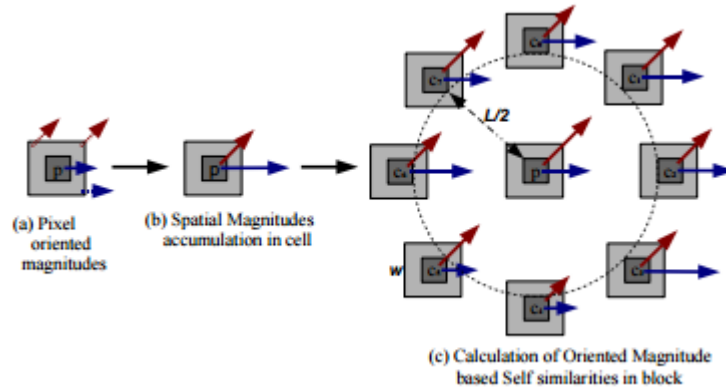


Obrázek 4.3: Ukázka výstupů metody POEM u rozpoznávání obličejů [13]

Postup se skládá z několika kroků, z nichž první je vypočtení gradientu obrázku. Informace o orientaci gradientu je pro každý pixel diskretizovaná na rozsah $0-\pi$ při bezznaménkové realizaci a $\pi-2\pi$ u realizace znaménkové. Pro každý pixel je gradientem 2D vektor, který reprezentuje m diskretizovaných směrů a jejich velikost.

Druhým krokem je zahrnutí informací o gradientu okolních pixelů pomocí určení lokálních histogramů pro každou z buněk (definováno na prostoru osmi okolí pro středový pixel). V poslední části použijeme nad doposud

získanými daty metodu LBP. Výsledný příznak sestavíme složením našich m směrových orientací.



Obrázek 4.4: Postupná konstrukce POEM operátoru [13]

5 Klasifikace

Po úspěšném připravení příznaků, které reprezentují zpracovaný obrázek obvykle pokračujeme klasifikací. Klasifikační algoritmus nám umožňuje snímky roztrždit do různých skupin podle specifických vlastností příznaků. Předpokládáme tedy, že při vhodné extrakci příznaků budou příznaky každé z námi definovaných skupin (v našem případě snímky zasažené/nezasažené rakovinou) rozdílné a pro počítač tak oddělitelné.

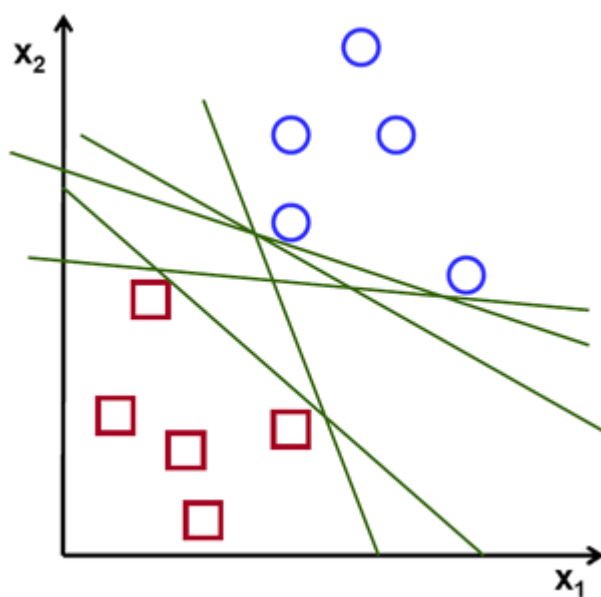
Pro samotnou klasifikaci budeme používat dva druhy klasifikátorů, metodu podpůrných vektorů (SVM) a vícevrstvý perceptron (MLP). Aby tyto klasifikátory byly schopné třídy separovat, musíme je nejprve třídy naučit. My zde použijeme učení s učitelem. Učení probíhá tak, že ještě před zahájením klasifikace jsou klasifikátoru předložena tzv. trénovací data (příznaky), u kterých je námi přidána informace, do které ze skupin data patří. Obvykle platí, že čím více klasifikátor učíme tím bude kvalitněji rozhodovat. To však není pravda vždy, protože může dojít i k tzv. přetrénování klasifikátoru, kdy klasifikátor svoje schopnosti separování tříd ztratí. Tento jev je individuální pro každý klasifikátor a je třeba vědět, že k němu může dojít. Dalším negativním vlivem při velkém množství trénovacích dat je delší doba trénování (učení) klasifikátoru.

5.1 Metoda podpůrných vektorů

SVM je hojně používaný klasifikační algoritmus v oblasti strojového učení, používá se pro celou řadu úloh vzhledem ke svým vlastnostem a krátké době trénování. Nutno také zmínit, že klasifikátor byl také úspěšně použit v článku [8], který byl pro tuto práci inspirací.

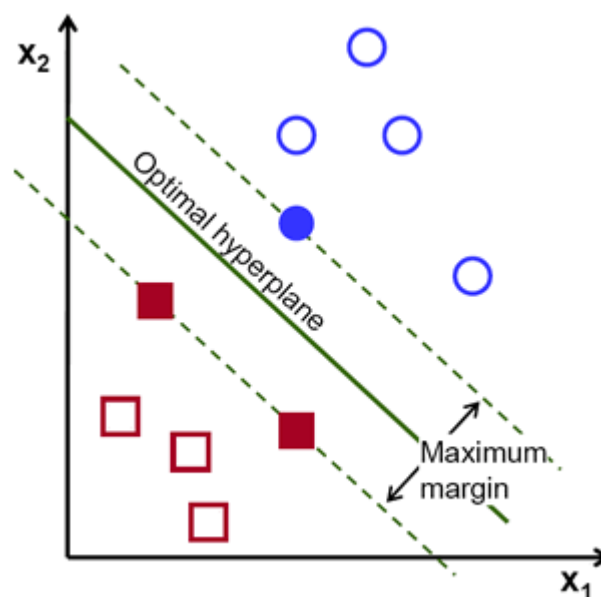
Algoritmus je postaven na výběru nadroviny (viz 5.1) s maximální separabilní účinností pro obě množiny, předpis pro nadrovinu můžeme vidět v rovnici 5.1.

$$g(\vec{x}) = \vec{w}^T \vec{x} + \omega_0 = \begin{cases} g(\vec{x} \geq 1) & \forall \vec{x} \text{ class } A \\ g(\vec{x} \leq -1) & \forall \vec{x} \text{ class } B \end{cases} \quad (5.1)$$



Obrázek 5.1: Hledání nadroviny klasifikátoru SVM [1]

Vzdálenost množin můžeme určit pomocí vztahu z rovnice 5.2, kde počítáme vzdálenost bodu od přímky (viz 5.1).



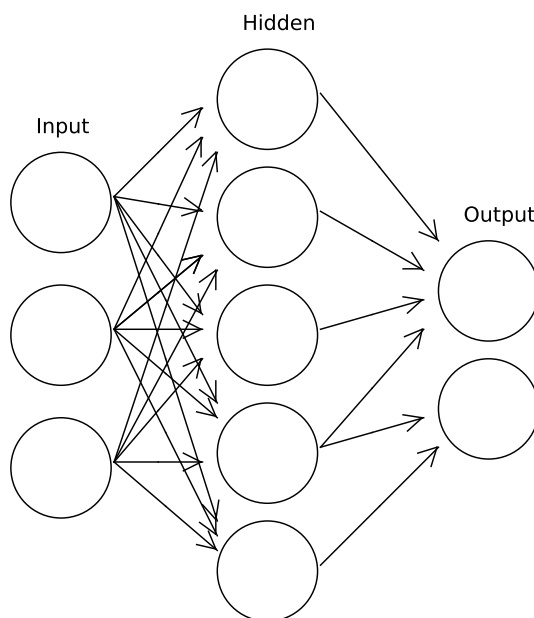
Obrázek 5.2: Nalezení nadroviny klasifikátoru SVM [1]

$$\frac{|g(x_{\omega_1})|}{|\omega|} + \frac{|g(x_{\omega_2})|}{|\omega|} = \frac{1}{|\omega|} + \frac{1}{|\omega|} = \frac{2}{|\omega|} \quad (5.2)$$

Nakonec zbývá minimalizovat ω a maximalizovat tak vzdálenost dvou množin a tím pádem i dosáhnout největší možné separability. Tato úloha se provádí pomocí tzv. Lagrangeových multiplikátorů.

5.2 Vícevrstvý perceptron

Vícevrstvý perceptron je nejpoužívanější neuronovou sítí. Sít' se skládá z vstupní vrstvy, minimálně jedné skryté vrstvy a vrstvy výstupní. Jak už název napovídá, vstupní vrstva je umístěna na vstupu neuronové sítě, vrstvy skryté jsou umístěny mezi vstupní a výstupní vrstvou. Výstupní vrstva je vrstvou poslední a z každého neuronu, který se v ní nachází, můžeme odečíst konkrétní hodnotu pro klasifikovanou třídu. Na obrázku 5.2 můžeme vidět trojvrstvý perceptron se vstupní vrstvou obsahující 3 neurony, skrytou vrstvou o 5 neuronech a výstupní vrstvou o 2 neuronech - budeme tedy klasifikovat tři různé vstupy.



Obrázek 5.3: Neuronová sít' [1]

Všechny neurony umístěné v MLP jsou stejné, přičemž mohou být spo-

jeny s n různými neurony z předchozí vrstvy a navazovat na m následujících neuronů v další vrstvě (neurony ve skryté vrstvě). Hodnota na vstupu neuronu je počítána jako suma výstupů z předchozí vrstvy ($\omega_i x_i$), které do neuronu vcházejí s přičtením váhové funkce θ (na vstupu váha ω_0 a $x_0 = 1$), viz rovnice 5.3. Výsledek je poté transformován na základě aktivační funkce 5.4.

$$u = \sum_{i=1}^n (\omega_i x_i + \theta) \quad (5.3)$$

$$y = f(u) \quad (5.4)$$

Aktivační funkce mohou mít více podob, mezi ty nejčastěji používané patří sigmoida (rovnice 5.5), hyperbolický tangens (rovnice 5.6) a nebo omezená lineární funkce.

$$f(u) = \frac{1}{1 + e^{-u}} \quad (5.5)$$

$$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (5.6)$$

6 Implementace

6.1 Vybrané metody

Po dohodě s vedoucím práce jsem se rozhodl, že práci budu navazovat na již funkční program pro rozpoznávání obličejů [5]. V programu byly z větší části integrovány metody obrazových deskriptorů LBP, LDP a POEM, přičemž chyběla klasifikační část. Program neuměl předzpracovat snímky z mamografu způsobem, který by odstranil rušivé elementy jako například destičku ve snímcích. Proto byly do práce zahrnuty všechny tři již implementované deskriptory a bylo zapotřebí program adaptovat pro klasifikaci mamogramů. Dále bylo nutné vytvořit část pro klasifikování snímků, kde byly jako metody pro klasifikaci zvoleny klasifikátory SVM a MLP pro velmi dobré výsledky v různých aplikacích. Dalším důvodem použití klasifikačního algoritmu SVM byl fakt, že v článku, jímž je práce inspirována, byl použit také.

Vzhledem k faktu, že původní program byl napsán v jazyce C++ a používal knihovnu OpenCV, rozhodl jsem se v tom pokračovat. Sám jsem se poté rozhodl přidat ještě knihovnu Boost C++.

6.2 Použité knihovny

6.2.1 OpenCV

OpenCV [1] je multiplatformní knihovna se zaměřením na zpracování obrazu, která je šířena pod BSD licenci. Je tedy možné ji využít i v komerční sféře (až na několik málo výjimek, které jsou patentovány v USA). Knihovnu můžeme využít při psaní v jazyce C/C++, Java (i pro Android) a nebo Python. Podporuje operační systémy Windows, Linux, iOS a Android. Dále je velkou výhodou vysoká míra optimalizace umožňující velmi rychlou práci nad velkým objemem dat a také univerzálnost. Zde je implementováno mnoho základních i pokročilých mechanismů pro zpracování obrazu. Počet stažení této knihovny překročil počet 9 milionů a vzniklo proto mnoho užitečných návodů jak s knihovnou pracovat. Zároveň je celá knihovna na

oficiálních stránkách výborně zdokumentovaná.

6.2.2 Instalace OpenCV na OS Ubuntu 14.04 LTS

Budeme potřebovat knihovnu OpenCV ve verzi 2.4.11, a to z důvodu zpětné nekompatibility původního programu s novějšími verzemi OpenCV.

V prvním kroku stáhneme požadované instalační balíčky OpenCV. Do terminálového okna postupně zadáme následující příkazy.

```
sudo apt-get install build-essential

sudo apt-get install cmake git libgtk2.0-dev pkg-config
libavcodec-dev libavformat-dev libswscale-dev

sudo apt-get install python-dev python-numpy libtbb2
libtbb-dev libjpeg-dev libpng-dev libtiff-dev
libjasper-dev libdc1394-22-dev
```

Následně naklonujeme zdrojové kódy OpenCV z repozitáře na Gitu. Webová stránka s repozitářem a verzí 2.4 se nachází na odkazu níže.

```
https://github.com/Itseez/opencv/tree/2.4
```

Poté vytvoříme složku odkud spustíme instalační soubory. Vstoupíme do složky a v ní zadáme následující příkaz.

```
cmake [<some optional parameters>] <path to the OpenCV
source directory>
```

Například:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=.
```

Nakonec v naší instalační složce zadáme posloupnost příkazů uvedených níže.

```
make

sudo make install
```

Zda instalace proběhla úspěšně můžeme ověřit zadáním následujícího příkazu.

```
pkg-config opencv --libs
```

Po jeho zadání by se nám měla ukázat nainstalovaná verze OpenCV.

6.2.3 Boost C++ libraries

Druhou použitou knihovnou je Boost, která je stejně jako OpenCV použitelná pro výzkumné i komerční účely a spustitelná na operačních systémech Windows a Linux. Jedná se o balík knihoven pro implementaci nejrůznějších synchronizačních prvků, vláknové komunikace, apod. Já jsem v práci použil knihovnu `filesystem` z důvodu přenositelné implementace při vytváření adresářů.

6.2.4 Instalace Boost C++ na OS Ubuntu 14.04 LTS

Instalace knihovny je velmi jednoduchá, pro stažení potřebných knihoven stačí dodržet následující postup.

```
sudo apt-get install libboost-all-dev
```

Pro kontrolu nainstalované verze zadáme následující:

```
dpkg -s libboost-dev | grep 'Version'
```

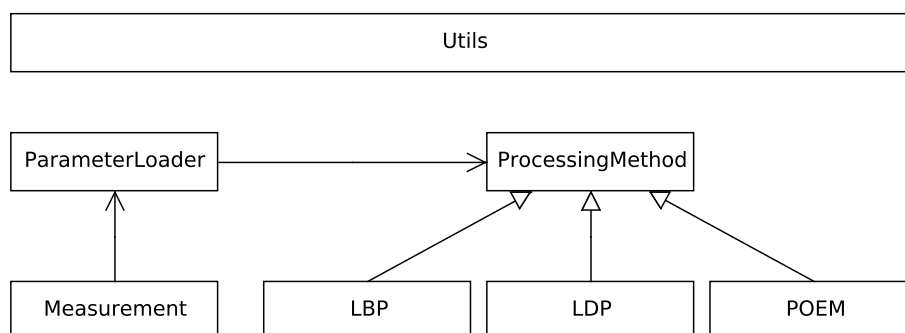
6.3 Knihovna pro rozpoznávání obličejů

Knihovna [5] sloužila pro rozpoznávání obličejů. Byly zde implementovány tři typy lokálních obrazových deskriptorů (LBP, LDP a POEM). Pro klasifikaci byly použity různé druhy metrik. Dále byl implementován genetický algoritmus.

6.4 Navržený systém pro automatickou identifikaci rakoviny

6.4.1 Architektura programu

Architekturu původního programu se mi podařilo zachovat, takže rozdělení programu do logických celků zůstává stejné jako v [5] (viz obrázek 6.4.1).



Obrázek 6.1: Architektura programu [5]

V následujícím bloku budou popsány implementované třídy a zároveň zdůrazněny jejich nejdůležitější funkce.

- **Utils** - Třída obsahující především funkce pro měření doby běhu programu (`startTimeMeasure`, `stopTimeMeasureGetMs`) a funkci pro normalizaci histogramu (`normalizeHistogram`).
- **ProcessingMethod** - Rodičovská třída, která je poté třídami LBP, LDP a POEM odděděna. Obsahuje společné metody používané i v implementacích potomků jako je načítání obrázku (`LoadImageFromFile AndSegmentate`, `LoadImageFromFile`), které byly změněny v rámci této práce, předzpracování daného obrázku (`PreprocessImage`) a konstrukce globálního histogramu (`ComputeGlobal Histogram`).
- **ParameterLoader** - Třída pro zpracování konfiguračního souboru. Uchovává všechny potřebné údaje jako je cesta ke kořenové složce s daty, počty ROI, typ klasifikátoru, velikost prahu pro klasifikaci, typ snímků a pro každou z metod lokálních deskriptorů konkrétní nastavení. Tyto data poté pomocí nově implementovaných metod `get` (`GetPositive TrainDirPath`, `GetTypeOfClassifier`, apod.) používají další třídy.

- **Measurement** - Jedná se o třídu pro klasifikaci. Nachází se zde metody, implementované v rámci této práce, pro označení zpracovávaných oblastí (**IsInteresting**), metody pro trénování klasifikátorů (**TrainClassifier**) nebo pro samotnou klasifikaci (**ComputeMLP**, **ComputeSVM**), funkce pro vypsání výsledků klasifikace (**GetResults**) či funkce pro experimentální pokus s algoritmem K-means.

6.4.2 Nahrávání obrázků

Při spuštění programu se nejdříve načte konfigurační soubor (viz 6.4.6) pomocí metody **LoadParametersFromFile** ve třídě **ParameterLoader**. Tím jsou programu zpřístupněná veškerá nastavení pomocí metod **get**, například počet ROI, práh dle kterého bude probíhat klasifikace, typ klasifikátoru, cesta ke kořenovému adresáři obsahujícím snímky (viz 6.4.7), atp. Dále jsou vstupní parametry zpracovány ve třídě **Measurement** funkcí **LoadFiles**, kde pomocí již zmíněných metod přetížíme funkci **LoadFiles**, ve které zavoláme funkci **FillFileImages**. Pro každý z datových setů bude naplněna jedna proměnná typu `std::vector<FileImage>` reprezentující daný dataset následujícím způsobem:

- **positiveTrainSet** - všechny snímky ze složky obsahující pozitivní trénovací množinu,
- **negativeTrainSet** - všechny snímky ze složky obsahující negativní trénovací množinu,
- **positiveTestingSet** - všechny snímky ze složky obsahující pozitivní testovací množinu,
- **negativeTestingSet** - všechny snímky ze složky obsahující negativní testovací množinu.

Přičemž o každém snímku je evidována informace o jeho jméně, indexu v dané složce a vektor všech histogramů pro daný snímek (struktura **FileImage**).

6.4.3 Předzpracování

Předzpracování probíhá v metodě `ProcessImage`, která je implementována ve třídě `Measurement` a je volána pro každý obrázek zvlášť. Na základě informace o tom, zda se jedná o ROI nebo celý snímek z mamogramu se odvíjí předzpracování

- `LoadImageFromFile` - metoda volaná při zpracování ROI, kdy se tento obrázek pouze načte,
- `LoadImageFromFileAndSegmentate` - zde je implementována metoda hledání největší kontury a základní prahování pro odstranění rušivých elementů ve snímku.

Následuje funkce (`LoadImageFromFile`) pro předzpracování obrázků implementované již v práci původní. Zde se nachází například gaussovské rozostření nebo gamma korekce. Následuje volání metod pro extrakci příznaků.

6.4.4 Extrakce příznaku

Extrakce příznaků je implementována pomocí virtuální funkce `ProcessImage` ze třídy `ProcessingMethod`, protože konkrétní metoda bude rozdílně implementována ve svém modulu

- LBP – třída pro realizaci algoritmu LBP,
- LDP – třída pro realizaci algoritmu LDP,
- POEM – třída pro realizaci algoritmu POEM.

Tyto třídy zůstaly zachovány z původní práce, přičemž všechny histogramy byly rozšířeny o informaci o své pozici z důvodu lepší identifikace daného segmentu a také ověření nově přidaných metod.

- K-means – v práci jsme se také pokusili využít metodu k-means, pro označení potenciálně podezřelých míst na snímku. Metoda je implementována v modulu `Measurement` pomocí metody `KMeans` a jejími pomocnými funkcemi jsou `ComputeCentroid`, `DetectChanges` a `ChangeColors`, která umožňuje kontrolu správnosti celého postupu díky vykreslení zpracovávaného snímku a označení rozdělených množin.

- **Metoda detekce nevýznamných oblastí** – dále nutno zmínit ještě metodu pro označení částí, které skutečně chceme trénovat, popřípadě klasifikovat. Ve snímcích (ROI nevyjímaje, ale v malém zastoupení) nachází prázdná místa (černé plochy bez informace), která by zhoršila výsledky klasifikátorů MLP a SVM. Pro označení a vyjmutí z dalšího zpracování používáme funkci `isInteresting`. Ta za předpokladu, že v daném segmentu, nad kterým je vypočten histogram, najdeme v obrázku černou plošku, plošku detekuje. Následně je nakonec histogramu zařazena -1 a tento histogram už dále není zpracován, jinak zařadíme na konec 1.

6.4.5 Klasifikace

Pro trénování klasifikátorů je implementována funkce `TrainClassifier` z modulu `Measurement`, přičemž je v těle metody naplněna matice `trainingMatTemp` pozitivními a negativními trénovacími daty, které jsou poté překonvertovány do vhodného formátu pro klasifikátor a uloženy do matice `trainingMat`. Následně je naplněna matice `labels` dle počtu pozitivních a negativních vzorků hodnotami -1 pro pozitivní, 1 pro negativní snímky. Po této akci se na základě zvoleného typu klasifikátoru v konfiguračním souboru zavolá funkce

- `TrainSVM` - pro trénování klasifikátoru SVM,
- `ComputeSVM` - pro testování klasifikátoru SVM,
- `TrainMLP` - pro trénování klasifikátoru MLP,
- `ComputeMLP` - pro testování klasifikátoru MLP.

Vracené výsledky následně uložíme do proměnné `vectorsOfThreshs` typu `std::vector<double>`, kterou předáme funkci `GetResults`. Tato funkce už pouze vypíše výsledky klasifikace.

6.4.6 Konfigurační soubor

V následující části se budeme věnovat popisu konfiguračního souboru, kde jsem se snažil zachovat strukturu z předchozí práce. Formát zápisu parametrů vypadá následovně

název parametru=hodnota parametru

Například: `rootDirectory=/home/michal/data/patches`

Pro psaní komentářů slouží znak `" ; "`.

Nyní se podíváme na volitelné parametry, které lze do konfiguračního souboru uvést pro nastavení programu

- `usedMethod=<int>` – parametr pro výběr typu obrazového deskriptoru (0 = LBP, 1 = LDP, 2 = POEM),
- `rootDirectory=<string>` – cesta ke kořenovému adresáři obsahujícímu trénovací a testovací množiny snímků (viz kapitola 6.4.7),
- `extension=<string>` – značí příponu, kterou mají zpracovávané snímky,
- `numberOfPositiveROIs=<int>` – počet ROI mezi pozitivními trénovacími daty,
- `numberOfNegativeROIs=<int>` – počet ROI mezi negativními trénovacími daty,
- `numberOfTestsNegativeROIs=<int>` – počet ROI mezi negativními testovacími daty,
- `numberOfTestsPositiveROIs=<int>` – počet ROI mezi pozitivními testovacími daty,
- `typeOfClassifier=<int>` – parametr pro výběr klasifikátoru (0 = SVM, 1 = MLP),
- `threshForClassification=<int>` – minimální počet lokálních histogramů, aby byl snímek označen jako zasažený.

Zde končí výčet základních parametrů, které nejsou spjaty s nastavením konkrétního obrazového deskriptoru. Pro nastavení LBP používáme

- `lbpParameters.useUniformBinsOnly=<int>` – informace o tom, zda se mají (hodnota 1) nebo nemají (hodnota 0) používat uniformní vzory,
- `lbpParameters.gridSize=<int>` – velikost mřížky, která rozdělí snímek pro výpočet lokálních histogramů.

Při použití LDP nastavujeme tyto proměnné

- `ldpParameters.useUniformBinsOnly=<int>` – informace o tom, zda se mají (hodnota 1) nebo nemají (hodnota 0) používat uniformní vzory,
- `ldpParameters.gridSize=<int>` – velikost mřížky, která rozdělí snímek pro výpočet lokálních histogramů,
- `ldpParameters.order=<int>` – řád metody.

Pro POEM je parametrů z důvodů větší složitosti metody nejvíce

- `poemParameters.useUniformBinsOnly=<int>` – informace o tom zda se mají (hodnota 1) nebo nemají (hodnota 0) používat uniformní vzory,
- `poemParameters.gridSize=<int>` – velikost mřížky, která rozdělí snímek pro výpočet lokálních histogramů,
- `poemParameters.gradientOrientationCount=<int>` – parametr určující počet orientací gradientů,
- `poemParameters.gradientType=<int>` – typ gradientu (0 = 2D filtr, 1 = Scharrův operátor, 2 = Sobelův operátor),
- `poemParameters.gradientSignedRepresentation=<int>` – určuje použití neznaménkové (hodnota 0) nebo znaménkové (hodnota 1) reprezentace gradientu,
- `poemParameters.cellSize=<int>` – čtvercové okolí pro středový pixel,
- `poemParameters.blockSize=<int>` – poloměr kružnice na které je použita metoda LBP v závěrečné fázi výpočtu POEM,
- `poemParameters.useBilinearInterpolation=<int>` – informace o tom, zda nebude použita bilineární interpolace (hodnota 0) nebo bude použita bilineární interpolace (hodnota 1),
- `poemParameters.tauConst=<float>` – parametr vyrovnávající stabilitu v obrázku.

6.4.7 Trénovací a testovací soubory

V konfiguračním souboru se dále nastavuje cesta ke kořenovému adresáři s datovými sety. V adresáři označeném jako `rootDirectory` je vyžadována následující struktura adresářů, které budou procházeny

- `train_negative` - adresář pro negativní trénovací snímky,
- `train_positive` - adresář pro pozitivní trénovací snímky,
- `test_negative` - adresář pro negativní trénovací snímky,
- `test_positive` - adresář pro pozitivní trénovací snímky.

Pokud tomu tak nebude program skončí chybovou hláškou.

```
michal@mmedek-pc:~/NetBeansProjects/data$ tree patches -L 1
patches
├── test_negative
├── test_positive
├── train_negative
└── train_positive
```

Obrázek 6.2: Ukázka adresářové struktury (`rootDirectory` je v tomto případě adresář `patches`)

7 Provedené experimenty

V této kapitole se budeme věnovat popisu provedených experimentů. Experimenty jsou popsány v následujícím pořadí:

- hledání optimální velikosti mřížky operátoru LBP pro oba klasifikátory,
- hledání optimální velikosti mřížky a řádu metody LDP pro oba klasifikátory.

Díky těmto experimentům jsme schopni určit optimální nastavení metod obrazových deskriptorů při základním nastavení klasifikátorů. Dále použijeme takto nastavené parametry a budeme pokračovat optimalizací výkonu klasifikátorů:

- analýza velikosti prahu pro MLP s použitím metody LBP,
- analýza velikosti trénovací množiny pro LBP v závislosti na počtu iterací pro klasifikátor MLP,
- analýza počtu neuronů ve skryté vrstvě MLP pro metodu LBP,
- analýza velikosti prahu pro MLP s použitím metody LDP,
- analýza velikosti trénovací množiny pro LDP v závislosti na počtu iterací pro klasifikátor MLP,
- analýza počtu neuronů ve skryté vrstvě MLP pro metodu LDP,
- analýza velikosti prahu pro SVM s použitím metody LBP,
- analýza velikosti prahu pro SVM s použitím metody LDP.

Pro klasifikátor SVM další optimalizaci provádět nebudeme, protože je použito automatické trénování.

7.1 Nastavení parametrů obrazových deskriptorů

Nejdříve se budeme věnovat hledání optimálních parametrů pro nastavení metod lokálních obrazových deskriptorů se základním nastavením klasifikátorů.

Při vyhodnocování efektivity metody klasifikátorem jsme jako veličinu sloužící k popsání kvality těchto metod a nastavení daného klasifikátoru použili úspěšnost (rovnice 7.1) a F-míru (rovnice 7.2).

$$Acc = (tp + tn)/(tp + tn + fp + fn) \quad (7.1)$$

Kdy jako Acc značíme úspěšnost, jako tp značíme *true positives* snímky (vyhodnocené jako pozitivní a pozitivními jsou). Dále se v rovnici nachází tn , tedy *true negatives* snímky (negativní a taktéž byly zařazeny do správné třídy). Na druhé straně máme snímky zařazené nesprávně a to fp (*false positives*) – snímky nesprávně označené jako pozitivní a také fn (*false negatives*) – snímky nesprávně označené jako negativní, kterých bychom se v této úloze chtěli obzvláště vyhnout. Klasifikace mamogramu jako falešně negativního by znamenala přehlédnutí pravděpodobného nádoru u pacienta.

$$F_m = 2 \cdot pr \cdot rec/(pr + rec) \quad (7.2)$$

Pro označení F-míry jsme v tomto případě použili F_m , přičemž neznámou pr , nebo-li přesnost (*precision*) získáme pomocí vztahu v rovnici 7.3. Druhou neznámou je rec - úplnost (*recall*) pro její výpočet vycházíme ze vztahu v rovnici 7.4.

$$pr = tp/(tp + fp) \quad (7.3)$$

$$rec = tp/(tp + fn) \quad (7.4)$$

7.1.1 Nastavení parametrů LBP

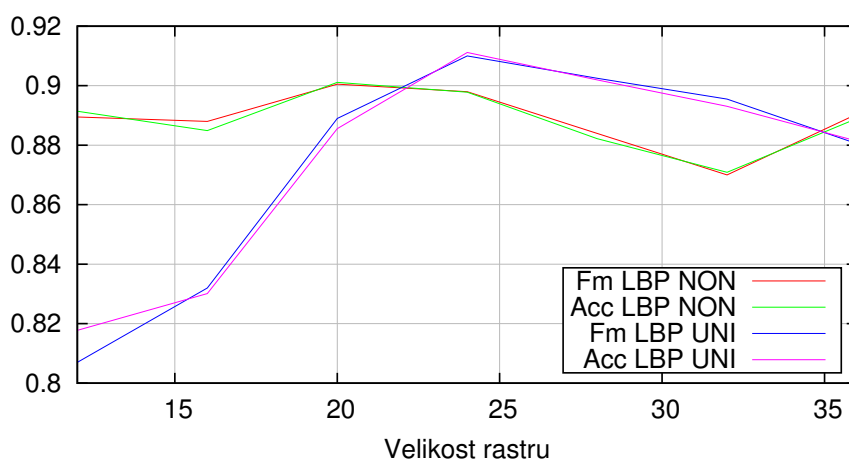
Jako první bude testován algoritmus LBP z důvodu jeho jednoduchosti. Dalším důvodem je skutečnost, že pokud by LBP operátor dosahoval špatných výsledků, další metody na tom pravděpodobně nebudou o moc lépe, protože jsou sofistikovanější a měly by dosahovat lepších výsledků. Mezi nastavitelné parametry u LBP patří `lbpParameters.useUniformBinsOnly` a `lbpParameters.gridSize`. Jako práh určující u snímků skutečnost, zda jsou nebo nejsou zasažené, budeme volit poměr vyjádřený v rovnici 7.5, kdy budeme procházet různé velikosti prahu (krok 0.05) a kontrolovat výsledky. Nejlepší výsledek bude poté zahrnut do měření. Výpočet tohoto prahu byl použit z důvodů stejné velikosti ROI a snazšímu měření, dále bude nahrazen prahem značícím počet zasažených lokálních histogramů. Zjednodušení experimentu spočívá v tom, že u relativního prahu nám nekomplikuje měření fakt, že při různé velikosti mřížky dostaneme různý počet extrahovaných příznaků. Pro trénování bude použito 100 pozitivních a 100 negativních ROI a pro potřeby testování použijeme množinu čítající 1000 pozitivních a 1000 negativních ROI z databáze, která vznikla na naší fakultě v rámci jednoho z projektů. Databáze se skládá z ROI o velikosti 128 x 128 pixelů ve formátu `.png` a obsahuje 14 842 negativních a 2 797 pozitivních snímků, které byly získány z databází uvedených v sekci 3.3.

$$práh = PPH / (PPH + PNH) \quad (7.5)$$

Kde jako PPH značíme počet pozitivních lokálních histogramů na snímku a jako PNH počet negativních příznaků.

Klasifikátor MLP byl nastaven na hodnoty doporučené v [1] a jeho optimalizaci se budeme věnovat dále (viz sekce 7.2). Pro klasifikátor SVM byla použita metoda automatického hledání ideálního nastavení `train_auto` z knihovny OpenCV a použili jsme lineární jádro. Automatické nastavení bylo použito také z důvodu, že SVM klasifikátor byl velmi citlivý na změny v datech, a tak by například při každé změně velikosti mřížky musel být klasifikátor znovu přenastaven, abychom dosáhli dobrých výsledků.

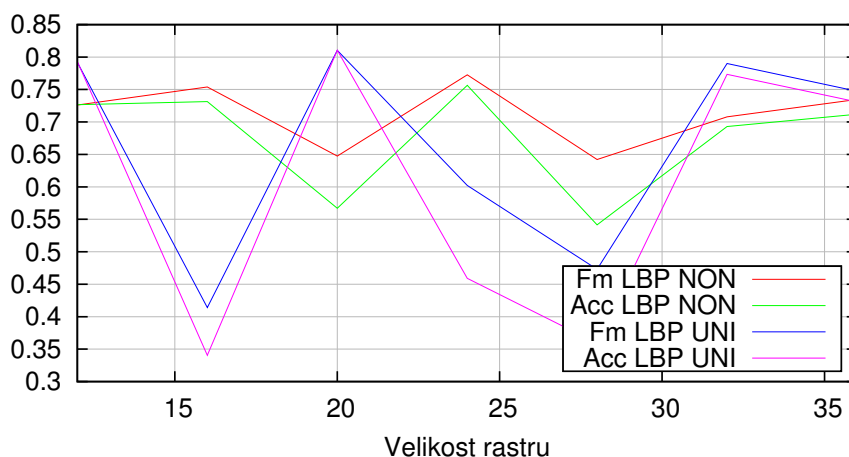
První experiment byl proveden pomocí metody LBP a použití klasifikátoru MLP se snahou najít ideální velikost mřížky pro výpočet lokálních histogramů.



Obrázek 7.1: Graf metody LBP znázorňující závislost úspěšnosti a F-míry na velikosti mřížky při použití klasifikátoru MLP

Graf jasně ukazuje, že při zmenšující se velikosti mřížky má neuniformní varianta stále stabilní výsledky, přičemž se uniformní varianta zhoršuje. Nejlepších výsledků se nám pro metodu LBP a klasifikátor MLP podařilo dosáhnout u uniformní varianty při velikosti mřížky 24 pixelů.

Pokračovali jsme stejnými experimenty pro klasifikátor SVM.



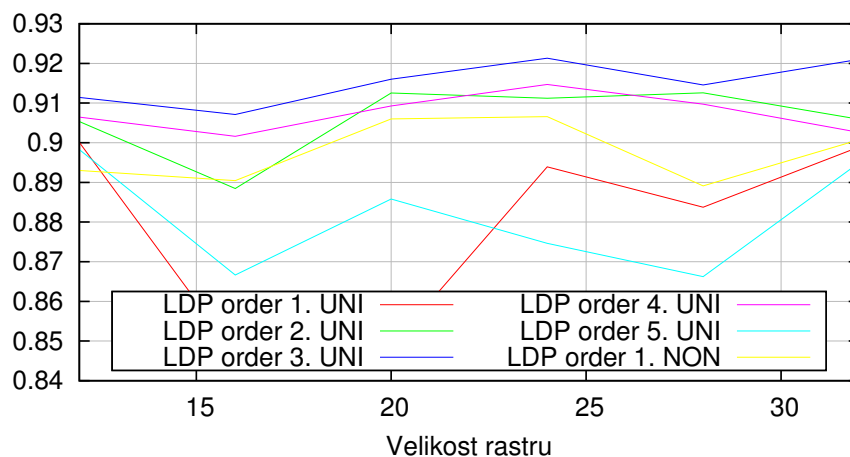
Obrázek 7.2: Graf metody LBP znázorňující závislost úspěšnosti a F-míry na velikosti mřížky při použití klasifikátoru SVM

Zde můžeme pozorovat opět lepší výsledky pro uniformní variantu LBP.

Dobré výsledky jsou však velmi nevyrovnané a dochází k výrazným propadům, které mohou být způsobeny rozdílnou kvalitou natrénování klasifikátoru. U neuniformní varianty stojí za povšimnutí lepší stabilita. Nejvyšší úspěšnosti a F-míry jsme dosáhli s velikostí mřížek 20 a 32 pixelů. Výsledky byly úspěšnější při velikosti 20 pixelů, kde lze opět říct, že jak úspěšnost tak F-míra jsou na tom procentuálně velice podobně.

7.1.2 Nastavení parametrů LDP

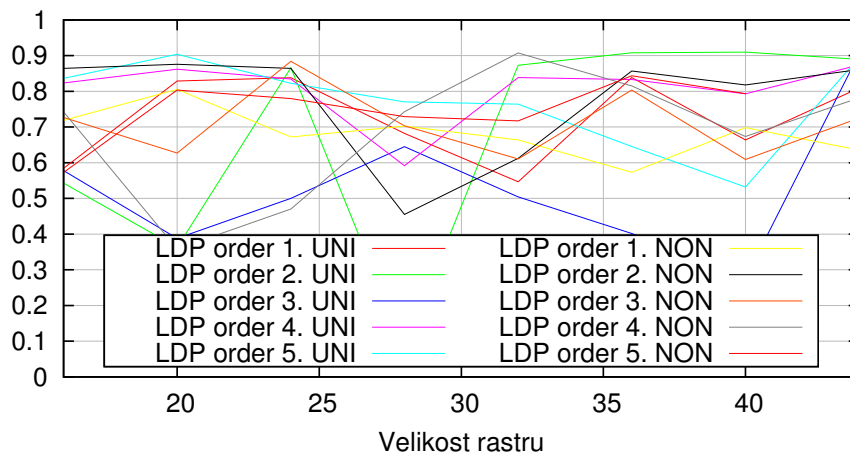
Od metody LDP očekáváme zlepšení výsledků, které jsme již získali pomocí metody LBP. Očekávanou negativní vlastností je velká nevyrovnanost výsledků [5] v kombinaci se složitým nastavením ideálních parametrů i přesto, že oproti předchozí metodě přibude parametr pouze jeden a to `ldpParameters.order`, který určuje řád metody LDP. Obdobně jako u LBP pak budeme nastavovat použití uniformních vzorů (`ldpParameters.useUniformBinsOnly`) a velikost mřížky (`lbpParameters.gridSize`). Práh a trénovací množinu použijeme pro tento experiment stejné jako při hledání optimálního nastavení LBP.



Obrázek 7.3: Graf znázorňující závislost F-míry při použití metody LDP na velikosti mřížky při použití klasifikátoru MLP

Nejllepších výsledků se nám podařilo dosáhnout u LDP třetího řádu a to pro uniformní variantu. Můžeme si také všimnout, že se od čtvrtého řádu výsledky zhoršují. Z důvodu klesající F-míry a velké časové náročnosti bylo od měření pro neuniformní varianty vyšších řádů upuštěno.

Pokračovali jsme experimenty s klasifikátorem SVM, kde jsme nejlepších výsledků dosáhli při použití uniformní varianty LDP při velikosti mřížky 40 pixelů a řádu 2.



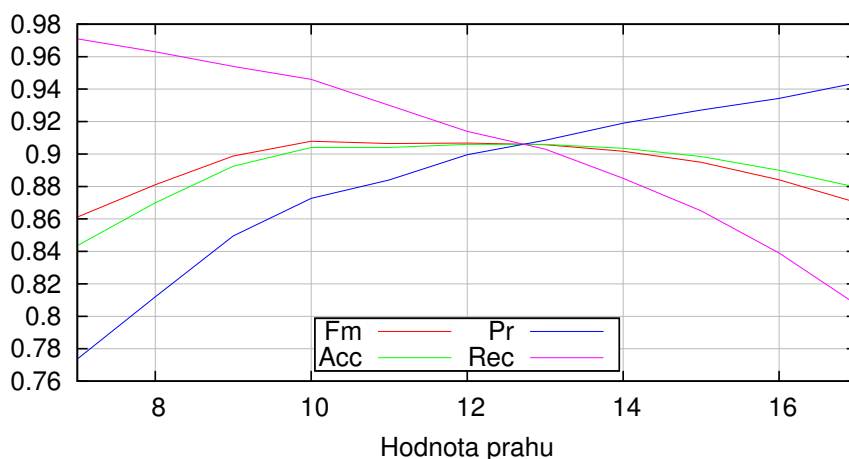
Obrázek 7.4: Graf znázorňující závislost F-míry při použití metody LDP na velikosti mřížky při použití klasifikátoru SVM

7.2 Optimalizace klasifikátorů

Po nastavení velikosti mřížky u metod obrazových deskriptorů jsme se zaměřili na optimalizaci klasifikátorů. Tento postup jsme zvolili, protože pokud bychom postupovali opačně mohlo by se stát, že bychom sice našli optimální nastavení klasifikátoru pro určitou velikost mřížky. Pokud by však bylo nastavení mřížky špatně, nikdy bychom tak kvalitních výsledků nedosáhli. Proto jsme nejdříve našli optimální velikost mřížky a nyní se budeme věnovat optimalizaci klasifikátoru.

7.2.1 Nastavení parametrů MLP

Další experimenty jsme provedli z důvodu nalezení optimální velikosti prahu – počtu pozitivních histogramů ve snímku. Z předchozích výsledků jsme zvolili mřížku o velikosti 24 pixelů, protože bylo měření nejúspěšnější právě při této velikosti a budeme ho používat i dále (`1bpParameters.gridSize=24`).



Obrázek 7.5: Graf znázorňující závislost úspěšnosti a F-míry při použití metody LBP na velikosti prahu při použití klasifikátoru MLP

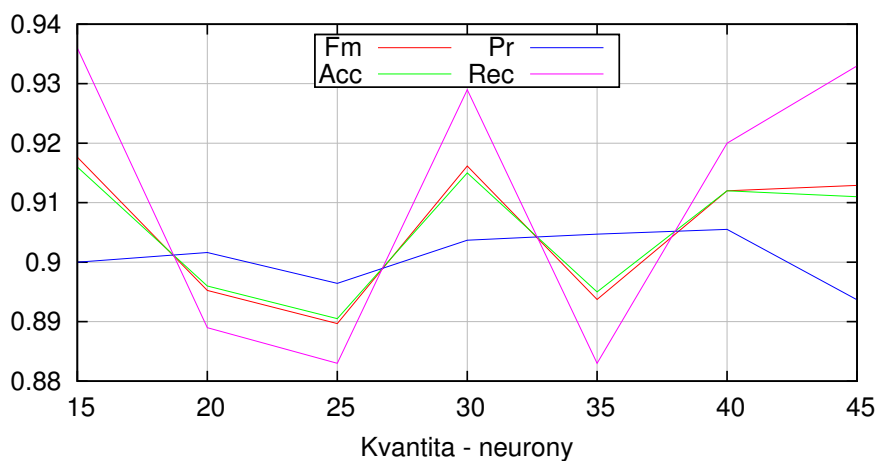
Pro metodu LBP jsme nejkvalitnější klasifikace docílili při nastavení prahu na 12 nalezených pozitivních lokálních histogramů. Při vyšším počtu bude počet falešně pozitivních chyb významně vyšší. V dalších měření s klasifikátorem MLP budeme pro metodu LBP používat nastavení prahu na 12 pozitivních nálezů (`threshForClassification=12`).

Následovala skupina experimentů, kdy jsme zjišťovali závislost výsledků na velikosti trénovací množiny a počtu iterací klasifikátoru. V tabulce 7.1 jako *PI* značíme počet iterací klasifikátoru uvedených v prvním sloupci a jako *PT* počet trénovacích vzorků pro každou z množin (pozitivní trénovací a negativní trénovací vzorky). Tabulka ukazuje, že v některých případech dochází k přetrénování klasifikátoru (viz sekce 5), kdy při větším počtu iterací a větší trénovací množině dochází k naměření horších výsledků. Jev je nejlépe pozorovatelný při velikosti množiny nad 500 vzorků. Nejlepších výsledků jsme dosáhli při velikosti trénovací množiny čítající 100 vzorků a při počtu 500 iterací.

Tabulka 7.1: Závislost výsledků na počtu iterací a velikosti dat pro metodu LBP a klasifikátor MLP

PI	PT	50	75	100	200	300	500	1000	1750
1	<i>Acc</i>	0.633	0.772	0.843	0.871	0.870	0.908	0.820	0.744
	<i>F_m</i>	0.723	0.795	0.850	0.872	0.870	0.910	0.791	0.668
	<i>pr</i>	0.580	0.721	0.811	0.863	0.868	0.888	0.938	0.950
	<i>rec</i>	0.960	0.885	0.893	0.882	0.871	0.933	0.684	0.514
10	<i>Acc</i>	0.872	0.909	0.901	0.893	0.892	0.909	0.886	0.744
	<i>F_m</i>	0.880	0.910	0.897	0.893	0.886	0.886	0.881	0.668
	<i>pr</i>	0.826	0.897	0.932	0.889	0.938	0.929	0.919	0.950
	<i>rec</i>	0.943	0.924	0.865	0.897	0.840	0.886	0.845	0.514
100	<i>Acc</i>	0.866	0.907	0.906	0.875	0.841	0.773	0.820	0.746
	<i>F_m</i>	0.874	0.908	0.907	0.869	0.821	0.723	0.791	0.671
	<i>pr</i>	0.824	0.890	0.900	0.912	0.939	0.928	0.938	0.951
	<i>rec</i>	0.930	0.928	0.914	0.829	0.729	0.592	0.684	0.518
500	<i>Acc</i>	0.866	0.907	0.915	0.836	0.760	0.680	0.820	0.744
	<i>F_m</i>	0.874	0.908	0.916	0.815	0.697	0.552	0.791	0.668
	<i>pr</i>	0.824	0.890	0.904	0.933	0.945	0.919	0.938	0.950
	<i>rec</i>	0.930	0.928	0.929	0.723	0.551	0.395	0.684	0.514
1000	<i>Acc</i>	0.866	0.907	0.915	0.829	0.736	0.672	0.820	0.744
	<i>F_m</i>	0.874	0.908	0.916	0.805	0.655	0.535	0.791	0.668
	<i>pr</i>	0.824	0.890	0.904	0.934	0.945	0.920	0.938	0.950
	<i>rec</i>	0.930	0.928	0.929	0.708	0.501	0.378	0.684	0.514

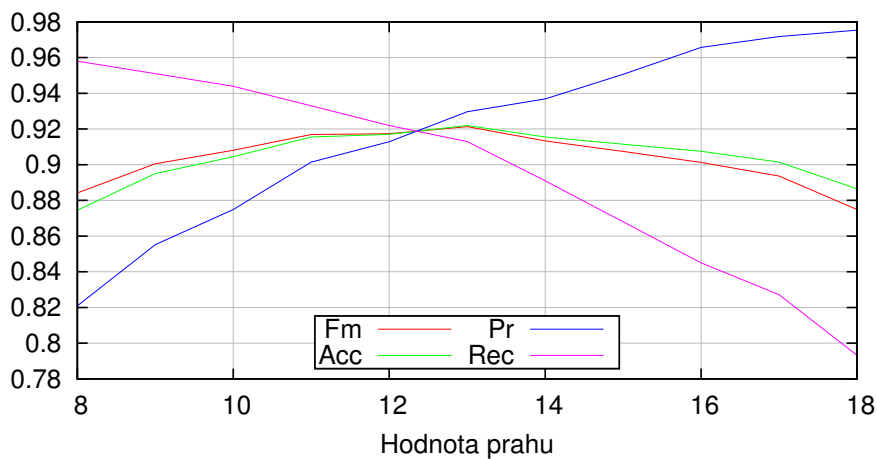
Experimentem pro zkvalitnění klasifikace byla analýza vlivu počtu neuronů ve skryté vrstvě na výsledek rozpoznávání. V našem případě se jednalo spíše o ověření pravidla ze vzorce 7.6. Použili jsme uniformní metodu LBP z důvodu lepších výsledků a pro ní dostali číslo 30, které bylo použito jako referenční bod (viz následující graf), kde je vidět, že pravidlo částečně zafungovalo (při počtu 15 neuronů jsme dostali o tisícinu lepší výsledek). Graf jsme získali při doposud zjištěném ideálním nastavením klasifikátoru (100 trénovacích vzorků a 500 iterací).



Obrázek 7.6: Graf znázorňující závislost výsledků na počtu neuronů ve skryté vrstvě klasifikace pro metodu LBP

$$\text{počet neuronů} = (\text{počet vstupů} + \text{počet výstupů})/2 \quad (7.6)$$

Nastavení klasifikátoru pro metodu LDP probíhalo stejně. Nejdříve jsme se pokusili najít nejvhodnější velikost prahu pro klasifikaci a následně sledovat závislost výsledků na velikosti testovací množiny a počtu iterací.



Obrázek 7.7: Graf metody LDP znázorňující závislost úspěšnosti a F-míry na velikosti prahu při použití klasifikátoru MLP

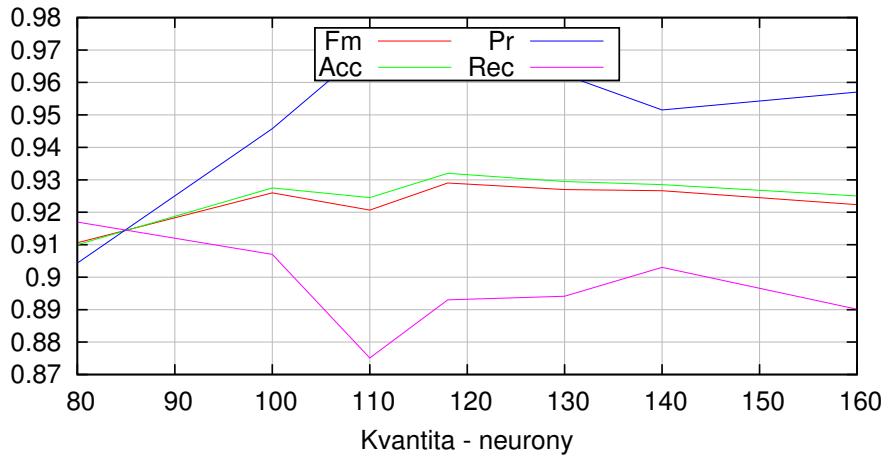
Graf 7.7 ukazuje, že pro metodu LDP je optimální velikost prahu rovna 13 (`threshForClassification=13`). Další nastavené parametry pro tento experiment byly `ldpParameters.order=3`, `ldpParameters.gridSize=24` a `ldpParameters.useUniformBinsOnly=1`. Graf se chová dle očekávání, přičemž při postupném zvětšování prahu pro zachycené histogramy F-míra a úspěšnost stoupá až do bodu, kdy začíná docházet ke klasifikaci zasažených snímků jako snímků nezasažených a křivka úspěšnosti i F-míry začíná povolna klesat.

Následuje tabulka pro zachycení vztahu mezi počtem iterací, velikostí trénovací množiny a F-míře. Některé hodnoty nebyly vypočítány z důvodu již potvrzeného přetrénování klasifikátoru. Výsledky jsou velmi podobné jako u metody LBP s tím rozdílem, že zde vede k vyšší F-míře menší počet iterací a to i na menších trénovacích množinách.

Tabulka 7.2: Závislost výsledků pro metodu LDP a klasifikátor MLP na počtu iterací a velikosti dat

PI	PT	50	75	100	200	300	500	1000	1750
1	<i>Acc</i>	0.836	0.8555	0.894	0.908	0.895	0.908	0.916	0.915
	<i>F_m</i>	0.835	0.854	0.888	0.906	0.890	0.906	0.914	0.913
	<i>pr</i>	0.840	0.862	0.941	0.921	0.935	0.929	0.932	0.929
	<i>rec</i>	0.831	0.846	0.841	0.891	0.849	0.883	0.897	0.898
10	<i>Acc</i>	0.904	0.910	0.932	0.919	0.899	0.918	0.912	0.911
	<i>F_m</i>	0.902	0.905	0.929	0.917	0.895	0.914	0.909	0.908
	<i>pr</i>	0.918	0.958	0.969	0.939	0.923	0.951	0.932	0.934
	<i>rec</i>	0.886	0.857	0.893	0.895	0.869	0.880	0.888	0.883
100	<i>Acc</i>	0.917	0.921	0.922	0.918	0.895	0.923	—	—
	<i>F_m</i>	0.915	0.919	0.921	0.917	0.892	0.920	—	—
	<i>pr</i>	0.928	0.938	0.930	0.922	0.922	0.957	—	—
	<i>rec</i>	0.903	0.900	0.913	0.912	0.863	0.886	—	—
500	<i>Acc</i>	0.917	0.921	0.922	0.918	—	—	—	—
	<i>F_m</i>	0.915	0.919	0.921	0.917	—	—	—	—
	<i>pr</i>	0.928	0.938	0.930	0.922	—	—	—	—
	<i>rec</i>	0.903	0.900	0.913	0.912	—	—	—	—
1000	<i>Acc</i>	0.917	0.921	0.922	—	—	—	—	—
	<i>F_m</i>	0.915	0.919	0.921	—	—	—	—	—
	<i>pr</i>	0.928	0.938	0.930	—	—	—	—	—
	<i>rec</i>	0.903	0.900	0.913	—	—	—	—	—

Posledním experimentem pro klasifikátor MLP bylo nalezení optimálního počtu neuronů ve skryté vrstvě, tedy ověření vzorce 7.6 i pro metodu LDP.

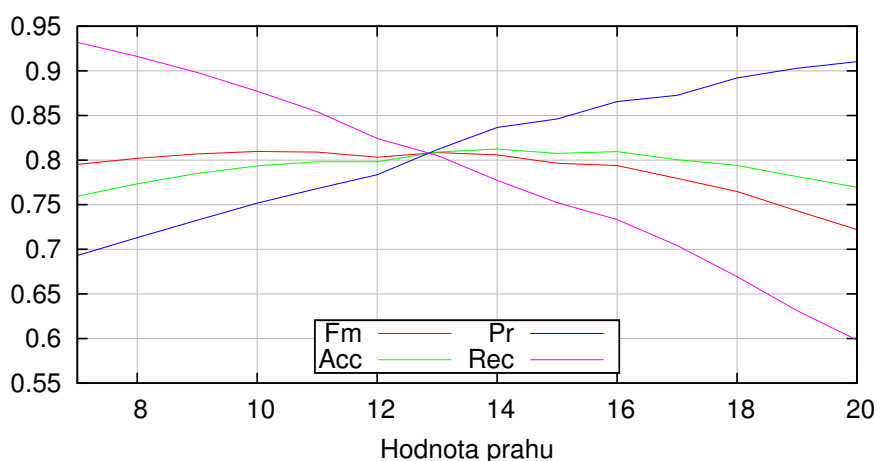


Obrázek 7.8: Graf znázorňující závislost výsledků pro metodu LDP na počtu neuronů ve skryté vrstvě

Díky grafu můžeme zřetelně vidět, že se nám podařilo potvrdit výsledek vzorce, přičemž byl výchozí počet neuronů v tomto případě roven 118.

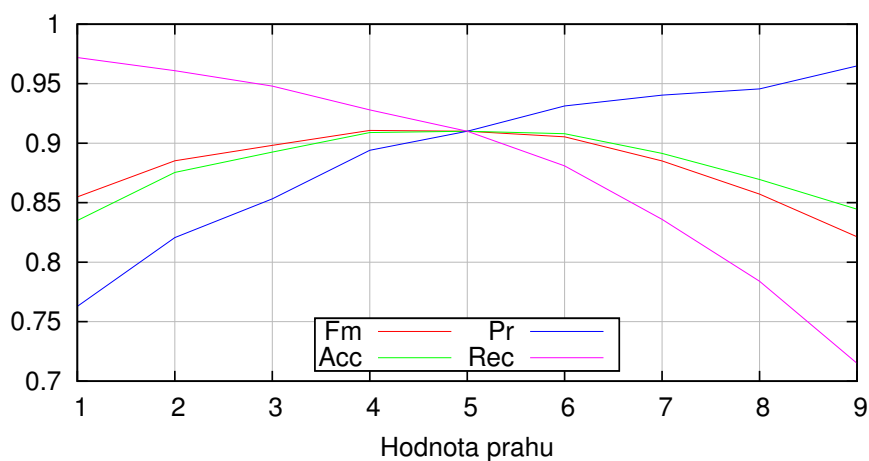
7.2.2 Nastavení parametrů SVM

Následující graf zobrazuje optimalizaci nastavení prahu pro klasifikátor SVM. Byla použita mřížka o velikosti 20 pixelů, která přinesla nejlepší výsledky v předchozím měření. Můžeme vidět, že nejlépe klasifikátor funguje při použití prahu 14, dosahuje zde totiž nejvyšší hodnoty F-míra a úspěšnost je také jedna z nejvyšších. Jako nejlepší se zde jeví práh o počtu 14 pozitivně označených lokálních příznaků.



Obrázek 7.9: Graf znázorňující závislost úspěšnosti a F-míry na velikosti prahu při použití klasifikátoru SVM a metody LBP

Vzhledem k tomu, že pro nastavení parametrů klasifikátoru používáme metodu `train_auto`, která by veškeré závislosti na trénovací množině či počtu iterací zkreslovala, není na místě sestavovat stejnou tabulku jako v případě MLP. Zbývá poslední experiment a to zjištění optimálního prahu pro metodu LDP. Pro měření jsme použili mřížku o velikosti 40 pixelů, kterou jsme zvolili při předchozích experimentech. Na následujícím grafu můžeme vidět, že ideální práh vychází na 5 pozitivních lokálních příznaků.

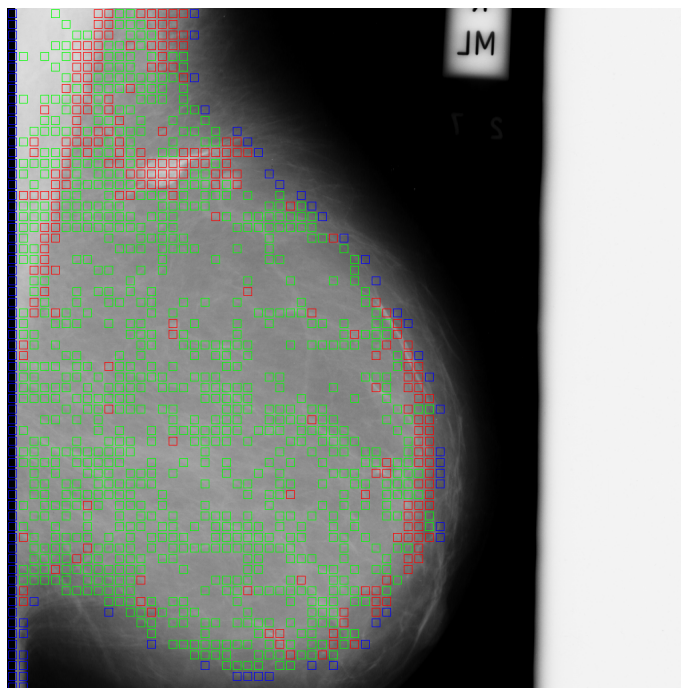


Obrázek 7.10: Graf znázorňující závislost úspěšnosti a F-míry na velikosti prahu při použití klasifikátoru SVM a metody LDP

7.3 Experimentální použití k-means

Jako rozšiřující experiment byl v práci implementován algoritmus k-means. Vize byla taková, že algoritmus zajistí rozdělení příznakových vektorů v podobě lokálních histogramů do několika kategorií z nichž právě jedna budou příznaky reprezentující zasaženou tkáň. Úvaha vznikla na základě skutečnosti, že radiolog je svým okem schopen rozpoznat podezřelou oblast a to z důvodu jiné jasové intenzity. Právě dle jasů by měla metoda k-means vhodně roztrždit lokální příznaky ve snímku.

Ve výsledku se nám nepodařilo z důvodu malého počtu experimentů a kvůli nedostatku času metodu řádně využít, největším problémem byla především náhodná inicializace středů shluků. Pokud bychom tuto část zlepšili dosáhli bychom lepších výsledků.



Obrázek 7.11: Obrázek zachycující experimentální rozdělení množin lokálních histogramů

Funkčnost algoritmu ilustruje obrázek 7.11. Z obrázku je patrné, že algoritmus rozdělil lokální histogramy do čtyř skupin - červené, modré, zelené a neohraničené čtverce dle úrovně intenzity jasů. Podezřelý útvar viditelný v levém horním rohu prsu je pak správně detekován a přiřazen do jedné ze

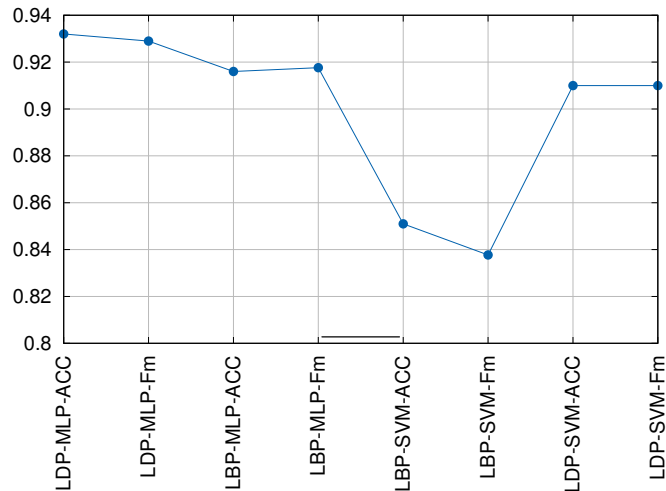
skupin (červená). Můžeme však pozorovat velké množství parazitních příznaků, které do skupiny zařazeny být neměly. Můžeme vidět, že čtverce se nevykreslují mimo prs a to z důvodu vhodného předzpracování snímku (viz 6.4.4), které soustředí průběh metody jen do oblasti detekovaného prsu.

7.4 Shrnutí

První zkoumaná metoda LBP byla metodou nejjednodušší a byla pro nás referenční vzhledem k tomu, že ostatní metody by její výsledky by měly dosahovat lepších výsledků. Podařilo se nám dosáhnout s klasifikátorem MLP F-míry a úspěšnosti dosahující okolo 90 %. Klasifikátor SVM u této metody dosáhl o zhruba 5 % horších výsledků.

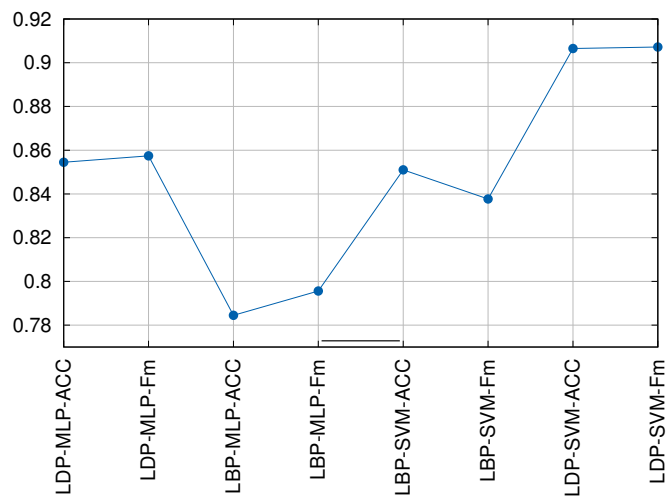
Operátor LDP dosáhl velmi kvalitních výsledků (oba klasifikátory dosahovaly F-míry a úspěšnosti nad 90 %). Pro klasifikační algoritmus MLP byly výsledky nečekaně stabilní. Na druhou stranu, ale klasifikace trvala o poznání déle a výsledky pro klasifikátor SVM byly velice nevyrovnané.

V následujícím grafu jsou nejlepší výsledky (F-míra) pro každý z obrazových deskriptorů v kombinaci s jedním z klasifikátorů. Nejlepších výsledků jsme dosáhli za použití operátorů LDP a klasifikačního algoritmu MLP.



Obrázek 7.12: Graf zobrazující závěrečný přehled metod obrazových deskriptorů s optimalizovanými klasifikátory

Provedli jsme také experiment pro porovnání s článkem [8], abychom dokázali zda je možné použít pro klasifikaci příznak v podobě globálního histogramu. Následující graf předčil naše nejlepší nastavení pro klasifikátor SVM a metodu LBP. Obecně klasifikátor SVM na tuto úlohu zafungoval velmi dobře.



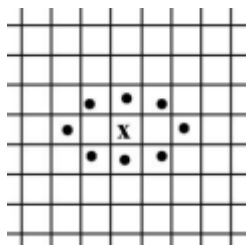
Obrázek 7.13: Graf zobrazující závěrečný přehled metod obrazových deskriptorů s optimalizovanými klasifikátory

8 Další možná rozšíření

Dalším krokem pro vylepšení klasifikace by mohla být implementace rozšíření některé z použitých metod pro extrakci příznaku jako například LBP. V následující kapitole si nastíníme možné modifikace a další rozšíření práce.

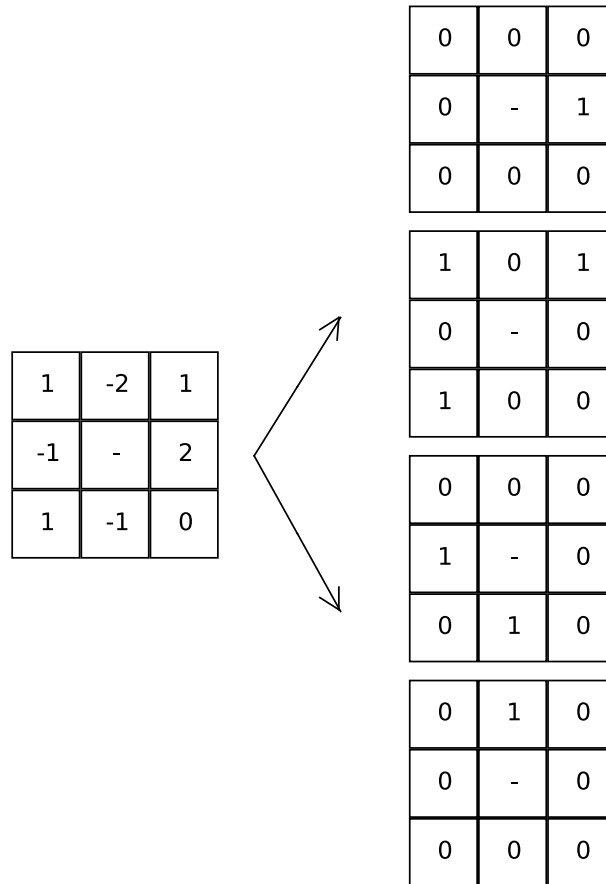
Mezi modifikace LBP patří:

- EBP - s centrálním pixelem neporovnáváme sousedy na kružnici, ale na elipse. Díky tomu získáme tzv. anizotropní informace (závislosti na volbě směru). Metoda je obzvláště vhodná pro rozpoznávání obličejů.
- LTP - používáme ternární prahování namísto binárního, tím bude deskriptor robustnější a můžeme ho tak použít například pro zašuměné obrázky.
- MBP - prahování je prováděno s hodnotou mediánu všech sousedů, opět z důvodu větší robustnosti.
- ILBP - na podobném principu jako MBP. Centrální pixel porovnáváme s intenzitou hodnot všech sousedů oproti konkrétnímu pixelu.
- CSLBP - u této metody porovnáváme pouze hodnoty symetrických sousedů. Díky tomu můžeme ušetřit paměť, protože dostaneme menší histogram.
- EQP - výsledek studie [6]. Jednoduše lze říci, že hodnoty sousedních pixelů odečítáme z elipsy a poté porovnáváme s centrálním pixelem. Navíc předtím provádíme quinární prahování pro co největší robustnost.



Obrázek 8.1: EQP s porovnáváním sousedů na elipse [6]

Nejlépe ze zmíněných metod se jeví EQP a to jak z důvodu vysoké robustnosti tak i z důvodu, že metoda je známá velmi krátce a je zde proto hodně prostoru pro její další aplikaci. V případném dalším pokračování bychom proto doporučili její implementaci a experimentální odzkoušení.



Obrázek 8.2: Ukázka quinárního prahování [6]

Dalším rozvinutím práce by bylo důkladné otestování metody POEM, která je ze všech tří implementovaných metod nejobtížněji nastavitelná a běží dlouho. Měla by být, ale robustnější pro různé množiny dat. Z důvodu velmi dobrých výsledků při rozpoznávání obličejů předpokládáme, že výsledky při použití této metody by měly být ještě lepší než dosažené hodnoty u LBP a LDP.

V neposlední řadě by se aplikace dala sjednotit se selektorem ROI, aby vznikla celistvá komponenta pro přímou klasifikaci snímku bez potřeby vybrat konkrétní ROI.

Posledním možným bodem pro rozšíření aplikace by mohlo být zařazení

dalšího klasifikátoru jako je například hluboká neuronová síť, který by byl schopen přinést ještě lepší výsledky.

9 Závěr

V rámci bakalářské práce byly popsány metody lokálních obrazových deskriptorů LBP, LDP a POEM. Zároveň byla nastíněna problematika věnující se automatickému rozpoznávání rakoviny prsu. Dále byly popsány metody jak detekovat nádor z lékařského hlediska. Také jsou uvedeny informace o použitých klasifikátorech SVM a MLP, které vyhodnocují výsledky.

Podářilo se nám získat několik databází pro detekci rakoviny. Databáze byly popsány v teoretické části. Jako základ byl pro implementaci použit funkční celek pro rozpoznávání obličejů v programovacím jazyce C++ a využitím knihovny OpenCV. Byl přepracován konfigurační soubor a struktura dat použitá v aplikaci. Vznikl algoritmus pro předzpracování celých snímků z mamografu, které by mohly být také klasifikovány. Vzhledem k tomu, že metody obrazových deskriptorů zde byly z části implementovány pro jinou úlohu byla práce rozšířena o použití klasifikátorů MLP a SVM. Finální program tedy umí klasifikovat zasažené či nezasažené ROI s využitím extrakce příznaku za pomoci metod LBP, LDP a POEM s klasifikátory MLP nebo SVM.

Dále byly shrnuty výsledky experimentů s použitím implementovaných metod a navržen další postup pro případné rozšíření aplikace. Navíc byl proveden experiment s použitím algoritmu k-means pro extrakci příznaků a zároveň učení tříd bez pomoci učitele. Nejlepších výsledků jsme dosáhli s metodou LDP (úspěšnost a F-míra nad 91 %) a použitím klasifikátoru MLP při jeho optimálním nastavení.

Literatura

- [1] BRADSKI, G. The OpenCV library, 2000.
- [2] HEATH, M. The Digital Database for Screening Mammography, 2001.
- [3] HEATH, M. Current status of the Digital Database for Screening Mammography, 1998. Dostupné z: <http://goo.gl/3XFmBM>.
- [4] KEGELMEYER, W. Computer-aided mammographic screening for spiculated lesions, 1994. Dostupné z: <http://europepmc.org/abstract/med/8153302>.
- [5] KOŠAŘ, V. Srovnání deskriptorů pro reprezentaci obrazu, 2015.
- [6] NANN, L. Local binary patterns variants as texture descriptors for medical image analysis, 2010. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0933365710000278>.
- [7] OJALA, T. Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns, 2002. Dostupné z: http://vision.stanford.edu/teaching/cs231b_spring1415/papers/lbp.pdf.
- [8] OLIVER, A. False Positive Reduction in Mammographic Mass Detection Using Local Binary Patterns, 2007. Dostupné z: <http://atc.udg.edu/~llado/CVpapers/miccai07.pdf>.
- [9] PETRICK, N. An adaptive density-weighted contrast enhancement filter for mammographic breast mass detection, 1996. Dostupné z: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=481441.
- [10] POLAKOWSKI, W. E. Computer-aided breast cancer detection and diagnosis of masses using difference of Gaussians and derivative-based feature saliency, 1997. Dostupné z: <https://goo.gl/kJZHmZ>.
- [11] SAMPAT, M. P. Computer-Aided Detection and Diagnosis in Mammography, 2005. Dostupné z: <http://sbisc.sharif.edu/~miap/Files/MammographyReview1.pdf>.
- [12] SUCKLING, J. The Mammographic Image Analysis Society Digital Mammogram Database, 1994. Dostupné z: <https://goo.gl/ri2YYJ>.

- [13] VU, N.-S. Face Recognition with Patterns of Oriented Edge Magnitudes, 2010. Dostupné z: <https://goo.gl/660sA9>.
- [14] VYHNÁLEK, L. Radiodiagnostika - Kapitoly z klinické praxe, 1998.

Přílohy

A Obsah přiloženého CD

- Software
 - src – makefile, konfigurační soubor a zdrojové soubory aplikace,
 - bin – spustitelná aplikace.
- Dokumentace
 - text – PDF soubor s textem bakalářské práce,
 - zdroj – zdrojové soubory textu bakalářské práce a obrázky.
- Použitá databáze
 - patches – složka, která byla použita jako `rootDirectory` obsahující požadovanou adresářovou strukturu s obrázky.